

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin Beşinci Maddesinin İkinci Fıkrası Çerçevesinde Bandrol Taşıması Zorunlu Değildir.

MESLEKİ VE TEKNİK ANADOLU LİSESİ BİLİŞİM TEKNOLOJİLERİ ALANI

ESNE

ABANLI

ROGRAMLAMA



MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

NESNE TABANLI PROGRAMLAMA

10

DERS KİTABI

YAZARLAR

ABDULLAH HOCAOĞLU DEVRİM ALTINKURT MURAT İMSİYATOĞLU MUSTAFA NACAR YASEMİN AKPINAR







HAZIRLAYANLAR

DİL UZMANI : PROGRAM GELİŞTİRME UZMANI : ÖLÇME DEĞERLENDİRME UZMANI : REHBERLİK UZMANI : GÖRSEL TASARIM UZMANI : Melek DEMİR Esra YAVUZ Arzu DURSUN URGUN Gülşen YALIN Nalan OFLAS GÜLER









İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak; Sönmeden yurdumun üstünde tüten en son ocak. O benim milletimin yıldızıdır, parlayacak; O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl! Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl? Sana olmaz dökülen kanlarımız sonra helâl. Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım. Hangi çılgın bana zincir vuracakmış? Şaşarım! Kükremiş sel gibiyim, bendimi çiğner, aşarım. Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar, Benim iman dolu göğsüm gibi serhaddim var. Ulusun, korkma! Nasıl böyle bir imanı boğar, Medeniyyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın; Siper et gövdeni, dursun bu hayâsızca akın. Doğacaktır sana va'dettiği günler Hakk'ın; Kim bilir, belki yarın, belki yarından da yakın. Bastığın yerleri toprak diyerek geçme, tanı: Düşün altındaki binlerce kefensiz yatanı. Sen şehit oğlusun, incitme, yazıktır, atanı: Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda? Şüheda fışkıracak toprağı sıksan, şüheda! Cânı, cânânı, bütün varımı alsın da Huda, Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlâhî, şudur ancak emeli: Değmesin mabedimin göğsüne nâmahrem eli. Bu ezanlar -ki şehadetleri dinin temeli-Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım, Her cerîhamdan İlâhî, boşanıp kanlı yaşım, Fışkırır ruh-ı mücerret gibi yerden na'şım; O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl! Olsun artık dökülen kanlarımın hepsi helâl. Ebediyyen sana yok, ırkıma yok izmihlâl; Hakkıdır hür yaşamış bayrağımın hürriyyet; Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

10. <u>sinif</u>



GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsait bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal ATATÜRK

IO. SINIF







MUSTAFA KEMAL ATATÜRK









İÇİNDEKİLER

	КІ́ТАР ТАЛІТІМІ	·13
	ÇALIŞMA ORTAMI VE TEMEL İŞLEMLER	15
- 1	1.1. NESNE TABANLI PROGRAMLAMA ÇALIŞMA ORTAMI	·16
	1.2. C# PROGRAMLAMA DİLİ	·17
	1.3NET FRAMEWORK	·18
	1.3.1. C# ve .NET Framework İlişkisi	- 18
2	1.3.2NET Framework Çalışma Mantığı	- 18
2	1.4. KOD EDİTÖRÜ ARAYÜZ EKRANI	·20
· El	1.4.1. FORM EKRANI	-21
	1.4.2. Araç Kutusu (Toolbox)	-21
-	1.4.3. Ozellikler (Properties)	-23
\geq	1.4.4. Olaylar (Events)	-24
2	1.4.5. Çözüm Penceresi (Solution Explorer)	-25
	1.4.6. Hata Listesi (Error List)	-25
50	1.5. ISIM UZAYLARI (NAMESPACE)	·31
io	1.6. DEGIŞKENLER VE TEMEL VERI TURLERI	·32
	1.6.1. Temel Veri Türleri	- 33
	1.6.2. Değişken Tanımlama	-34
	1.6.3. Değişkene Değer Atama	- 35
	1.6.4. Değişken Isimlendirme Kuralları	- 35
	1.6.5. Değişken Veri Türü Dönüştürme (Convert) Işlemleri	-36
	1.7. ARITMETIKSEL OPERATORLER	·37
	1.7.1. Işlem Onceliği	- 39
	OLÇME VE DEGERLENDIRME - 1	·43

	KARAR VE DÖNGÜ YAPILARI	45
	2.1. KARAR İFADELERİ	46
444	2.1.1. Karşılaştırma Operatörleri	46
Σ	2.1.2. if Yapısı	47
·H	2.1.3. if-else Yapısı	49
-H	2.1.4. else if Yapısı	51
8	2.1.5. İç İçe Şart İfadeleri	52
	2.1.6. Switch-Case	53
Σ	2.2. MANTIKSAL OPERATÖRLER	56
Z	2.2.1. AND(&&) Operatörü	56
Щ	2.2.2. OR() Operatörü	60
	2.2.3. Mantıksal Operatör Önceliği	61
Ň	2.2.4. NOT(!) Operatörü	63
.0	2.3. DÖNGÜLER	65
N	2.3.1. Sayaçlar	65
	2.3.2. Artırma ve Azaltma Operatörleri	66
	2.3.3. For Döngüsü	67
	2.3.4. WHİLE DÖNGÜSÜ	72
	2.3.5. Do-while Döngüsü	72
	2.3.6. Döngüyü Kesme (Durdurma)	74
	2.3.7. Döngüyü Devam Ettirme	76
	2.4. HATA AYIKLAMA	77
	2.4.1. Try-Catch-Finally Bloku	77
	ÖLÇME VE DEĞERLENDİRME - 2	79



	SINIFLAR (CLASSES)	81
	3.1. NESNE TABANLI PROGRAMLAMAYA GİRİŞ	82
	3.1.1. NTP Öncesi	82
	3.1.2. NTP Temel Prensipleri	82
	3.2. SINIFLAR VE NESNELER	82
l	3.2.1. Sınıf Tanımlama	83
	3.2.2. Nesne Oluşturma	84
l	3.3. KAPSULLEME, ALANLAR VE OZELLIKLER (ENCAPSULATION, FIELDS, PROPERTIES)	84
l	3.4. ERIŞIM BELIRLEYICILER (ACCESS MODIFIERS)	85
L	3.5. ALANLAR (FIELDS)	85
1	3.6. OZELLIKLEK (PROPERTIES)	86
l	3.6.1. Sadece Ukunabilir Üzellikler	89
L	3.0.2. Saucce fazilabilit Ozellikier	89
L	3.7. Metotlar (Methods) 3.7.1. Varcaulan Dožori Parametrolor (Ontional Parametore)	90
L	3.7.1. Valsaylian Degen ralameteler (Optional ralametels)	92
L	3.7.2. Isimilerining radiileriel (Nameu radiileris)	93
l	3.7.4 Matodu Solandirma	94
1	3.75 Matot Asiri Viklemeleri (Method Overloads)	97
L	3.8 VAPICI VE VIKICI METOTI AR	96
	3.8.1. Yapıcı Metotlar (Constructors)	96
	3.8.2. Yıkıcı Metotlar (Destructors)	99
	3.9. DEGER VE REFERANS TIPLER	100
	3.9.1. Metotlarda ref ve out Kullanımı	102
	3.10. KALITIM (INHERITANCE)	103
	3.10.1. Hiyerarşik Kalıtım	105
	3.10.2. new Operatörüyle Metot Gölgeleme (Shadowing)	106
	3.10.3. Sanal Metotlar (Virtual Methods)	107
	3.11. SOYUT SINIFLAR (ABSTRACT CLASSES)	108
	3.12. ARAYUZLER (INTERFACES)	110
	I 3 13 COK RICIMI II IK (DOI VMORDHISM)	
		112
	3.14. STATİK SINIFLAR (STATİC CLASSES)	112 115
	3.14. STATİK SINIFLAR (STATİC CLASSES)	112 115 117
	3.14. STATİK SINIFLAR (STATİC CLASSES)	112 115 117 118
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMAŞ (NUMAPALANDIPMALAR)	112 115 117 118 118
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) 3.18. ENUMS (NUMARALANDIRMALAR)	112 115 117 118 118 119 121
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3	112 115 117 118 118 121
	3.14. STATİK SINIFLAR (STATİC CLASSES)	112 115 117 118 118 121
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS)	112 115 117 118 118 121 123
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER	112 115 117 118 118 121 123 124
	3.13. STATIK SINIFILAR (STATIC CLASSES) 3.14. STATIK SINIFILAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFILAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFILAR (SEALED CLASSES) 3.17. PARÇALI SINIFILAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER	112 115 117 118 118 121 123 124 124 124
	3.13. STATİK SINIFLAR (STATİC CLASSES) 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER- 4.1.2. Bir Boyutlu Diziler- 4.1.2. Bir Boyutlu Dizilerin Oluşturulması	112 115 117 118 118 121 123 124 124 124 124
	3.13. STATİK SINIFLAR (TOLINIONIDAN) 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER- 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma	112 115 117 118 118 121 123 124 124 124 124 124 124
	3.13. STATİK SINIFLAR (STATİC CLASSES) 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER- 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizi Elemanlarına Erişim	112 115 117 118 121 123 124 124 124 124 124 125 125
	3.13. STATİK SINIFLAR (STATİC CLASSES) 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER- 4.1.2. Bir Boyutlu Diziler 4.1.3. Bir Boyutlu Dizilerin Oluşturulması 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.5. Dizilerde Foreach Döngüsü Kullanımı	112 115 117 118 121 123 124 124 124 124 124 125 127 128
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER- 4.1.2. Bir Boyutlu Diziler 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar	112 115 117 118 118 121 123 124 124 124 124 124 125 127 128 128
	3.13. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar 4.1.6. Bir Boyutlu Diziler	112 115 118 118 121 123 124 124 124 124 124 124 125 125 128 128 123 124
	3.13. GOR BASHLLER (FOLLEOR LASSES) 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1. Bir Boyutlu Diziler 4.1.3. Bir Boyutlu Dizilerin Oluşturulması 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilere Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Dizilere- 4.1.8. İki Boyutlu Dizilere-	112 115 118 118 121 123 123 124 124 124 124 124 127 128 128 133 134 134 128
	3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1. Bir Boyutlu Diziler 4.1.3. Bir Boyutlu Dizilerin Oluşturulması 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Diziler 4.1.8. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler	112 115 118 118 121 123 123 124 124 124 124 124 125 128 128 133 134 128 128 128 128 128 128 128 124 124 124 124 124 124 124 124 125 124 124 125 124 124 125 124 124 124 125 124 124 124 125 128 128 124 124 128 124 124 124 124 124 124 124 124 128 128 128 128 128 128 124 128 128 124 128 124 128 124 128 124 128 124 128 124 127 128
	3.13. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1.1. Bir Boyutlu Dizilerin Oluşturulması 4.1.2. Bir Boyutlu Dizilere Değer Aktarma 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilere Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Diziler 4.1.8. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler 4.1.10. İki Boyutlu Diziler 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim	112 115 117 118 118 119 121 123 124 125 124 125 126 127 128 133 134 135 137 137
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Diziler 4.1.7. Çok Boyutlu Diziler 4.1.8. İki Boyutlu Diziler 4.1.9. İki Boyutlu Diziler 4.1.9. İki Boyutlu Dizi Tanımlama 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim	112 115 117 118 118 121 123 123 124 124 124 128 128 128 133 134 137 141 141
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilere Değer Aktarma 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilerde Kullanımı 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Diziler- 4.1.8. Iki Boyutlu Dizilere Değer Aktarma 4.1.9. İki Boyutlu Dizilere- 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Diziler- 4.1.8. Iki Boyutlu Dizilere- 4.1.9. İki Boyutlu Diziler- 4.1.6. Bir Boyutlu Dizilere- 4.1.8. Iki Boyutlu Diziler- 4.1.9. İki Boyutlu Diziler- 4.1.9. İki Boyutlu Diziler- 4.1.10. İki Boyutlu Dizi Tanımlama 4.1.2. KOLEKSİYONLAR 4.1.3. Bir Boyutlu Dizi Elemanlarına Erişim 4.2. KOLEKSİYONLAR- 4.2.1. Boxing (Kutulama)-Unboxing (Kutu Aç	112 115 117 118 118 119 121 123 124 124 125 127 128 133 134 135 137 141 142 142 142 142 142
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilere 4.1.7. Çok Boyutlu Dizilere 4.1.8. İki Boyutlu Dizilere 4.1.9. İki Boyutlu Dizilere 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.1.2. Bir Boyutlu Dizilere 4.1.4. Bir Boyutlu Dizilere 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizi Tanımlama 4.1.7. Çok Boyutlu Dizi Tanımlama 4.1.9. İki Boyutlu Dizi Elemanlarına Erişim 4.1.10. İki Boyutlu Dizi Elemanlarına Erişim 4.2. KOLEKSİYONLAR 4.2.1. Boxing (Kutulama)-Unboxing (Kutu Açma) 4.2.2. AırayList Koleksiyonu	112 115 117 118 118 119 121 123 124 125 124 125 127 128 133 134 135 137 137 141 142 142 142 142 142 142 144
	 3.14. STATİK SINIFLAR (STATİC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilerin Oluşturulması 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanını 4.1.6. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar 4.1.8. İki Boyutlu Dizi Elemanlarına Erişim 4.1.9. İki Boyutlu Dizi Elemanlarına Erişim 4.2. KOLEKSİYONLAR 4.2. ArrayList Koleksiyonu 4.2.4. Oueue-Stack Koleksiyonları 	112 115 118 118 121 123 123 124 124 124 124 128 128 128 128 128 124 144 144 148 148 148 148 148 148 148 148 150
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1. DİZİLER 4.1.3. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilere Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Dizilere 4.1.8. İki Boyutlu Dizilere 4.1.9. İki Boyutlu Dizilere 4.1.10. İki Boyutlu Dizilere 4.1.2. ArrayList Koleksiyonu 4.1.3. İki Boyutlu Dizilerde Kullanıları Özellikler ve Metotlar 4.1.4. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar 4.1.10. İki Boyutlu Diziler 4.1.10. İki Boyutlu Diziler 4.1.10. İki Boyutlu Diziler 4.2.1. Boxing (Kutulama)-Unboxing (Kutu Açma) 4.2.2. ArrayList Koleksiyonu 4.2.3. List Koleksiyonu 4.2.4. Queue-Stack Koleksiyonları- 4.2.5. Dictionary Koleksiyonu <td> 112 115 118 118 123 123 123 124 124 124 124 128 128 128 128 128 124 144 144 144 155 155 155</td>	112 115 118 118 123 123 123 124 124 124 124 128 128 128 128 128 124 144 144 144 155 155 155
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. İSİMSİZ SINIFLAR (STATİC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTİAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. Bir Boyutlu Diziler 4.1. Bir Boyutlu Diziler 4.1.3. Bir Boyutlu Diziler 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilered Kullanılan Özellikler ve Metotlar 4.1.7. Çok Boyutlu Dizileremanlarına Erişim 4.1.8. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizileremanlarına Erişim 4.1.9. İki Boyutlu Dizi Lemanlarına Erişim 4.2. KOLEKSİYONLAR 4.2.1. Boxing (Kutulama)-Unboxing (Kutu Açma) 4.2.2. ArrayList Koleksiyonu 4.2.3. List Koleksiyonu 4.2.4. Queue-Stack Koleksiyonları- 4.2.6. Hashtable Koleksiyonu	112 115 118 118 123 123 123 124 124 124 124 124 128 128 128 128 124 144 144 155 155 156
	3.14. STATIK SINIFLAR (STATIC CLASSES) 3.15. ISIMSIZ SINIFLAR (STATIC CLASSES) 3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES) 3.17. PARÇALI SINIFLAR (PARTIAL CLASSES) 3.18. ENUMS (NUMARALANDIRMALAR) ÖLÇME VE DEĞERLENDİRME - 3 DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS) 4.1. DİZİLER 4.1.1. Bir Boyutlu Diziler 4.1.2. Bir Boyutlu Dizilerin Oluşturulması 4.1.3. Bir Boyutlu Dizilere Değer Aktarma 4.1.4. Bir Boyutlu Dizilere Değer Aktarma 4.1.5. Dizilerde Foreach Döngüsü Kullanımı 4.1.6. Bir Boyutlu Dizilere Değer Aktarma 4.1.7. Çok Boyutlu Dizilere 4.18. İki Boyutlu Dizilere beğer Aktarma 4.1.9. İki Boyutlu Dizilere 4.1.9. İki Boyutlu Dizilere 4.1.9. İki Boyutlu Dizilere 4.1.1.8. İki Boyutlu Dizilere 4.1.2. Cok Boyutlu Dizilere 4.1.3. Bir Boyutlu Dizilere 4.1.4. Bir Boyutlu Dizilere 4.1.5. Dizilerde Foreach Döngüsü Kullanını 4.1.6. Bir Boyutlu Dizilere 4.1.7. Çok Boyutlu Dizi Tanımlama 4.1.9. İki Boyutlu Dizi Elemanlarına Erişim 4.2.4. KOLEKSİYONLAR 4.2.5. Dixing (Kutulama)-Unboxing (Kutu Açma) 4.2.4. Queue-St	112 115 117 118 119 121 123 124 125 124 125 127 128 133 134 135 137 134 135 137 138 139 1314 135 136 137 138 139 1314 135 136 137 138 139 1314 135 136 137 138 139 1314 132 133 134 135 136 150 151 152 153 157

10

10. sinif

3. ÖĞRENME BİRİMİ

4. ÖĞRENME BİRİMİ

	FORM UYGULAMALARI	163
	5.1. FORMLAR	164
-	5.1.1. Form Sinifi	164
- HE	5.1.2. Kontrol Sınıfı	168
Σ	5.1.3. Konteyner Kontrolleri	171
-	5.2. MENULER	175
	5.2.1. MenuStrip Kontrolü	176
	5.2.2. ContextMenuStrip Kontrolü	179
1.1	5.3. ILETIŞIM KUTULARI (DIALOG BOXES)	179
	5.3.1. Mesaj Iletişim Kutusu (MessageBox)	180
2	5.3.2. Dosya Kaydet Iletişim Kutusu (SaveFileDialog)	181
Z	5.3.3. Dosya Aç İletişim Kutusu (OpenFileDialog)	182
Щ	5.3.4. Yazdırma İletişim Kutusu (PrintDialog)	183
œ	5.3.5. Yazı Tipi İletişim Kutusu (FontDialog)	184
»О	5.3.6. Renk Iletişim Kutusu (ColorDialog)	184
:0	5.4. VERI DOGRULAMA (VALIDATION)	184
1.1	5.4.1. Ipucu (ToolTip)	185
	5.4.2. Veri Girişi Doğrulama (Input Validation)	185
	5.4.3. Veri Girişi Maskeleme (MaskedTextBox)	188
	5.5. VERI BAGLAMA (DATA BINDING)	190
	5.5.1. Basit Veri Bağlama (Simple Data Binding)	190
	📮 👖 5.5.2. Kompleks Veri Bağlama (Complex Data Binding)	192
	OLÇME VE DEGERLENDIRME - 5	197
_		100
	VERI TABANI IŞLEMLERI	199
	6.1. VERI TABANI YAZILIMININ KURULUMU	200
	6.1.1. Veri Tabani Yonetim Sistemi (Database Management System)	201
	6.1.2. SQL (Structured Query Language)	201
	6.1.3. Büyük Veri (Big Data) ve Veri Madenciliği	201
	6.1.3.1. Big Data	201
	6.1.3.2. Veri Madenciliği	202
	6.1.4. MySQL Veri Tabanı	202
	6.1.4.1. MySQL Veri Tabanının Kurulumu	202
	6.2. MYSQL SERVER ARAYUZ (WORKBENCH) EKRANI	212
100	6.3. VERI TURLERI	214
Σ	6.4. VERI TABANI TASARIMI	214
	6.4.1. Veri Tabanı Olusturma	215

6.1.4. MySQI Veri Taban	202
6.1.4.1 MySQL Veri Tabaninin Kurulumu	- 202
6.2. MYSOL SERVER ARAYIZ (WORKBENCH) EKRANT	212
	214
6.4. VERİ TABANI TASARIMI	214
6.4.1. Veri Tabanı Olusturma	215
6.4.2. Veri Tabanında Anahtarlar (Kevler)	216
6.4.3. Tablo Olusturma	217
6.5. TABLOLARI İLİSKİLENDİRME	221
6.5.1. İlişkişel Veri Tabanları	221
6.5.2. Tablolar Arası Bağlantı Yapılması	224
6.6. VERİ TABANINA BİLGİ GİRİŞİ	228
6.7. SQL KOMUTLARI KULLANIMI	232
6.7.1. Select Devimi	232
6.7.2. Where Sart İfadesi	233
6.7.3. Karsılaştırma Operatörleri	234
6.7.4. Mantıksal Operatörler	234
6.7.5. Arama Operatörü	235
6.7.6. Order By Komutu (Sıralama)	235
6.7.7. Insert Into Komutu (Kayıt Ekleme)	235
6.7.8. Update Komutu (Kayıt Güncelleme)	236
6.7.9. Delete Komutu (Kayit Silme)	236
6.7.10. Create	236
6.7.11. Alter	236
6.7.12. Drop	237
6.8. MYSQL VERİ TABANI ALMA VE YÜKLEME	237
6.9. SQL VE NTP BAĞLANTISI	238
6.9.1. Form Tasarımları	238
6.9.2. Form Ozellikleri	240
6.9.3. TabControl Bileşeni	240
6.9.4. ImageList Bileşeni	242
6.10. ADO.NET	244
6.11. VERI TABANI BAGLANTISI VE BILEŞENLERIN EKLENMESI	244



6.11.1. MySQL Bağlantı Kontrolü	245
6.11.2. MySQL Connection String	246
6.11.3. Projeve Giris	246
6.11.4. DataGridView Bileseni	248
6.11.5. Proienin Kodlamasina Giris	248
6.11.6. Dersler Sekmesi	249
6.11.7. Notlar Sekmesi	250
6.11.8. Öğrenci İslemleri Sekmesi	251
6.11.9. Anasayfa Sekmesinin Doldurulması	252
6.12. KAYIT EKLEME	255
6.13. ARAMA METODU	260
6.14. EKLEME, SİLME VE GÜNCELLEME İŞLEMLERI	260
6.14.1. Öğrenci Sekmesi İşlemleri	261
6.14.1.1. Öğrenci Sekmesi Kayıt Ekleme	262
6.14.1.2. Öğrenci Sekmesi Kayıt Güncelleme	263
6.14.1.3. Öğrenci Sekmesi Kayıt Silme	264
6.14.2. Notlar Sekmesi	266
6.14.2.1. Notlar Sekmesi Kayıt Güncelleme	266
6.14.2.2. Notlar Sekmesi Kayıt Silme	268
6.14.3. Dersler Sekmesi	269
6.14.3.1. Dersler Sekmesi Kayıt Ekleme	269
6.14.3.2. Dersler Sekmesi Kayıt Silme	270
6.15. KURULUM (SETUP) HAZIRLAMA	270
6.16. ENTITY FRAMEWORK	278
ÖLÇME VE DEĞERLENDİRME - 6	293
CEVAP ANAHTARLARI	294
GÖRSEL KAYNAKÇA	301
KAYNAKCA	306

NESNE TABANLI PROGRAMALAMA

12



















ÇALIŞMA ORTAMI VE TEMEL İŞLEMLER

NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Nesne tabanlı programlama yazılımı çalışma ortamını tanımlayacak,
- Nesne tabanlı programlama yazılımı ortamında yeni bir proje oluşturacak,
- .NET Framework kavramını ve çalışma mantığını açıklayacak,
- Form ekranı üzerine nesne ekleyecek,
- Form uygulamasında nesneye kod yazacak,
- İsim uzaylarını programda tanımlayıp kullanacak,
- Değişken kavramını tanımlayacak,
- Değişken türlerini sıralayacak,
- Kod yazarken değişkenleri yazım kurallarına uygun kullanacak,
- Aritmetiksel operatörleri kavrayacak,
- Aritmetiksel operatörlerin işlem önceliklerini öğreneceksiniz.

ANAHTAR KELİMELER

Proje, programlama, form, nesne, kod, isim uzayı, değişken, aritmetiksel operatörler





HAZIRLIK ÇALIŞMALARI

- 1. Programlama dili size ne ifade ediyor?
- 2. Hangi programlar C# dili kullanılarak yapılmış olabilir?

1.1. NESNE TABANLI PROGRAMLAMA ÇALIŞMA ORTAMI



Görsel 1.1: Nesne tabanlı programlama çalışma ortamı

Program, herhangi bir elektronik cihaza bir işlem yaptırmak için yazılan komutlar dizisidir. Programlar; bilgisayar, cep telefonu, tablet, elektronik ev eşyaları, araba ve daha birçok yerde kullanılır.

Elektronik cihazlara bilgisayar, cep telefonu, akıllı saat, akıllı televizyon vb. örnek olarak verilebilir (Görsel 1.1). Program yazmak denilince akla ilk gelen bilgisayarlar olsa da günümüzde pek çok elektronik cihaza kod yazılabilir. Bilgisayar ve diğer elektronik cihazlar çalışma prensibi olarak programları kullanıp önce girilen bilgiyi alır, ardından bu bilgiyi işler ve en sonunda ortaya bir sonuç çıkarır. Bu sonuç bazen bir mesaj bazen de bir işlemi gerçekleştirme şeklindedir. Bu durum, insanların birbiriyle iletişimi gibi düşünülebilir. İnsanlar da konuşmalarında önce karşı tarafın ifadelerini algılar, algıladıklarını işler ve son olarak karşı tarafa bir cevap verir.

Programlama Temelleri dersinde de bahsedildiği gibi programlar önceden **"1" ve "0"** lardan oluşan kod blokları ile **makine dilinde** yazılırdı. Makine dilinde kod yazılması ve yazılan kodun anlaşılması oldukça zordu. Bu nedenle daha okunaklı ve kolay kod yazılabilen **Assembly dilleri** geliştirildi. İlerleyen zamanlarda daha çok anlaşılır, konuşma diline daha yakın ifadelerle kod yazma imkânı sağlayan C, C++, Turbo Pascal, Visual Basic, C#, Python gibi **programlama dilleri** kullanıldı.

Programlama; bilgisayar programlarının yazılması, test edilmesi ve bakımının yapılması sürecine verilen isimdir. Bu sürecin daha verimli geçmesi için amaca yönelik bir programlama dili seçilmelidir. Nesne Tabanlı Programlama ders kitabının içeriğinde C# programlama dilinin kullanımı anlatılacaktır.





1.2. C# PROGRAMLAMA DİLİ



Görsel 1.2: Program çalışma süreci

C# programlama dili, nesne tabanlı olarak geliştirilmiş bir dildir. C# programlama dili ile;

- Mobil uygulamalar,
- Konsol uygulamaları,
- Web servisleri,
- Dinamik kütüphaneler (DLL),
- Oyun tasarımı,
- Form uygulamaları yapılabilir.

Günlük hayatın birçok alanında büyük küçük pek çok şirket C# ile geliştirilmiş programlar kullanmaktadır (Görsel 1.2). C# programlama dilinin çok tercih edilmesinin sebeplerinden bazıları şunlardır:

- Yazılması ve anlaşılması kolay kod yapısına sahiptir.
- Yeni teknolojileri destekler.
- Kullanışlıdır.
- Ekip çalışmasına elverişlidir.
- Kullanıcıyla etkileşimlidir.
- Grafik arayüzlü tasarımlar yapılabilir.
- Ağ üzerinden birbiriyle uyumlu çalışabilir.
- Çevrimiçi veya çevrimdışı kullanılabilir.
- QR kod okuyucu, kamera, yazıcı vb. cihazlarla etkileşimlidir.
- Verileri depolayıp işleyerek analiz yapabilir.
- Sosyal medya platformları ile etkileşimlidir.
- Cep telefonu uygulamaları ile etkileşimlidir.
- Yapay zekâ teknolojisi kullanılarak yüz tanıma, nesne tanıma, ses tanıma işlemleri yapılabilir.
- Birçok özelliğe sahip farklı programlar geliştirebilme imkânı sağlar.







1.3. .NET FRAMEWORK



Framework kelimesi **geliştirme çatısı** anlamına gelmektedir. .NET Framework, çoğu kişi tarafından bir programlama dili sanılmaktadır fakat programlama dillerinden bağımsız ve farklı programlama dilleri ile ortak çalışma imkânı sağlayan bir uygulama geliştirme platformudur. .NET logosu Görsel 1.3'te görülmektedir. Farklı dilleri bilen programcılar ortak bir projeyi yürütebilir. .NET Framework çalışma mantığı bunun için uygundur. Yaygın kullanılan çoğu dil (C#, Visual Basic, Visual C++, Visual F#, Python) .NET Framework desteklidir. Ayrıca .NET Framework altyapısında kullanıma sunulmuş hazır kod kütüphaneleri ile kod yazma çok daha hızlı ve verimli olmaktadır. .NET Framework kütüphaneleri tüm .NET dillerinde ortak kullanılır. Bu yüzden .NET çok güçlü bir Framework'tür.

1.3.1. C# ve .NET Framework İlişkisi

C# bir programlama dilidir, .NET Framework ise C# dili ve birçok dilin kütüphanelerinin yüklü olduğu bir uygulama geliştirme platformudur. C# dilinde kullanılan kütüphanelerin tümü .NET Framework kütüphaneleridir.

1.3.2. .NET Framework Çalışma Mantığı

Programlama dilleri ile yazılan kodlar makine için anlamlı değildir, kodların makine dilinde yazılması veya makine diline çevrilmesi gerekir. Program yazarken kullanılan kodlar derlendiğinde doğrudan makine diline çevrilmez. Görsel 1.4'te görüldüğü gibi kodlar önce **ara dil** (Intermediate Language) koduna, ardından da **çalışma zamanı derleyicisi** (Just-In-Time Compiler) tarafından makine diline çevrilir. Böylelikle kodlar sorunsuz bir şekilde çalışır. Bu işlemlerin yapılmasını .NET Framework altyapısı sağlar.



Görsel 1.4: .NET Framework çalışma mantığı







Uygulama-1

Adım 1: Kod editörü arayüzünü açarak Görsel 1.5'te görülen başlangıç ekranından Create a new project'i seçiniz ve yeni bir proje oluşturunuz.



Görsel 1.5: Yeni proje oluşturma

Adım 2: Görsel 1.6'da görülen ekrandan Form Uygulamalarını (.NET Framework) seçiniz.



Görsel 1.6: Programlama dili seçimi







ÇA ÖĞRENME BİRİMİ	LIŞMA ORTAMI VE TEMEL İŞLEMLER	
Adım 3	3: Projeye isim veriniz ve projenin kayıt yerini belirleyiniz (Gö	rsel 1.7).
		- 🗆 ×
	Configure your new project	
	Windows Forms App (NET Framework) C* Windows Desktop	
	Project name	
	MerhabaDunya	
	Location	
	C:\Users\bilisimPc\source\repos *	
	Solution name 🕦	
	MerhabaDunya	
	Place solution and project in the same directory	
	Framework	
	INELE Framework 4.7.2 *	

Görsel 1.7: Proje adı belirleme

1. "MerhabaDunya" isimli bir Form uygulaması oluşturunuz.

2. Oluşturduğunuz "MerhabaDunya" isimli projeyi kaydedip kapattıktan sonra **var olan projeyi açma** seçeneği ile yeniden açınız.

1.4. KOD EDİTÖRÜ ARAYÜZ EKRANI

Görsel 1.8'de görüldüğü gibi kod editörü arayüz ekranı açıldığında ekranın üst tarafında **Menü Çubuğu** ve **Araç Çubuğu**, ekranın orta kısmında **Form Ekranı** ve form ekranının etrafında **4 adet panel** bulunmaktadır. Bu panellerin yerleri isteğe bağlı olarak sabitlenebilir, sürükle bırak yöntemi ile yerleri değiştirilebilir veya paneller tamamen kaldırılabilir.



Görsel 1.8: Kod editörü arayüz ekranı





1.4.1. Form Ekranı

Form ekranı, programın görsel tasarımının yapıldığı yerdir (Görsel 1.9). **Araç Kutusu (Toolbox)** panelinde bulunan nesneler form üzerinde istenilen pozisyona yerleştirilerek programın tasarımı yapılır.



Görsel 1.9: Form ekranı

1.4.2. Araç Kutusu (Toolbox)

Form üzerinde tasarım için kullanılabilecek nesneler bu panelde bulunur. Görsel 1.10'da görüldüğü gibi tüm nesneler aynı anda veya işlevlerine göre çeşitli kategorilerde listelenebilir. Örneğin, Data bölümüne tıklandığında **Data (Veri)** ile ilgili nesneler listelenir. Ayrıca **Search Toolbox (Arama Çubuğu)** ile istenilen nesnenin ismi yazılarak da nesne listelenebilir.



Görsel 1.10: Araç kutusu

Toolbox paneli kullanılarak form üzerine nesne ekleme işlemi iki farklı şekilde yapılabilir:

- 1. Eklenecek nesne çift tıklanır.
- 2. Eklenecek nesne sürüklenip formun üzerinde herhangi bir pozisyona bırakılır.

Sıra Sizde

1. Toolbox panelinde Label ve TextBox nesnelerini çift tıklayarak bu nesneleri formun üzerine yerleştiriniz.

2. Label ve TextBox nesnelerini formun ortasına yerleştiriniz.







En Çok Tercih Edilen Toolbox Nesneleri

Toolbox'ta 50'den fazla nesne bulunmaktadır. Bu nesnelerin büyük çoğunluğu Görsel 1.11'de ve Görsel 1.12'de görülmektedir. Projelerde en çok kullanılan nesneler aşağıda listelenmiştir.



Button: Programlarda bazı kodları çalıştırmak için kullanılan komut düğmeleridir. Button nesnesine tıklandığında içeri doğru basma efekti gerçekleştiği için tıklama (Click) olayları için vazgeçilmez bir nesnedir.

CheckBox: Kullanıcıya bir veya aynı anda birden çok seçeneği işaretleme imkânı sağlayan nesnedir.

ComboBox: Açılır liste ile açılan seçenekler arasından seçim yapılmasına olanak sağlayan araçtır. Listeye yeni eleman ekleme ve çıkarma işlemleri, tasarım ekranından veya program çalışırken kod ile yapılabilir.

DateTimePicker: Tarih ve saat seçme işlemlerine olanak sağlayan nesnedir.

GroupBox: Form elemanlarını kendi aralarında gruplamak için kullanılan nesnedir. Nesneler gruplar hâlinde olduğu için daha anlaşılır tasarımlar yapılabilir.

Label: Form üzerinde bilgi vermek için kullanılan nesnedir.

ListBox: Sunulacak seçeneklerin açık bir liste hâlinde gösterildiği nesnedir. **MenuStrip:** Programda menü başlıkları ve alt başlıkları oluşturmak için kullanılan nesnedir.

Görsel 1.11: En çok tercih edilen nesneler-1

PictureBox: Form üzerinde resim göstermek için kullanılan nesnedir.

ProgressBar: Yapılan bir işlemin ne kadarının tamamlandığını göstermek için kullanılan nesnedir.

RadioButton: CheckBox nesnesinden farklı olarak birçok seçenek içinden sadece birinin seçilmesine imkân sağlayan nesnedir.

RichTextBox: Birden çok satır içine metin girişi yapılabilen nesnedir.

TabControl: Form elemanlarının gruplara ayrıldığı, grupların içindeki elemanları görmek için sekmelerin kullanıldığı nesnedir. Her sekme, bir grubu temsil eder.

TextBox: İçine tek satır metin girişi yapılabilen nesnedir. Bilgi girişi için en çok tercih edilen nesnedir.

Timer: Kodların zamanlanarak çalışmasını sağlayan nesnedir.



Görsel 1.12: En çok tercih edilen nesneler-2

10. SINIF

Sıra Sizde

En çok tercih edilen nesnelerin özelliklerini araştırınız ve bulduğunuz özellikleri uygulama içinde kullanınız. Çalışmanızı yaparken öğretmeninizden destek alınız.





1.4.3. Özellikler (Properties)

BİRİMİ

Form nesnesinin ve diğer tüm nesnelerin özelliklerinin listelendiği, değiştirildiği ve ayrıca nesnelere ait **olayların (events)** listelendiği paneldir.

Properties	≂ ₽ >
button1 System.Win	dows.Forms.Button
월 🖓 🦻 🗲 🌙	5
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImagel	ayo Tile
Cursor	Default
E FlatAppearance	
FlatStyle	Standard
🗄 Font	Microsoft Sans Serif; 8,
ForeColor	ControlText
Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageKey	(none)
ImageList	(none)
RightToLeft	No
Text	button1 🗸
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBack	Colc True
UseWaitCursor	False
Behavior	
🗄 Data	

Her nesnenin kendine ait benzersiz özellikleri olduğu gibi diğer nesneler ile ortak özellikleri de bulunmaktadır. Tüm nesnelerin en temel ortak özelliği, **isim (name)** özelliğidir. Örneğin, Picture-Box nesnesinin genişlik ve yükseklik özelliği vardır fakat Check-Box nesnesinin yoktur. Button nesnesinin yazı rengi özelliği vardır fakat PictureBox nesnesinin yoktur. DateTimePicker nesnesinin tarih belirleme özelliği vardır fakat TextBox nesnesinin yoktur.

Görsel 1.13'te seçili **button1** nesnesine ait bazı özellikler aşağıda verilmiştir.

BackColor: Nesnenin arka plan rengini değiştirir.

BackgroundImage: Nesnenin arka planına resim ekler.

Font: Nesnenin yazı tipini, boyutunu ve kalınlığını değiştirir.

ForeColor: Nesnenin yazı rengini değiştirir.

Text: Nesnenin yazı metnini değiştirir.

TextAlign: Nesnenin yazısını hizalar.

Görsel 1.13: Özellikler sekmesi

Sıra Sizde >

Sizler de form üzerine 2 adet farklı nesne ekleyip her nesnenin en az 7 özelliğini değiştiriniz.

Uygulama-2

Adım 1: İki kişilik gruplar hâlinde eşleşiniz. Yanınızdaki veya öğretmeninizin belirlediği bir arkadaşınız ile grup oluşturabilirsiniz.

Adım 2: Görsel 1.14'te görüldüğü gibi Form üzerine Button, CheckBox, Label ve PictureBox nesnelerini ekleyiniz. Toolbox panelinden sürükle bırak yöntemi ile nesneleri form üzerine yerleştiriniz.

Adım 3: Bu nesnelerin ortak ve farklı özelliklerini belirleyip defterinize not alınız. Form üzerindeki nesnelerin hepsi fare ile aynı anda seçildiğinde properties panelinde nesnelerin sadece ortak özellikleri listelenmektedir.



Görsel 1.14: Özellikler paneli uygulama-1







1.4.4. Olaylar (Events)

Her nesnenin form üzerinde bir görevi bulunmaktadır. Bazı nesneler sadece programın işlevi ile ilgili bilgiyi ve görseli yansıtmak için kullanılır. Bazı nesneler ise belli durumlarda (üzerine tıklandığında, bir tuşa basıldığında vs.) kod parçacıklarını çalıştırmak için kullanılır. Nesneler, kullanıcı ile etkileşimi olaylar sayesinde sağlamaktadır. Günlük hayatta sosyal medya uygulamalarında, web sitelerinde, oyunlarda ve daha birçok alanda nesnelere tanımlanmış olay metotları çalışarak kullanıcıyla etkileşim sağlanır. Olay metotları tanımlanarak nesnelerin hangi durumda, nasıl kodlar çalıştırabileceği belirlenir. Örneğin; butona tıklandığında şifre kontrolünün yapılması, klavyeden sağ ok tuşuna basıldığında bir sonraki resmin gösterilmesi, PictureBox nesnesinin üzerine çift tıklama yapıldığında resme ait ilgili bilgilerin MessageBox ile göstermesi vb. Olaylar için metot tanımlanırken istenilen olayın adının hemen yanındaki boş beyaz kutucuğa çift tıklanır ve otomatik olarak aşağıdaki gibi bir metot oluşturulur.



Sıra Sizde >

Form nesnesinin Load, Enter, Click olaylarına ait metotları oluşturunuz.

Uygulama-3

Adım 1: Form üzerine TextBox ve Button nesnelerini ekleyiniz. Adım 2: Eklediğiniz nesnelerin özeliklerini Görsel 1.16'daki gibi tasarıma uygun hâle getiriniz.

Adım 3: Form üzerindeki nesnelerin ortak olan olaylarını belirleyip sınıf arkadaşlarınızla paylaşınız. Form üzerindeki nesnelerin hepsi fare ile aynı anda seçildiğinde properties panelinde nesnelerin sadece ortak olayları listelenmektedir.



Görsel 1.16: Özellikler paneli uygulama-2





CALIŞMA ORTAMI VE TEMEL İŞLEMLER



ÖĞRENME BİRİMİ

Proje ile ilgili tüm dosya ve klasörlerin listelenerek silme, kopyalama, taşıma, isim değiştirme işlemlerinin yapılabildiği paneldir (Görsel 1.17). Projenin detaylı bir haritası gibi düsünülebilir. Cözüm penceresi ile projeye yeni sınıf (class), form ve başka ögeler eklenebilir. Veri tabanı dosyası, resim, müzik, video dosyaları projeye dâhil edilebilir.

 Form1.cs Form1.Designer.cs

Projeve eklenen her form, cözüm penceresinde

Porm1.resx seklinde ayrı ayrı liste-

lenmektedir. Listedeki Form1.cs ifadesine fare sağ tuşu ile tıklandığında açılan listeden View Code tıklanır ise Form1'e ait kod ekranı, View Designer tıklanır ise Form1'e ait tasarım ekranı açılır. Doğrudan Form1.cs ifadesine çift tıklanır ise tasarım ekranı açılır.



Görsel 1.17: Çözüm penceresi

Kod veya tasarım ekranını göstermenin diğer yolları şunlardır:

1. Form ekranı üzerinde herhangi bir noktada fare sağ tuşu kullanılarak açılan listede View Code (Kodu Göster) tıklanarak kod ekranı, View Designer (Tasarımı Göster) tıklanarak tasarım ekranı görüntülenir.

 Klavyeden F7 tusuna basıldığında kod ekranı acılır. Klavyeden Shift+F7 tusuna basıldığında tasarım ekranı acılır.

1.4.6. Hata Listesi (Error List)

Kod yazarken, kod derlenirken veya kod calışırken oluşan hataların ve uyarıların listelendiği Görsel 1.19'da görülen paneldir.

 Errors (Hatalar) Bölümü: Çok kritik ve programın çalışmasını engelleyen hatalardır. Örneğin; değişkeni tanımlamadan bir kod bloku içinde kullanmak, kod satırının sonunda noktalı virgül koymamak, kod yazarken açılan parantezin kapatılmaması vb.

Warnings (Uyarılar) Bölümü: Programın çalışmasını engellemeyecek düzeydeki iletilerdir. Örneğin, değişkenin tanımlanıp hiçbir zaman kullanılmaması vb.

Görsel 1.18'de hatalı bir kod bloku verilmiştir.



Görsel 1.18: Hatalı kod bloku







Birinci hata, 28. satırda bulunan kodun sonunda noktalı virgül olmamasıdır. İkinci hata ise 28. satırdaki kodda bulunan "a" isimli değişkenin tipi sayısal olmasına rağmen değişkene metinsel ifade atanmış olmasıdır. Hatalar kritik seviyede olduğu için program çalışmaz. Görsel 1.19'da görüldüğü gibi program derleyicisi bu iki hatayı yakalar ve Error List panelinde listeler.

Entire So	olution		😮 2 Errors	1 0 W	arnings	1 0 of 2 Message	es 🎽	Build + Ir	telliSense		Search Error List 🔎
.9	Code	Description	1		Project		File		Line	Sup	pression State
•	CS1002	; expected			Merhab	aDunya	Form1	.cs	28	Act	ive
0	CS0029	Cannot im 'string' to '	plicitly conver int'	t type	Merhab	aDunya	Form1	. <mark>c</mark> s	28	Act	ive



Adım 1: Form üzerine Görsel 1.20'de görüldüğü gibi bir tane Button nesnesi ekleyiniz.

Adım 2: Özellikler penceresinden formun Text özelliğini "Merhaba Dünya" yapınız.

Adım 3: Özellikler listesinden button1'in Text özelliğini mesaj göster yapınız.

Adım 4: Olaylar listesinden button1 için Click olayı metodunu button1'in üzerine çift tıklayarak oluşturunuz.

Adım 5: Oluşturduğunuz metodun içine MessageBox.Show ("Merhaba Dünya"); kodunu yazınız.

Adım 6: Araç Çubuğunda bulunan 🕨 Başlat 🚽 butonu ile programı çalıştırınız.

Merhaba Dünya	<u> </u>		
mesaj göster			
Görsel 1 2	0: Merhaha Düny	/2	

private void button1_Click(object sender, EventArgs e)	
{ MessageBox Show(``Merhaba Dünva''):"	
}	

Örnek uygulamada en önemli nokta, Click olayının metodunu oluşturmaktır. Click, programcıların en çok kullandığı olaydır ve bu olay için metot oluşturma işlemi en pratik şekilde seçili nesneye çift tıklanarak gerçekleştirilir. Uygulamada diğer olaylar için metot tanımlama işlemi istenilseydi Görsel 1.15'teki gibi button 1'e ait olayların listesinden istenilen olayın adının hemen yanındaki boş kutucuğa çift tıklanarak metot oluşturulurdu.

 Not: MessageBox adından da anlaşılacağı gibi ekrana mesaj verdiren sınıfın adıdır. Programcılar tarafından çok fazla kullanılır. Farklı kullanım şekilleri vardır. En temel kullanımı, örnek kodda verildiği gibi sadece tek bir mesajı gösterme şeklindedir. MessageBox sınıfının bazı kullanım şekilleri aşağıda verilmiştir.

- 1. MessageBox.Show("mesaj metni","mesaj başlığı");
- 2. MessageBox.Show("mesaj metni","mesaj başlığı",MessageBoxButtons.YesNoCancel);
- 3. MessageBox.Show("mesaj metni","mesaj başlığı",MessageBoxButtons.OKCancel);





MessageBox sınıfının farklı kullanım şekilleri neler olabilir? Araştırıp bulduklarınızı program içinde kullanınız ve arkadaşlarınızla paylaşınız.

|--|

Jygulama-5

Adım 1: Form üzerine Görsel 1.21'de görüldüğü gibi birer tane Button ve TextBox nesnesi ekleyiniz. **Adım 2:** Button1'e tıklandığında textBox1 nesnesinin içine Hello World yazdırınız.



Görsel 1.21: Hello World mesaj uygulaması



Yukarıdaki örnekte mesaj, bir önceki uygulamadan farklı olarak TextBox nesnesi üzerinden verilmiştir. **textBox1.Text="Hello World";** kod parçasına bakılırsa nesnenin bir özelliğine kod ile müdahale edildiği görülür. Nesnelerin özellikleri sadece Properties panelinden değiştirilmez. Programın çalışma zamanında kodla da değiştirilebilir. Örnek kodlar aşağıda verilmiştir.

textBox1.ForeColor = Color.Red; //Yazı rengini kırmızı yapar.
textBox1.Enabled = false; //Nesneyi pasifleştirir. Artık metin girişi yapılamaz.
textBox1.Visible = false; //Nesneyi görünmez hâle getirir.
textBox1.Font = new Font("Broadway", 16); //Yazı tipi ve boyutu değişir.

Uygulama-6		• \$2 • •		KodSor.ph
 Adım 1: Form üzerine Görsel 1.22'de görüldüğü gibi iki tane GroupBox nesnesi ekleyiniz. Adım 2: Üst tarafta bulunan GroupBox nesnesinin içine bir tane Label nesnesi ekleyiniz. Adım 3: Alt tarafta bulunan GroupBox nesnesinin içine on tane Button nesnesi ekleyiniz ve bunları numaralandırınız. Adım 4: Numaraların yazılı olduğu Button nesnelerinin arka plan rengini "MenuHighlight" yapınız. Adım 5: Button nesnelerinin ve Label nesnesinin yazı tipi stilini "kalın", yazı boyutunu "10" yapınız. Adım 6: Form nesnesinin başlığını "0-9" ve Form nesnesinin arka plan rengini "SandyBrown" yapınız. Adım 7:Tıklanan Button nesnesine ait sayıyı Label nesnesinin Text özelliğine aktaran programı yazınız. 	Bran 5 Tuştakım 1 4 7	2 3 5 6 8 9		hp?KOD=21070
	Görsel	1.22: Tuş ta	akımı	





```
private void button1_Click(object sender, EventArgs e)
             label1.Text = 1'';
     {
                                     ł
private void button2_Click(object sender, EventArgs e)
             label1.Text = 2'';
     {
                                     }
private void button3_Click(object sender, EventArgs e)
             label1.Text = "3";
     ł
                                     ł
private void button4_Click(object sender, EventArgs e)
             label1.Text = 4'';
     Ł
                                     }
private void button5_Click(object sender, EventArgs e)
             label1.Text = 5'';
     Ł
private void button6_Click(object sender, EventArgs e)
             label1.Text = "6";
                                     }
private void button7_Click(object sender, EventArgs e)
              label1.Text = 7'';
     {
private void button8_Click(object sender, EventArgs e)
             label1.Text = "8";
     {
                                     }
private void button9_Click(object sender, EventArgs e)
             label1.Text = "9";
     {
                                     }
private void button10_Click(object sender, EventArgs e)
             label1.Text = 0'';
     {
```

ÖĞRENME BİRİMİ

1. Form üzerine 2 adet **Button** nesnesi ve 1 adet **PictureBox** nesnesi ekleyiniz.

2. Button1'e tıklandığında PictureBox nesnesini görünmez hâle getiren, button2'ye tıklandığında PictureBox nesnesini görünür hâle getiren programı yazınız.

Uygulama-7

Sepete Ekle butonuna tıklandığında TextBox nesnesindeki değeri ListBox nesnesine aktaran Görsel 1.23'teki gibi bir tasarıma sahip programı yazınız.



Görsel 1.23: Manav sepeti uygulaması





ÖĞRENME BİRİMİ

1. Görsel 1.23'teki program tasarımına Sepeti Temizle butonu ekleyiniz.

2. Yeni eklediğiniz butona tıklandığında ListBox nesnesinin içindekileri tamamen temizleyen programı yazınız.

Sıra Sizde

3. Form üzerine 2 adet Button nesnesi ekleyiniz.

4. Button1'e tıklandığında **"Button1'e tıkladınız."**, button2'ye tıklandığında **"Button2'ye tıkladınız."** mesajlarını verdiriniz.

5. Oluşturduğunuz kodları aşağıdaki kutucuğa yazınız.

Sıra Sizde >

- 1. Form üzerine 3 adet Button nesnesi ekleyiniz.
- 2. Form nesnesinin arka plan rengini turuncu yapınız.
- **3.** Button'ların hepsinin arka plan ve yazı renklerini birbiriyle uyumlu olacak şekilde tasarlayınız.

4.

- Mouse simgesi ile button1'in üzerine gelindiğinde "Mouse benim üzerimdedir.",
- mouse simgesi ile button2'nin üzerine çift tıklandığında "Mouse iki kere tıklandı.",
- button3'ün üzerinden mouse simgesi kaldırıldığında **"Mouse üzerinde değil."**

mesajlarını verdiginiz ve oluşturduğunuz kodları aşağıdaki kutucuğa yazınız.

Not: Bir nesne için sadece bir tane olay metodu oluşturulmaz. Nesne için tanımlanabilecek ne kadar olay varsa o kadar da olay metodu oluşturulabilir.

Uygulama-8

Mouse simgesi ile forma eklenen bir Button nesnesinin üzerine gelindiğinde **"Mouse şimdi üzerimdedir."** mesajını, mouse simgesi Button nesnesinin üzerinden kaldırıldığında **"Mouse artık üzerimde değildir."** mesajını verdiriniz.

private void button1_MouseMove(object sender, EventArgs e)
{
 MessageBox.Show ("Mouse Üzerimdedir.");
}
private void button1_MouseLeave(object sender, EventArgs e)
{
 MessageBox.Show ("Mouse Artık Üzerimde Değildir.");
}







Bir Button nesnesine fare ile tıklandığında kaç farklı olay gerçekleşir? Araştırınız, gerçekleşen olayları not ediniz ve notlarınızı arkadaşlarınızla paylaşınız.

Sıra Sizde

Bir TextBox nesnesine yazı yazıldığı andan itibaren kaç farklı olay gerçekleşir? Oluşturacağınız küçük gruplarla araştırınız, sonuçlarınızı diğer gruplarla karşılaştırınız.

Uygulama-9		

Adım 1: Form üzerine Görsel 1.24'te görüldüğü gibi GroupBox nesnesi içine 4 adet Button nesnesi ekleyiniz.

Adım 2: Form nesnesinin arka plan rengini beyaz yapınız.

Adım 3: Kırmızı yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini kırmızı, yeşil yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini yeşil, mavi yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini mavi, gri yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini mavi, gri yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini mavi, gri yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini mavi, gri yazılı Button nesnesine tıklandığında Form nesnesinin arka plan rengini gri yapan programı yazınız.







Uygulama-10

Adım 1: Form üzerine Görsel 1.25'te görüldüğü gibi dört tane PictureBox nesnesini üst üste olacak şekilde yerleştiriniz.

Adım 2: PictureBox nesnelerinin "Visible" özelliğini "false" yapınız (Visible özelliği ile nesnenin görünürlüğü belirlenir.).

Adım 3: Dört tane RadioButton nesnesi ekleyiniz ve bunları numaralandırınız.

Adım 4: radioButton1 nesnesi işaretli ise sadece pictureBox1 nesnesini görünür hâle getirecek, radioButton2 nesnesi işaretli ise sadece pictureBox2 nesnesini görünür hâle getirecek, radioButton3 nesnesi işaretli ise sadece pictureBox3 nesnesini görünür hâle getirecek, radioButton4 nesnesi işaretli ise sadece pictureBox4 nesnesini görünür hâle getirecektir.







Görsel 1.25: Resim galerisi

private void radioButton1_CheckedChanged(object sender, EventArgs e)

{ pictureBox1.Visible = radioButton1.Checked; }

private void radioButton2_CheckedChanged(object sender, EventArgs e)

{ pictureBox2.Visible = radioButton2.Checked; }

private void radioButton3_CheckedChanged(object sender, EventArgs e)

{ pictureBox3.Visible = radioButton3.Checked; }

private void radioButton4_CheckedChanged(object sender, EventArgs e)

{ pictureBox4.Visible = radioButton4.Checked; }

Not: RadioButton nesnesi işaretlenince veya RadioButton nesnesinin işareti kaldırılınca CheckedChanged olayı devreye girer.

1.5. İSİM UZAYLARI (NAMESPACE)



Görsel 1.26: İsim uzayları

İsim uzayları; program yazımı esnasında kullanılan metot, sınıf, değişken, sabit gibi yapıları mantıksal olarak kategorize etme sistemidir. İsim uzayları, proje büyüdükçe kod yapısının karmaşıklığını engeller. Örneğin, aile fotoğraflarına bakmak isteyen biri sabit diskindeki dosyaların hepsini tek klasörde tutuyorsa aradığı fotoğrafa ulaşması çok zor ve karmaşık olacaktır. Dosyalarını müzik, video, fotoğraf klasörleri şeklinde düzenleyip hatta fotoğraflar klasörünün içine de aile, okul, doğa diye ayrı ayrı kategorilerde klasörler oluşturursa karmaşa ortadan kalkacaktır (Görsel 1.26).





ÇALIŞMA ORTAMI VE TEMEL İŞLEMLER

.NET Framework ile gelen standart isim uzayları kullanılabileceği gibi proje yazımı esnasında sonradan oluşturulan isim uzayları da kullanılabilir.

İsim uzayını projeye dâhil etmek için **using** kodu kullanılır. Varsayılan olarak **using** System; kod parçacığı projelerde bulunur. Bu kod parçacığındaki using ifadesi "kullanılıyor" anlamına gelir, System ifadesi ise kullanılan isim uzayını temsil eder.

using System.IO; using System.Data; using System.Windows.Forms;

ÖĞRENME BİRİMİ

Kodlara yukarıdaki gibi isim uzayları eklenebilir.

Not: Bu kütüphanelerin hepsi System içinde mevcuttur. Bu kütüphaneler eklendiğinde kodlar daha kısa yazılır. Örneğin "System.IO.File.WriteAllText" diye bir kod kullanılacaksa using System.IO;" isim uzayı eklendiği için kodu sadece "File.WriteAllText" şeklinde yazmak yeterli olacaktır.

Programda kişisel bir isim uzayı tanımlanarak kodların organize olması sağlanabilir. Veri tabanına kayıt yapma işlemlerini içeren bir kütüphane geliştirilirken Kaydet isimli bir sınıf var ise bu kodlara ait isim uzayını tanımlamak için "namespace" anahtar sözcüğünün kullanımı aşağıdaki örnekte verilmiştir.

```
namespace VeriTabani {
    public class Kaydet
    {
        //kodlarınız
    }
}
```

Yukarıdaki Kaydet sınıfını kullanabilmek için programa using ifadesi ile isim uzayı eklenmelidir. Bu işlem aşağıda verilmiştir.

using VeriTabani

Not: İki tane sağa yatık **slash** (//)karakteri konulduğunda o satırdaki kodlar C# derleyicisi tarafından dikkate alınmaz ve kodların programa hiçbir etkisi olmaz. Slash işaretleri ile programın kodları içine kendinize ait notlar yazabilirsiniz. Böylece kodların ne anlama geldiğini, ne işlem yaptığını hatırlamanız kolaylaşır.

Sıra Sizde >

Siz de isim uzayı yazma ve projeye dâhil etme kurallarına dikkat ederek kendi isim uzaylarınızı tanımlayıp projenize ekleyiniz.

1.6. DEĞİŞKENLER VE TEMEL VERİ TÜRLERİ

Değişkenler, programlamanın en temel kavramıdır. Program çalışması sırasında çeşitli türlerde verileri hafızada saklayan değişkenler, ihtiyaca göre tekrar tekrar kullanılan veri tutuculardır. Program çalıştığı sürece değişkenler RAM bellekte bulunur, program durdurulduğunda RAM bellekten silinir.





CALIŞMA ORTAMI VE TEMEL İŞLEMLER





Görsel 1.27: Değişkenler

Değişkenler; aynı anda içinde sadece tek ürün taşınabilen, farklı taşıma kapasitelerine sahip ve üzerlerinde seri numaraları yazılı **alışveriş sepetleri** gibi düşünülebilir (Görsel 1.27). Bu sepetlerden seçilen herhangi biri ile ihtiyaca göre ürün taşınabilir. Taşınacak ürünün boyutuna ve ağırlığına en uygun sepet seçimi yapılmalıdır. Büyük ebatta kutuya sahip bir ürün taşınacaksa büyük bir sepet seçilmeli, küçük ebatlı kutuya sahip bir ürün taşınacaksa çok büyük bir sepet seçilmemeli ki en verimli şekilde taşıma işlemi yapılabilsin. Aynı seri numaralı sepetle gün içinde farklı ürünler de taşınabilir. Sepetin içeriği her taşımada sürekli değişebilir. Örneğin bir saat önce SP00052 seri numaralı sepete bakıldığında sepetin içinde parfüm olduğu görülüyorken on dakika önce muz, şimdi ekmek, sonrasında saatlerce sepetin boş bir şekilde kaldığı da görülebilir.

Yukarıdaki örnekte sepetin boy ve ağırlık kapasitesi, **değişkenin veri türü**; "SP00052" ifadesi, **değişkenin ismi;** sepetin içindeki parfüm ise **değişkenin değeri** olarak düşünülebilir.

1.6.1. Temel Veri Türleri

Değişkenlerin içine aktarılacak verilerin türleri çeşitlilik gösterebilir. Her veri türünün RAM bellekte kapladığı alan ve değişkenin türüne göre verinin değer aralığı farklıdır. Tablo 1.1, Tablo 1.2, Tablo 1.3 ve Tablo 1.4'te değişkenlerin veri türlerine göre kapladığı alan ve değer aralığı verilmiştir.

Değişkenler, RAM bellekte yer kaplar ve programda kullanılan değişken sayısı arttıkça bu durum RAM belleğin kullanılabilir hafıza kapasitesini düşürür. Örneğin öğrenci notlarının girileceği bir değişkenin veri türü, değer aralığı 100 sayısına en yakın olan **byte** veya **sbyte** olarak tanımlanabilir. Değer aralığı 100 sayısından çok daha fazla olan **short**, **ushort** gibi veri türlerinde tanımlanmamalıdır. Bu, 1 kişinin 100 odalı bir evde yaşamayı seçmesi gibi çok verimsiz, masraflı ve gereksiz bir eylemdir.

Tam Sayı Türü	Kapladığı Alan	Değer Aralığı
byte	1 Bayt	0,,255
sbyte	1 Bayt	1 Bayt -128,,127
short	2 Bayt	-32768,,32767
ushort	2 Bayt	0,65535
int	4 Bayt	-2147483648,,2147483647
uint	4 Bayt	0,,4294967295
long	8 Bayt	-9223372036854775808
ulong	8 Bayt	0,,18446744073709551615

Tablo 1.1: Tam Sayı Veri Türleri







Ondalık Sayı Türü	Kapladığı Alan	Değer Aralığı
float	4 Bayt	±1.5*10-45,,±3.4*1038
double	8 Bayt	±5.0*10-324,,±1.7*10308
decimal	16 Bayt	±1.5*10-28,,±7.9*1028

Tablo 1.3: Metinsel Veri Türleri

Metinsel Veri Türü	Kapladığı Alan	Değer Aralığı
string	Sinirsiz	Metinsel ifade tutar.
char	2 Bayt	Tek bir karakter tutar.

Tablo 1.4: Mantıksal Veri Türleri

Mantıksal Veri Türü	Kapladığı Alan	Değer Aralığı	
bool	1 bit	True-False veya 1-0	

1.6.2. Değişken Tanımlama

(Değişkenin Veri Türü) (Değişen Adı);

t sayi
ouble pi
ring soru
nar karakter
ool cevap

Not:	Aynı veri türü	ndeki değişkenl	er, aynı kod s	atırında ve arala	arına virgül konul	larak tanımla-
' nabilir.					-	

Örnek:

ÖĞRENME BİRİMİ

> int sayi,deger,sonuc; string isim1,isim2,soyad;

Değişken isimleri belirli yazım standartlarına göre verilmelidir. Bu standartlara uymak zorunlu değildir fakat kodların okunabilirliği ve rahat anlaşılması açısından çok önemlidir. Programın tüm kodlarında aynı standarda göre değişken isimleri verilmelidir.

Programda en çok tercih edilen yazım standartları; camel case (deve gösterimi), snake case (yılan gösterimi), pascal case'dir.

• **Camel Case:** Bu yazım standardına göre değişkenin ismindeki ilk kelimenin baş harfi küçük, diğer kelimelerin baş harfleri büyük olur.

Örnek:

string kullaniciAdi; int toplamHesapTutari;



ÇALIŞMA ORTAMI VE TEMEL İŞLEMLER

• **Snake Case:** Bu yazım standardına göre değişkenin ismindeki kelimelerin arasına _ (alt çizgi) kullanılır.

Örnek:

ÖĞRENME BİRİMİ

string kullanici_adi; int toplam_hesap_tutari; string boy_Uzunlugu; string Urun_Fiyati;

Pascal Case: Bu yazım standardına göre değişkenin ismindeki tüm kelimelerin baş harfleri büyük olur.

Örnek:

string KullaniciAdi; int ToplamHesapTutari; string BoyUzunlugu; string UrunFiyati;

1.6.3. Değişkene Değer Atama

(Değişkenin Adı) = (Değişkenin Değeri);

sayi=52
PiSayisi = 3.14159
Soru="Ülkemizde hangi deniz en kuzeydedir?"
karakter='A'
cevap=true
adi_soyadi="Ali YOLCU";
sinavNotOrtalamasi=76

Not: Değişken tanımlanırken de değer atama yapılabilir.

Örnek: int sayi=52;

1.6.4. Değişken İsimlendirme Kuralları

- 1. Değişken isimlerinde boşluk kullanılmaz. Boşluk yerine alt çizgi (_) kullanılabilir.
- Örnek: string ad soyad; şeklinde kullanım yanlıştır.

string ad_soyad; veya string adsoyad; şeklinde kullanım doğrudur.

2. ?, !, :, %, +, -, . gibi özel karakterler kullanılmaz.

Örnek: string soru?; şeklinde kullanım yanlıştır.

string soru; şeklinde kullanım doğrudur.

3. Değişken isimleri sayı ile başlamaz.

Örnek: byte 1not; şeklinde kullanım yanlıştır.

byte not1; şeklinde kullanım doğrudur.

4. Değişken isimleri büyük ve küçük harfe duyarlıdır.

Örnek:

ulong toplamTutar; şeklinde tanımlanan değişken ile ulong toplamtutar; veya ulong ToplamTUTAR şeklinde tanımlanan değişken aynı değildir.

5. Herhangi bir kodla aynı isimde değişken tanımlanamaz. Değişkenlerde if, else, random gibi programa ait ifadeler isim olarak kullanılmaz.

6. Zorunluluk yoktur fakat Türkçe karakterlerin (ç, ö, ü, ğ, ş vb.) kullanılması tavsiye edilmez.







Aşağıdaki değişken tanımlamalarını doğru ya da yanlış olarak değerlendiriniz. Yanlış olan değişken tanımlamalarının sebebini açıklamalar bölümüne yazınız.

Değişken Adı	Doğru	Yanlış	Açıklamalar
Musteri_Tc			
Müsteri_adı			
1.isim			
toplam			
textBox			
Yüzde%18			
seri no			
Yuzde10			

1.6.5. Değişken Veri Türü Dönüştürme (Convert) İşlemleri

Değişkenlerin veri türlerini bazen değiştirmek gerekebilir. Sayısal bir ifade, bir nesnenin Text özelliğine aktarılmak istendiğinde program hata verecektir. İçeriği tamamen sayı olsa da metinsel bir ifadeyi sayısal veri türüne sahip bir değişkene aktarırken program yine hata verecektir. Bu tip durumlarda değişkenlerin veri türlerini dönüştürmek gerekir. Tür dönüşümünü sağlayacak hazır metotlar şunlardır:

ToString() >>>> Her türden değişkeni string türüne dönüştürür. ToString(), en sık kullanılan dönüştürme metodudur.

>>>>>	Byte'a çevirir.
>>>>>	Short'a çevirir.
>>>>>	Int'e çevirir.
>>>>>	Long'a çevirir.
>>>>>	Float'a çevirir.
>>>>>	Double'a çevirir.
>>>>>	Decimal'a çevirir
>>>>>	Char'a çevirir.
>>>>>	Bool'a çevirir.
	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Örnek:

int sayi1=100; string deger; deger=sayi1; //Program bu noktada convert type `int' to `string' seklinde hata verecektir.

Örnek:

```
int sayi1=100;
string deger;
deger=sayi1.ToString(); //Program artık hata vermeyecektir çünkü veri türleri uyumludur.
```

Uygulama-11

Adım 1: Form üzerine Görsel 1.28'de görüldüğü gibi bir CheckBox nesnesi ekleyiniz. **Adım 2:** İki adet Label nesnesi ekleyiniz.




Adım 3: CheckBox nesnesi işaretli ise label2 nesnesine "True" değerini, CheckBox nesnesi işaretli değil ise label2 nesnesine "False" değerini gösteren programı yazınız.



private	void checkBox1_CheckedChanged(object sender, EventArgs e)
{	
	bool secim;
	secim = checkBox1.Checked; //Checked özelliği True veya False değerleri alır.
	label2.Text = secim.ToString();
}	

Sıra Sizde

Bir değişken tanımlayıp içine 500 sayı değerini atayınız. Butona tıklandığında bu değişkendeki değeri ekrana mesaj olarak verdiriniz.

1.7. ARİTMETİKSEL OPERATÖRLER

Aritmetiksel operatörler, matematiksel işlemlerde kullanılan özel karakterlerdir (Tablo 1.5).

Operatör Adı Sembolü Örnek				
Toplama	+	2+8		
Çıkarma	-	8-2		
Çarpma	*	8*2		
Bölme	/	8/2		
Mod alma	%	8%2		

Tablo 1.5: Aritmetiksel Operatörler

Not: Bir sayı başka bir sayıya bölündüğünde kalan sayıya **mod** denir. **Örnek:** 8%2=0 iken 8%5=3 sonucunu verir.







Not: TextBox nesnesinin içine sayısal bir değer girilmiş olsa bile matematiksel işlemler yapılamaz çünkü TextBox nesnesi metinsel veri türüne sahip değerler alabilen bir nesnedir. Bu tip durum larda Convert kodu ile veri türü dönüşümü yapılmalıdır. Ayrıca TextBox nesnesine sayısal veri türünde bir değer yazmak için ToString() metodu ile veri türü dönüşümü yapılmalıdır.



Aşağıdaki örnek kodları çalıştırınız.

1.Örnek Kodlar	2.Örnek Kodlar
string ad, soyad,topla;	
ad = "Zeynep";	textBox1.Text="25";
soyad = "Sare";	textBox2.Text="2"
topla = ad +" "+ soyad;	textBox3.Text= textBox1.Text + textBox2.Text
MessageBox.Show(topla);	

Not: Programlama dillerinde toplama operatörü sadece sayıları toplamak için kullanılmaz. Metinsel veri türüne sahip değişkenlerin arasına artı (+) operatörü konularak string değerlerini birleştirmek için de toplama operatörü kullanılır. Yukarıdaki 1. örnek kodlar çalıştığında ekrana "Zeynep Sare" mesajı gelecektir. 2. örnek kodlar çalıştığında ise textBox3'ün değeri "252" olur.

Sıra Sizde >

Girilen iki sayının toplama, çıkarma, çarpma ve bölme işlemlerini yapan programı tasarlayıp yazınız.





Sıra Sizde

Button nesnesine çift tıklandığında TextBox nesnesine girilen sayının karesini alıp mesaj verdiren programı Görsel 1.30'da görüldüğü gibi tasarlayıp yazınız.



1.7.1. İşlem Önceliği

Matematiksel bir ifade yazılan kod satırında birden fazla aritmetiksel operatör kullanılabilir. Böyle durumlarda sonuçların doğru çıkması için operatörler arasındaki işlem önceliklerine dikkat edilerek kod yazılmalıdır.

Kodlar çalıştırılırken öncelikle varsa parantez içindeki işlemler, daha sonra parantez dışındaki işlemler yapılır. Çarpma, bölme ve mod alma işlemleri kendi aralarında, toplama ve çıkarma işlemleri de kendi aralarında aynı önceliğe sahiptir. Kendi aralarında aynı önceliğe sahip operatörlerin olduğu ifadelerde ise sırasıyla soldan sağa doğru işlem yapılır.

Örnek: 4*3+15/3-1 işlemini yapınız.

Çözüm: =12+5-1 -->Önce çarpma sonra bölme işlemi yapılır.

=17-1 -->Toplama işlemi ve sonrasında çıkarma işlemi yapılır.

=16

Örnek: (3+4-2)*(10/5*2)-4 işlemini yapınız.

Çözüm:

=5*4-4 -->Önce parantez içindeki işlemler yapılır.

=20-4 -->Çarpma işlemi toplama işleminden öncelikli olarak yapılır.

=16

Not: Kod yazımı sırasında karmaşayı ortadan kaldırmak ve en doğru sonuca ulaşmak için öncelikli olan ifadeleri parantez içine alabilirsiniz.

Sıra Sizde

Siz de ikili gruplar hâlinde içinde aritmetiksel operatörler geçen 10 adet ifade yazınız ve kendi sonuçlarınızla grup arkadaşınızın sonuçlarını karşılaştırınız.

Sıra Sizde >

Aşağıdaki işlemlerin sonucunu yazınız.

7*5-3	(5*4)-(3+9)	
20+30-4*10	(5*8/4)+3-5*2	
9/3+2*2-5+1		











İki sayıyı toplayıp mesaj verdiren programı yazınız (Değişkenlere istenilen değerler atanabilir.).

```
private void button1_Click(object sender, EventArgs e)
{
  int sayi1;
  int sayi2;
  int toplam;
  sayi1 = 5;
  sayi2 = 20;
  toplam = sayi1 + sayi2;
  MessageBox.Show(toplam.ToString());
```

Sıra Sizde >

İki sayıyı toplayıp 2 ile çarparak çıkan sonuca 5 ekleyen ve oluşan değeri mesaj verdiren programı işlem önceliklerine dikkat ederek yazınız (Değişkenlere istenilen değerler atanabilir.).



nesnesi ekleyiniz.

Adım 2: Dört adet Button nesnesi ekleyiniz.

Adım 3: Button nesnelerinin arka plan renklerini değiştiriniz.

Adım 4: İndirim oranlarını Button nesnelerinin üzerine yazınız.

Adım 5: TextBox nesnesine etiket fiyatı girilen ürünün seçilen indirim oranına göre indirimli fiyatını bulup Label nesnesinde gösteren programı yazınız.

İndirimli Ürün			
Etiket Fiya	tu: 52		
Yüzde 10 İNDİRİM	Yüzde 25 İNDİRİM	Yüzde 50 İNDİRİM	Yüzde 75 İNDİRİM

Indirimli Fiyatı: 46,8

Görsel 1.31: İndirimli ürün





Not: "etiketFiyati - etiketFiyati * 0.10" ifadesinde işlem önceliği çarpmada olduğu için önce çarpma işlemi sonra çıkarma işlemi yapılacaktır. Aynı ifade "etiketFiyati - (etiketFiyati * 0.10)" şeklinde yazılarak da işlem yapılabilir.

ÖĞRENME







Adım 1: Form üzerine Görsel 1.32'de görüldüğü gibi 1 adet GroupBox, 1 adet Button, 5 adet Label, 6 adet TextBox nesnesi ekleyiniz.

Adım 2: Türkçe ve Matematik netlerinin gösterileceği TextBox nesnelerinin Enabled özelliğini bilgi girişini engellemek için false yapınız.

Adım 3: Button nesnesi tıklandığında ilgili derslere ait net sayısını hesaplayan programı yazınız (Net sayısı, doğru sayısından yanlış sayısının dörtte biri çıkarılarak hesaplanır.).



Görsel 1.32: Net hesaplama





É

ÖLÇME VE DEĞERLENDİRME - 1

1. Aşağıdaki cümlelerde parantez içine yargılar doğru ise (D), yanlış ise (Y) yazınız.

- () Kod editörü platformu kullanılarak sadece C# programlama dili kodlanabilir.
- () string veri türüne sahip bir değişken int veri türüne dönüştürülebilir.
- () int a="52"; hatalı bir koddur.

ÖĞRENME BİRİMİ

- () string a="16"; hatalı bir koddur.
- () Değişken isimleri sayı ile başlamaz.
- () Bir nesne için aynı anda birden fazla olay metodu oluşturulabilir.
- () Button nesnesi için sadece Click olay metodu oluşturulabilir.

2. Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

- İsim uzayını projeye dâhil etmek için kodu kullanılır.
- Form üzerine eklenen nesneler panelinden seçilir.
- Mod alma işlemi için aritmetiksel operatörü kullanılır.
- Sayısal veri türüne sahip bir değişken metodu ile string veri türüne dönüştürülür.

3. Aşağıdaki kod blokunda boş bırakılan yerlere doğru sözcükleri yazınız.

using; ResimEkle { public class Kaydet { //kodlarınız }

4. Aşağıdaki kod satırlarından hangisi <u>çalışmaz</u>? Kodun çalışmama sebebini yazınız.

int öğrenci no;	
int 1.not;	
string isim;	
double yaricap;	
bool cinsiyet;	
int sayi1=13.4;	
<pre>char karakter = "kmr";</pre>	
string tc = "00334412376";	





	• ••	
5. Aşağıdaki kodlar ça	lıştırıldığında ekranda görülecek	k sayı kaçtır?
double a, b, c, d, islem;		
a = 1;		
b = 6;		
c = 5;		
d = 2;		
islem = $a * b / c + d;$		
MessageBox.Show(islem	.ToString());	
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	ayan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	ayan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.
7. Yarıçapı girilen daiı	renin alanını ve çevresini hesapl	layan programı yazınız.



10. SINIF

-

•

•



NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Karşılaştırma operatörlerini sıralayacak,
- Karar ifadelerinin kullanım mantığını kavrayacak,
- if, if-else, else if ifadelerini kullanacak,
- İç içe karar ifadelerini kullanacak,
- Mantıksal operatörleri sıralayacak,
- Döngü çalışma mantığını kavrayacak,
- For, while, do-while döngülerini kullanacak,
- Hata ayıklama bloklarının kullanımını öğreneceksiniz.

ANAHTAR KELİMELER

Karar ifadeleri, döngü, if, else, for, while, hata ayıklama



HAZIRLIK ÇALIŞMALARI

1. Karar ifadelerinin neler olabileceğini arkadaşlarınızla tartışınız ve günlük hayattaki kullanım alanlarına örnekler veriniz.

2. Program yazarken döngü yapılarının sağladığı kolaylıklar neler olabilir? Açıklayınız.

2.1. KARAR İFADELERİ



Görsel 2.1: Karar ifadeleri

Karar ifadeleri; tüm programlama dillerinde bulunan, koşula veya koşullara bağlı bir şekilde programın nasıl ilerleyeceğini belirleyen yapılardır. Programın yol ayrımı noktalarını karar ifadeleri temsil eder. Bu yapı sayesinde programlar daha dinamik ve etkileşimlidir.

Karar ifadeleri, programlamanın vazgeçilmez yapılarıdır. Örneğin; profil girişi için gerekli olan **kullanıcı adı ve şifre** bilgileri, karar ifadeleri kullanılarak kontrol edilir ve program girişe izin verir ya da izin vermez. Bu durum, programın ilerleyeceği yönü belirler (Görsel 2.1). TextBox nesnesine girilebilecek en büyük sayının 100 olması isteniyorsa karar ifadeleri ile TextBox nesnesinin değerinin kontrol edilmesi ve veri girişinin sağlanması da bu konuya bir başka örnektir.

2.1.1. Karşılaştırma Operatörleri

Karşılaştırma operatörleri, karar ifadelerinde iki değeri birbiriyle karşılaştırmak için kullanılan operatörlerdir (Tablo 2.1).

Operatör	Anlamı	
<	Küçükse	
>	Büyükse	
==	Eşitse	
<=	Küçük veya Eşitse	
>=	Büyük veya Eşitse	
!=	Eşit Değilse	

Tablo 2.1:	Karşılaştırma	Operatörleri
------------	---------------	--------------





2.1.2. if Yapısı

if, kelime olarak eğer anlamına gelmektedir. Sadece şart sağlandığında çalışması istenen kodlar için kullanılır. Şart ifadesi sağlandığında true, sağlanmadığında false değeri oluşur. if(true) olduğunda if yapısına bağlı kodlar çalışır, if(false) olduğunda kodlar çalışmaz.

if(şart ifadesi) { // Sart ifadoci sağlanıv

// Şart ifadesi sağlanıyorsa çalışacak kodlarınız

Uygulama-1

Adım 1: Form üzerine Button nesnesi ekleyiniz. **Adım 2:** Button nesnesinin Click olay metoduna aşağıdaki kodları yazınız.

```
byte skor1,skor2;
skor1 = 4;
skor2 = 1;
if (skor1 > skor2)
{
    MessageBox.Show("1. Takım Kazandı.");
}
```

•Not: Yukarıdaki uygulamada (skor1 > skor2) şart ifadesidir. Şart ifadesi sağlandığında çalışması istenilen kodlar küme parantezlerinin içine yazılır. Şart sağlandığı için parantez içindeki kod çalışacaktır. Şart sağlanmadığında parantez içindeki kod çalışmayacak, parantezden sonra hangi kod varsa onlar işletilecektir.

Uygulama-2

Adım 1: Form üzerine Button ve TextBox nesnesi ekleyiniz.

Adım 2: Button nesnesi tıklandığında TextBox nesnesine girilen sayı 17'den büyük ise "Ehliyet başvurusunda bulunabilirsiniz." şeklinde mesaj veren programı yazınız.



Not: Yukarıdaki kod blokunda şart ifadesinde kullanılan "yas" değişkeni yerine "Convert.ToByte(textBox1.Text)" ifadesi de kullanılabilir.



≽ Uygulama-3

Adım 1: Form üzerine 2 adet Label, 2 adet TextBox ve 1 adet Button nesnesi ekleyiniz. **Adım 2:** Programı Görsel 2.2'de görüldüğü gibi tasarlayınız.

Adım 3: Button nesnesi tıklandığında textBox1'deki değer ile textBox2'deki değeri büyüklük, küçüklük ve eşitlik bakımından karşılaştıran ve sonucu ekrana mesaj olarak veren programı yazınız.



Görsel 2.2: Sayıları karşılaştırma uygulaması

```
private void button1_Click(object sender, EventArgs e)
     {
        byte sayi1, sayi2;
        sayi1 = Convert.ToByte(textBox1.Text);
        sayi2 = Convert.ToByte(textBox2.Text);
        if (sayi1 > sayi2)
        {
           MessageBox.Show("1.sayı 2.sayıdan büyüktür.");
        }
        if (sayi1 == sayi2)
        {
           MessageBox.Show("Sayılar birbirine eşittir.");
        }
        if (sayi1 < sayi2)
        {
           MessageBox.Show("2.sayı 1.sayıdan büyüktür.");
        }
      }
```

Uygulama-4

Adım 1: Form üzerine Button ve TextBox nesnesi ekleyiniz.

Adım 2: Button nesnesi içine tıklandığında TextBox nesnesine girilen sayının tek mi, çift mi olduğunu bulan programı yazınız.

```
int sayi;
sayi =Convert.ToInt32(textBox1.Text);
if(sayi % 2 == 0)
{
    MessageBox.Show("Bu bir çift sayıdır.");
}
if(sayi % 2 ==1)
{
    MessageBox.Show("Bu bir tek sayıdır.");
}
```





Not: Bir sayının 2'ye bölümünden kalan sayı 0 ise bölünen sayı çifttir, kalan sayı 1 ise bölünen sayı tektir.

2.1.3. if-else Yapısı

else; kelime olarak **değilse, aksi durumda** anlamına gelmektedir. Şart ifadesi sağlandığında if kod bloku içindeki kodlar çalışır, şart ifadesi sağlanmadığında ise else kod bloku içindeki kodlar çalışır.





Uygulama-5

Adım 1: Form üzerine TextBox ve Button nesnesi ekleyiniz.

Adım 2: TextBox nesnesine girilen kullanıcı adı milliegitim@meb.k12.tr'ye eşit ise "Kullanıcı sisteme kayıtlıdır." mesajını, kullanıcı adı milliegitim@meb.k12.tr'ye eşit değil ise "Kullanıcı adınız yanlıştır." mesajını veren programı yazınız.

string kullanici_adi; kullanici_adi=textBox1.Text; if (kullanici_adi == "milliegitim@meb.k12.tr.com") { MessageBox.Show("Kullanıcı sisteme kayıtlıdır."); } else { MessageBox.Show("Kullanıcı adınız yanlıştır."); }

Not: Yukarıdaki kod blokunda (**kullanici_adi == "milliegitim@meb.k12.tr"**) şart ifadesi sağlanmadığında **else** kod blokundaki kodlar çalışır.

Sıra Sizde

TextBox nesnesine girilen sayının tek mi, çift mi olduğunu bulan programı if-else yapısını kullanarak yazınız.



•		KARAR VE DÖNGÜ YAPILARI
	ÖĞRENME BİRİMİ	

Uygulama-6

Adım 1: Form üzerine 2 adet GroupBox, 2 adet CheckBox, 1 adet Button ve 1 adet ListBox nesnesi ekleyiniz.

Adım 2: Form ve GroupBox nesnelerinin renklerini beyaz yapınız.

Adım 3: Button nesnesine tıklandığında "Lamba Aç/Kapa" işaretli ise "Lambalar Açık" mesajını, "Lamba Aç/Kapa" işaretli değil ise "Lambalar Kapalı" mesajını, "Kombi Aç/Kapa" işaretli ise "Kombi Açık" mesajını, "Kombi Aç/Kapa" işaretli değil ise "Kombi Kapalı" mesajını ListBox nesnesine ekleyen programı Görsel 2.3'teki gibi tasarlayıp yazınız.

🖳 Akıllı Ev		<u>1000</u> 5	×
Kontrol Paneli	Sistem Raporu		_
☑ Lamba Aç/Kapa ☑ Kombi Aç/Kapa	Lambalar Açık. Kombi Açık.		
Sistemi Kontrol Et			

Görsel 2.3: Akıllı ev

```
if (checkBox1.Checked == true)
{
    listBox1.Items.Add(``Lambalar Açık'');
    }
else
    {
    listBox1.Items.Add(``Lambalar Kapalı'');
    }
    if (checkBox2.Checked == true)
    {
        listBox1.Items.Add(``Kombi Açık'');
    }
    else
    {
        listBox1.Items.Add(``Kombi Kapalı'');
    }
}
```

🕹 Uygulama-7

Adım 1: Form üzerine 2 adet RadioButton, 2 adet TextBox, 4 adet Label, 1 adet Button ve 1 adet GroupBox nesnesi ekleyiniz.

Adım 2: Programı Görsel 2.4'te görüldüğü gibi tasarlayınız.

Adım 3: Button nesnesine tıklanıp RadioButton nesnelerinden "Toplama İşlemi" işaretlendiğinde sayıları toplayan, "Çarpma İşlemi" işaretlendiğinde sayıları çarpan ve sonucu ekrandaki Label nesnesinde gösteren programı if-else yapısını kullanarak yazınız.





Görsel 2.4: Toplama / Çarpma uygulaması

```
private void button1_Click(object sender, EventArgs e)
{
    int deger1 = Convert.ToInt32(textBox1.Text);
    int deger2 = Convert.ToInt32(textBox2.Text);
    int sonuc;
    if (radioButton1.Checked == true)
    {
        label4.Text = (deger1 + deger2).ToString();
    }
    else
    {
        label4.Text = (deger1 - deger2).ToString();
    }
}
```

2.1.4. else if Yapısı

Şartın sağlanmadığı durumlarda else if kullanılarak yeni bir şart ifadesi daha yazılabilir. else if ifadesinden sonra tekrar else if ifadesi veya else ifadesi kullanılabilir.



Uygulama-8 Adım 1: Form üzerine TextBox ve Button nesnesi ekleyiniz. Adım 2: Button nesnesine tıklandığında TextBox nesnesinin içine yüzlük sistemde girilen bir notu beşlik sisteme çeviren programı yazınız.





int sayi; sayi = Convert.ToInt32(textBox1.Text); if (sayi < 0){ MessageBox.Show("0'dan büyük bir sayı giriniz!"); } else if (sayi < 25) { MessageBox.Show("Notunuz 0"); } else if (sayi < 45) { MessageBox.Show("Notunuz 1"); } else if (sayi < 55) { MessageBox.Show("Notunuz 2"); } else if (sayi < 70) ł MessageBox.Show("Notunuz 3"); 2 else if (sayi < 85) ł MessageBox.Show("Notunuz 4"); else if (sayi <= 100) { MessageBox.Show("Notunuz 5"); } else { MessageBox.Show("Hatalı giriş yaptınız!"); }

Sıra Sizde

Hava sıcaklığı 10 derecenin altında ise "Hava soğuk" mesajını, 10-25 derece arasında ise "Hava hafif sıcak" mesajını, 25 derecenin üstünde ise "Hava sıcak" mesajını veren programı yazınız.

2.1.5. İç İçe Şart İfadeleri

İç içe şart ifadeleri, birbirini izleyen birden çok şart ifadesinin kontrolünü gerçekleştirmek için kullanılır.





Adım 3: Kontrol et butonuna tıklandığında not ortalaması girilen kişinin notu 85 veya üzeri ise "Takdir Belgesi Almaya Hak Kazandınız." mesajını, not ortalaması girilen kişinin notu 70 veya üzeri ise "Teşekkür Belgesi Almaya Hak Kazandınız." mesajını, not ortalaması girilen kişinin notu 50'nin altında ise "Sınıf Geçmek İçin Yeterli Not Alamadınız." mesajını veren programı yazınız.

🖳 Belge İşlemleri		
Not Ortalaması:		
	Kontrol et	
Takdir Belgesi Alr	naya Hak K	(azandınız.

Görsel 2.5: Belge işlemleri uygulaması



2.1.6. Switch-Case

Switch-case, bir ifadenin aldığı değere bağlı olarak program için birçok farklı çalışma yolu belirleyen bir komuttur. Switch ifadesindeki değer hangi durumun değeri ile eşleşiyorsa o duruma ait kodlar çalışır. Hiçbir durum ile eşleşme olmaz ise default ifadesinde belirtilen kodlar çalışır. "break;" komutu, kodlar çalıştıktan sonra switch-case ifadesinden çıkmayı sağlar. Switch ifadesi içine yazılan değerin veri türü ile case ifadelerindeki değerin veri türleri aynı olmalıdır. Case ifadelerindeki değerler için değişken kullanılmaz.

SRENME <u>sirimi</u>		
switch (sayısal değer)){	switch (karakter değer)){	switch (metinsel değer)){
case 1:	case 'X':	case "kırmızı":
break:	break:	break:
case 2:	case 'Y':	case "vesil":
//kodlar	//kodlar	//kodlar
break;	break;	break;
case 3:	case 'Z':	case "mavi":
//kodlar	//kodlar	//kodlar
break;	break;	break;
default: //kodlar	default: //kodlar	default: //kodlar
break;	break;	break;
}	}	}

Not: Switch-case ile yapılabilecek tüm işlemler if, if-else, else if yapıları kullanılarak da yapılabilir. Kod karmaşıklığını ortadan kaldırmak için uygun durumlarda switch-case kullanılabilir.

Uygulama-10

Adım 1: Form üzerine Button nesnesi ekleyiniz.

Adım 2: Button nesnesine tıklandığında bilgisayarın tarih bilgisine göre haftanın hangi gününde olduğunuzu bulan programı yazınız.



Not: DateTime.Now.DayOfWeek kodu ile haftanın kaçıncı gününde olunduğu bilgisi elde edilir.



•		KARAR VE DÖNGÜ YAPI	LARI
2.	ÖĞRENME		•
•	DIRIMI	/	\

Uygulama-11

Adım 1: Form üzerine Button nesnesi ekleyiniz.

Adım 2: Button nesnesine tıklandığında bilgisayarın tarih bilgisine göre o günün hafta içi mi, hafta sonu mu olduğunu bulan programı yazınız.

int gun = Convert.ToInt32(DateTime.Now.DayOfWeek);
switch (gun)
{
case 1:
case 2:
case 3:
case 4:
case 5:
MessageBox.Show("Hafta içi");
break;
case 6:
case 7:
MessageBox.Show("Hafta sonu");
break;
default:
MessageBox.Show("Hata oluştu.");
break;
}

Not: Yukarıdaki uygulamada görüldüğü gibi birden fazla durum için aynı kodlar çalıştırılabilir.

Sıra Sizde

ComboBox nesnesi içinden seçilen bir harfin alfabenin kaçıncı harfi olduğunu bulan programı switch-case kullanarak Görsel 2.6'da görüldüğü gibi tasarlayıp yazınız.

Alfabe			Comb	oBox itemleri
Harf Seçiniz:	M	-	A	~
	Kontrol Et		в С С	
M harfi, alfat	penin 16. ha	arfidir.	DEF	
			GĞ	
			J K	
			M	
			0	
	Görsel 2.6: A	lfabe uygu	laması	
	•	55	•	∖ 10 .



Sıra Sizde 🔸

ComboBox nesnesi içinden seçilen bir harfin sesli harf mi, sessiz harf mi olduğunu bulan programı yazınız.

2.2. MANTIKSAL OPERATÖRLER



Görsel 2.7: Mantıksal operatörler

Kod blokları, if yapılarında mantıksal operatörler kullanılmadığı zaman tek bir şart ifadesinin sonucuna bağlı olarak çalışmaktadır. Bazen birden fazla şart ifadesinin kullanılması gerekebilir. Örneğin, kullanıcı doğrulama işlemi için kullanıcı adının ve şifre bilgilerinin aynı şart ifadesinde kontrol edilmesi gerekmektedir (Görsel 2.7). Bu durumda Tablo 2.2'deki mantıksal operatörlere ihtiyaç vardır.

Operatör Adı	Sembolü	Örnek
AND (ve)	&&	(A <b) &&="" (a<c)<="" td=""></b)>
OR (veya)	Π	(A <b) (a<c)<="" td="" =""></b)>
NOT (değil)	!	!(A <b)< td=""></b)<>

Tablo 2.2: Mantıksal Operatörler

Not: İç içe if yapıları kullanılarak birden fazla şart ifadesi kontrol edilebilir fakat bu durum kod yapısını karmaşık hâle getirebilir. Örneğin, 4 tane şart ifadesi için iç içe if yapısı kullanılırsa kodu okumak ve yazmak zorlaşacaktır.

2.2.1. AND (&&) Operatörü

Şart ifadelerinin hepsinin sağlanması gerektiğinde "and" operatörü kullanılır. Şart ifadelerinden herhangi biri sağlanmadığında if yapısına bağlı kod bloku çalışmaz.

```
if(user_name=="admin@meb.k12.tr" && password == "12345")
```

// Kodlarınız }



ekleyiniz.

Adım 2: TrackBar nesnesi için "Scroll" olay metodu oluşturunuz (TrackBar nesnesi kaydırıldığında Scroll olayı tetiklenir.).

Adım 3: TrackBar nesnesinin Minimum özelliğini "0", Maximum özelliğini "15" yapınız.

Adım 4: TrackBar nesnesinin Value özelliği 0 ise siyah renkli yazı ile Label nesnesine "Ses Yok", TrackBar nesnesinin Value özelliği 1 ile 10 arasında ise yesil renkli yazı ile Label nesnesine "Normal Ses Seviyesi", TrackBar nesnesinin Value özelliği 11 ile 15 arasında ise kırmızı renkli yazı ile Label nesnesine "Yüksek Ses Seviyesi" mesajlarını veren programı Görsel 2.8'deki gibi tasarlayıp yazınız (Value özelliği,

TrackBar nesnesinin hangi değere sahip olduğu bilgisini verir.).



Görsel 2.8: Ses ayarı

```
private void trackBar1_Scroll(object sender, EventArgs e)
 { int ses = trackBar1.Value;
    if (ses == 0)
    {
       label1.Text = "Ses Yok";
       label1.ForeColor = Color.Black;
    }
    if (ses > 0 \&\& ses <= 10)
    {
       label1.Text = "Normal Ses Sevivesi";
       label1.ForeColor = Color.Green;
    if (ses >= 11)
    {
       label1.Text = "Yüksek Ses Seviyesi";
       label1.ForeColor = Color.Red;
    }
 }
```





Adım 1: Form üzerine Label ve ComboBox nesnesi ekleyiniz.

Adım 2: ComboBox nesnesinin içine 1 ile 12 arasındaki sayıları ekleyiniz.

Adım 3: ComboBox nesnesi içinden sınıf bilgisi seçildiğinde, seçilen sınıfın hangi eğitim-öğretim kademesine ait olduğunu mesaj veren programı Görsel 2.9'da görüldüğü gibi tasarlayıp yazınız.



Görsel 2.9: Eğitim-öğretim kademeleri uygulaması



Not: ComboBox nesnesinin değeri değiştiğinde kodların çalışması için SelectedValueChanged olay metodu oluşturulur.





Sıra Sizde

TextBox nesnelerine kilo ve boy bilgileri girilen kişinin vücut kütle indeksini hesaplayıp çıkan sonuca göre vücut kütle indeksinin hangi kategoride olduğunu mesaj veren programı Görsel 2.10'da görüldüğü gibi tasarlayıp yazınız.



Görsel 2.10: Vücut kütle indeksi uygulaması

Kategori	Vücut Kütle İndeksi (kg/m²)
İdeal Kilonun Altı	< 18,49
İdeal Kilo	18,5 - 24,99
İdeal Kilonun Üzeri	25 - 29,99
İdeal Kilonun Çok Üzeri	> 30

Tablo 2.3: Vücut Kütle İndeksi

Not: Vücut kütle indeksi; bir kişinin kilogram cinsinden ağırlığının, metre cinsinden boy uzunluğunun karesine bölünmesiyle hesaplanır. Örneğin, ağırlığı 78 kg ve boyu 1.78 m olan biri

için **vücut kütle indeksi=kg/m**² formülünden 23,4 kg/m² olarak hesaplanacaktır. Tablo 2.3'e göre 23,4 kg/m² indeksine sahip kişi ideal kilosundadır.

Sıra Sizde >

TextBox nesnelerine girilen üç sayıdan hangisinin en büyük sayı, hangisinin en küçük sayı olduğunu bulan programı yazınız.





2.2.2. OR(||) Operatörü

Şart ifadelerinin en az bir tanesinin sağlanması gerektiğinde "or" operatörü kullanılır. Şart ifadelerinin hiçbiri sağlanmadığında if yapısına bağlı kod bloku çalışmaz.



Adım 1: Form üzerine 1 adet TextBox, 1 adet Label, 1 adet Button, 1 adet GroupBox ve 5 adet RadioButton nesnesini ekleyiniz.

Adım 2: Form nesnesinin rengini "ActiveCaption" yapınız.

Adım 3: TextBox nesnesine toplam tutarı girilen bir alışverişin RadioButton nesneleri ile işaretlenen ödeme şekli 2 taksit veya 3 taksit ise ödenecek toplam tutara %5 ek fiyat, ödeme şekli 4 taksit veya 5 taksit ise ödenecek toplam tutara %10 ek fiyat ilave eden programı Görsel 2.11'de görüldüğü gibi tasarlayıp yazınız.



Görsel 2.11: Kasa uygulaması

```
private void button1_Click(object sender, EventArgs e)
{
    double tutar;
    tutar = Convert.ToDouble(textBox1.Text);
    if(radioButton2.Checked==true || radioButton3.Checked == true)
    {
        tutar = tutar + (tutar * 0.05);
    }
    if (radioButton4.Checked == true || radioButton5.Checked == true)
    {
        tutar = tutar + (tutar * 0.10);
    }
    MessageBox.Show("Ödenecek Toplam Tutar:" + tutar.ToString() + "TL");
}
```



Adım 1: Form üzerine 1 adet ListBox ve 1 adet Button nesnesini ekleyiniz. Adım 2: ListBox nesnesinin Items özelliğine "Mouse, Yazıcı, Klavye, Hoparlör, Kamera, Tarayıcı, Projeksiyon" değerlerini ekleyiniz.

Adım 3: ListBox nesnesi içinden seçilen bir bilgisayar parçasının giriş donanım birimi mi, çıkış donanım birimi mi olduğunu mesaj veren programı Görsel 2.12'de görüldüğü gibi tasarlayıp yazınız.



Görsel 2.12: Bilgisayar donanım birimleri uygulaması

```
private void button1 Click(object sender, EventArgs e)
     ł
     string secim;
     secim = listBox1.SelectedItem.ToString();
    if(secim=="Mouse" || secim=="Klavye" || secim == "Kamera" || secim=="Tarayıcı")
        {
           MessageBox.Show("Bu parça, giriş birimidir.");
    if (secim == "Yazıcı" || secim == "Hoparlör" || secim == "Projeksiyon")
        {
           MessageBox.Show("Bu parca, cikis birimidir.");
        }
     }
```

2.2.3. Mantıksal Operatör Önceliği

Karar ifadelerinde ve(&&) operatörü veya(||) operatörüne göre daha öncelikli işleme alınır.

Uygulama-16

Adım 1: Form üzerine 2 adet GroupBox, 2 adet RadioButton, 1 adet ListBox ve 1 adet Button nesnesi ekleyiniz.

Adım 2: ListBox nesnesinin Items özelliğine "MP4, JPG, MOV, PNG" değerlerini ekleyiniz.

Adım 3: ListBox nesnesi içinden seçilen dosya uzantısı ile RadioButton nesnelerinden seçilen dosya türü bilgisinin eşleşmesini kontrol eden programı Görsel 2.13'te görüldüğü gibi tasarlayıp yazınız.











Not: Yukarıdaki uygulamada ve(&&) operatörü öncelikli olarak işleme alınacağı için veya(| |) operatörleri ile oluşturulan şart ifadeleri ayrı bir parantez içine alınmıştır. Aksi durumda program hatalı çalışacaktır.

Sıra Sizde

Yukarıdaki örnek uygulamayı veya(| |) operatörlerini ayrı bir parantez içine almadan çalıştırınız, oluşacak hatalı mesajları arkadaşlarınızla paylaşınız.





2.2.4. NOT(!) Operatörü

Bu operatör; şart ifadesinin alacağı sonuç true ise false, şart ifadesinin alacağı sonuç false ise true şeklinde değiştirir. Şart ifadesinin önüne değil(!) operatörü konularak kullanılır.

if (!radioButton1.Checked == true)
{
 // kodlarınız
}

Vot: Yukarıdaki örnek kodda değil(!) operatörü kullanıldığı için radioButton1 nesnesi işaretli değilken kod çalışacaktır.

if (!(yas>=15 && yas<=65))
{
// kodlarınız
}

Not: Yukarıdaki örnek kodda yas değişkeni şart ifadesinde belirtilen aralığın dışında ise kodlar çalışacaktır (Örneğin; yas=32 olduğunda kodlarınız çalışmayacak, yas=70 olduğunda kodlarınız çalışacaktır.).

Uygulama-17

Adım 1: Form üzerine 2 adet RadioButton nesnesi ve 1 adet Button nesnesi ekleyiniz.

Adım 2: Form nesnesinin arka plan rengini gri yapınız.

Adım 3: Button nesnesine tıklandığında üstteki RadioButton işaretliyse Form nesnesinin arka plan rengini siyah, alttaki RadioButton işaretliyse arka plan rengini beyaz yapan programı Görsel 2.14'te görüldüğü gibi tasarlayıp yazınız.



Görsel 2.14: Arka plan ayarları





private void button1_Click(object sender, EventArgs e)
{
 if (radioButton1.Checked == true)
 {
 this.BackColor = Color.Black;
 radioButton1.ForeColor = Color.White;
 radioButton2.ForeColor = Color.White;
 }
 if (!radioButton1.Checked == true)
 {
 this.BackColor = Color.White;
 radioButton1.ForeColor = Color.Black;
 radioButton2.ForeColor = Color.Black;
 }
}

Not: Yukarıdaki uygulamada if (!radioButton1.Checked == true) kodu yerine if (radioButton1.Checked == false) kodu veya else yapısı da kullanılabilir.

Sıra Sizde >

Yanınızdaki arkadaşınızla birlikte aşağıdaki şart ifadelerini yeni bir proje dosyası açarak deneyiniz ve çıkan sonuçlara göre kodların hangi durumlarda çalışacağını açıklama kısmına yazınız.

Şart İfadesi	Açıklama
if (!(Stok>=10))	Stok değişkeninin değeri 10'dan küçük olduğunda çalışır.
if (!(sayi > 0 && sayi <10))	
if (!(cinsiyet=="erkek"))	
if (!(numara%2==1))	
if (!(renk=="sarı") && numara%10==2)	
if (textBox1.Text != "Murat")	



2.3. DÖNGÜLER



Görsel 2.15: Döngüler

Programda belirli kodların tekrar tekrar çalıştırılmasını sağlayan yapılara **döngü** denir. Duruma göre aynı kod 2 kez çalıştırılabileceği gibi 200 kez hatta 2000 kez de çalıştırılabilir. Böyle durumlarda program yazmak zorlaşacak, zaman alacak ve programın kod yapısı karmaşıklaşacaktır. Örneğin 1000 adet ürün arasından girilen barkod numarasına ait ürünün bilgilerini getirmek için 1000 adet if komutu kullanmak yerine döngü ifadesi içinde sadece bir tane if komutu kullanılabilir.

2.3.1. Sayaçlar

Bir değişkene bağlı değeri farklı aralıklarla artırmak, azaltmak, katlamak veya bölmek gerekebilir. Böyle durumlarda sayaç adı verilen değişkene değer atama yöntemleri kullanılır (Görsel 2.15). Tablo 2.4'te sayaçların kullanımına örnekler verilmiştir.

Şart İfadesi	Açıklama
x=x+1;	x+=1;
x=x-2;	x-=2;
x=x*3;	x*=3
x=x/5;	x/=5;

Tablo 2.4: Sayaçlar



Adım 1: Form üzerine 2 adet Button ve 1 adet TextBox nesnesi ekleyiniz.

Adım 2: Eksi bir yazılı Button nesnesine tıklandığında TextBox nesnesindeki sayıyı 1 azaltan, artı bir yazılı Button nesnesine tıklandığında TextBox nesnesindeki sayıyı 1 artıran programı Görsel 2.16'daki gibi tasarlayıp yazınız.



Görsel 2.16: Sayaçlar





2.3.2. Artırma ve Azaltma Operatörleri

Artırma ve azaltma operatörleri döngü yapılarında çok sık kullanılmaktadır. Döngü değişkeninin değeri birer artıyor veya birer birer azalıyorsa sayaç yerine pratik bir kullanıma sahip olan **artırma ve azaltma** operatörleri tercih edilebilir (Tablo 2.5).

Tablo 2.	5: Artırma	ve Azaltma	Operatörleri
----------	------------	------------	--------------

Operatör	Açıklama
x++;	Değişkenin değerini 1 artırır.
X;	Değişkenin değerini 1 azaltır.



Uygulama-19

Adım 1: Form üzerine 2 adet PictureBox ve 4 adet Button nesnesi ekleyiniz.

Adım 2: PictureBox nesnelerine at resmi yerleştiriniz.

Adım 3: Normal yazılı Button nesnelerine tıklandığında PictureBox nesnesinin Left özelliğini 10 artırarak, Hızlı yazılı Button nesnesine tıklandığında PictureBox nesnesinin Left özelliğini 25 artırarak Picture-Box nesnelerini hareket ettiren programı Görsel 2.17'deki gibi tasarlayıp yazınız (Herhangi bir nesne, Left özelliğinin değeri değiştirilerek Form nesnesi üzerinde sola veya sağa kaydırılabilir.).



Görsel 2.17: Artırma operatörü



2.3.3. For Döngüsü

Genellikle kodların tekrar sayısı belli olduğunda for döngüsü kullanılır. Döngü için tanımlanan şart ifadesi her sağlandığında döngüdeki kodlar tekrar çalışır. For döngüsünün kaç kez çalışacağını belirlemek oldukça basittir.

for (Döngü değişkeni başlangıç değeri; Döngü şart ifadesi; Döngü değişkeni sayacı)
{
//kodlarınız
}

for (int i = 0; i < 10; i++) { //kodlarınız }

Yukarıdaki örnek kodda "i" değişkeninin alacağı her değer için aynı kodlar çalışır. "i" değişkenindeki değer artışı "i++" sayacı (artış operatörü) ile sağlanır. "i" değişkeni sırasıyla 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 değerlerini alır ve böylece döngü içindeki kodlar toplam 10 kez çalışır. "i" değişkeninin değeri 10 olduğunda "i<10" şart ifadesi sağlanamadığı için döngü durur (Döngü değişkeni ikişer ikişer artmış olsaydı döngü içindeki kodlar 5 kez çalışacaktı.).

Uygulama-20

Adım 1: Form üzerine 1 adet ListBox nesnesi ekleyiniz. **Adım 2:** ListBox nesnesinin içine 7 tane "Bilişim Teknolojileri" metni ekleyen programı yazınız.

```
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < 7; i++)
    {
        listBox1.Items.Add("Bilişim Teknolojileri");
    }
}</pre>
```



KARAR VE DÖNGÜ YAPILARI

Not: For döngüsü ile pratik bir şekilde ListBox nesnesine elemanlar eklendi.

private void button1_Click(object sender, EventArgs e)
{

listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri"); listBox1.Items.Add("Bilişim Teknolojileri");

Not: Döngü kullanılmadan uygulama yapıldığında ListBox'a eklenmek istenen elemanların hepsini tek tek yazmak gereklidir.

Uygulama-21

}

Adım 1: Form üzerine 1 adet ListBox nesnesi ekleyiniz. **Adım 2:** ListBox nesnesinin içine 1'den 10'a kadar olan sayıları ekleyen programı yazınız.

```
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 1; i <= 10; i++)
    {
        listBox1.Items.Add(i);
    }
}</pre>
```

Not: Döngü değişkeninin başlangıç değeri "1", bitiş değeri (şart ifadesindeki değeri) "10", artış miktarı da "1"dir. Böylece "i" değişkeni sırasıyla 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 değerlerini alır. Bunlar, ListBox nesnesine eklenecek değerlerdir.

Uygulama-22

Adım 1: Form üzerine 1 adet ComboBox nesnesi ekleyiniz. **Adım 2:** ComboBox nesnesinin içine 6'dan 16'ya kadar olan sayıları ekleyen programı yazınız.

private void button1_Click(object sender, EventArgs e)
{
 for (int i = 6; i <=16; i++)
 {
 comboBox1.Items.Add(i);
 }
}</pre>

•		KARAR VE DÖNGÜ YA	PII
2.	ÖĞRENME BİRİMİ	•	-•

Not: Döngü değişkeninin başlangıç değeri "6", bitiş değeri (şart ifadesindeki değeri) "16", artış miktarı da "1"dir. Böylece "i" değişkeni sırasıyla 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 değerlerini alır.

.ARI

Uygulama-23

Adım 1: Form üzerine 1 adet ComboBox nesnesi ekleyiniz.

Adım 2: ComboBox nesnesinin içine 100 ile 150 arasında 5'in katı olan sayıları ekleyen programı vazınız.

private void button1_Click(object sender, EventArgs e) for (int i = 100; i <=150; i=i+5) { comboBox1.Items.Add(i); }

.Not: Döngü değişkeninin başlangıç değeri "100", bitiş değeri (şart ifadesindeki değeri) "150", artış miktarı da "5"tir. Böylece "i" değişkeni sırasıyla 100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150 değerlerini alır.

Sıra Sizde >

ListBox nesnesinin icine sırasıyla 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 sayılarını ekleyen programı yazınız.

Sıra Sizde >

ListBox nesnesinin içine sırasıyla 110, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10 sayılarını ekleyen programı yazınız.



Adım 1: Form üzerine 2 adet Label, 2 adet TextBox ve 1 adet Button nesnesi ekleyiniz. Adım 2: Button nesnesine tıklandığında TextBox nesneleri içine girilen başlangıç ve bitiş değerleri arasındaki sayıları toplayarak sonucu mesaj veren programı Görsel 2.18'de görüldüğü gibi tasarlayıp yazınız.

KARAR VE	DÖNGÜ YAPILARI
OĞRENME BİRİMİ	
	🛃 Hesaplamalar — 🗆 X
	Başlangıç Değeri: 1 Bitiş Değeri: 100
	Hesapla
private void button1 Click	Görsel 2.18: Hesaplamalar
{ int sayi1, sayi2, to	oplam;
sayi1 = Convert.T sayi2 = Convert.T	ToInt32(textBox1.Text); ToInt32(textBox2.Text);
toplam = 0; for (int i = savi1:	$i \leq = savi2 \cdot i + +)$
{	$r = -\frac{3}{2} + \frac{1}{2} + \frac{1}{2}$
}	$w(\mathcal{C}_{\mathcal{C}}) = \frac{1}{2} + \frac{1}{2}$
MessageBox.Snov	w(sayıların toplamı= + toplam. tostring());

Not: Yukarıdaki uygulamada döngünün başlangıç ve bitiş değerleri için değişkenler kullanılmıştır.

Uygulama-26

Adım 1: Form üzerine 1 adet ListBox ve 1 adet Button nesnesi ekleyiniz.

Adım 2: 10 sayısının 0'dan 4'e kadar olan kuvvetlerini ListBox nesnesi içine ekleyen programı Görsel 2.19'da görüldüğü gibi tasarlayıp yazınız.

1 10 100		
1000 10000		
İsləmi Yan	1	

Görsel 2.19: Kuvvet alma

```
private void button1_Click(object sender, EventArgs e)
{
    double kuvvet;
    for (int i = 0; i < 5; i++)
    {
        kuvvet = Math.Pow(10, i);
        listBox1.Items.Add(kuvvet);
    }
}</pre>
```



Adım 1: Form üzerine 3 adet ListBox, 3 adet Label ve 1 adet Button nesnesi ekleyiniz.

Adım 2: İlk ListBox nesnesinin Items özelliğine "Ram, Hoparlör, Anakart, Ekran Kartı, Monitör, CPU, Mikrofon, Harddisk" değerlerini ekleyiniz.

Adım 3: Button nesnesine tıklandığında listBox1 nesnesi içinde karışık bir şekilde listelenmiş bilgisayar parçalarını dış donanım ve iç donanım birimleri olarak ayırıp listBox2 ve listBox3 nesnelerine aktaran programı Görsel 2.20'de görüldüğü gibi tasarlayıp yazınız.

🖳 Donanım Bilgisi	33 <u>—</u> 83		×	
Tüm Parçalar Ram Hoparlör Anakart Ekran Kartı Monitör CPU Mikrofon Hardisk	Dış Donanım Parçalan Hoparlör Monitör Mikrofon	İç Donar Ram Anakar Ekran CPU Hardisł	nım Parça t Kartı c	lan
Aktar				

Görsel 2.20: Donanım bilgisi



Not: listBox1.Items.Count kodu listBox nesnesinin içindeki item sayısını verir. Yukarıdaki uygulamada listBox1 nesnesinde 8 tane parça ismi ekli olduğu için listBox1.Items.Count kodu ile 8 sayısı elde edilir.



Sıra Sizde

Rastgele 20 adet sayıyı listBox1 nesnesi içine ekleyiniz. Butona tıkladığınızda listBox1 nesnesindeki tek sayıları listBox2 nesnesine, çift sayıları ise listBox3 nesnesine aktaran programı yazınız.

2.3.4. While Döngüsü

Döngü için tanımlanan şart ifadesi sağlanıyorsa döngü çalışmaya başlar. Şart ifadesi sağlandığı sürece while döngüsü çalışmaya devam eder.

while (Şart ifadesi) { //kodlar

Uygulama-27

Adım 1: Form üzerine ListBox nesnesi ekleyiniz. **Adım 2:** ListBox nesnesi içine 1'den 10'a kadar olan sayıları ekleyen programı yazınız.

```
int say = 1;
while (say<=10)
{
    listBox1.Items.Add(say);
    say++;
}</pre>
```

Not: While ve do-while döngülerinde döngü değişkeni, for döngüsünden farklı olarak döngünün dışında tanımlanır ve değişkenin değeri döngünün içinde değiştirilir.

Sıra Sizde

ListBox nesnesi içine 20 tane farklı sayı ekleyiniz. Button nesnesine tıklandığında ListBox nesnesindeki ilk sayıdan başlayarak sayıları toplayan programı while döngüsü kullanarak yazınız.

Uygulama-28

0'dan başlayarak sırasıyla sayıları toplayan ve sayıların toplamı 1000'den fazla olduğunda döngünün kaç kez çalıştığını bulan programı yazınız.

```
int dongu_say = 1;
int toplam = 0;
while (toplam<=1000)
{
    toplam = toplam + dongu_say;
    dongu_say += 1;
}
MessageBox.Show("Döngü toplam " + dongu_say.ToString() + " kez çalıştı");
```

2.3.5. Do-while Döngüsü

Döngü için tanımlanan şart ifadesi sağlanmasa da do-while döngüsü en az bir kez çalışır çünkü while döngüsünde şart ifadesi döngünün başlangıcındayken do-while döngüsünde şart ifadesi döngünün sonundadır. Do-while döngüsü de diğer döngüler gibi şart sağlandığı sürece çalışmaya devam eder.


•		KARAR VE DÖNGÜ YAPILARI	
	ÖĞRENME BİRİMİ		
	do{//kodlar		

}while (Şart ifadesi)

Uygulama-28

Adım 1: Form üzerine ListBox nesnesi ekleyiniz.

Adım 2: ListBox nesnesi içine 1'den 10'a kadar olan sayıları ekleyen programı yazınız.

Do-while döngüsü şart sağlanmasa bile içindeki kodları bir kez çalıştırır.

do-while	while
int say = 1;	int say = 1;
do	while (say > 5)
{	{
listBox1.Items.Add(say);	listBox1.Items.Add(say);
say++;	say++;
} while (say > 5);	}

Not: While döngüsünde şart ifadesi sağlanmadığı için listBox1 nesnesine hiçbir sayı eklenmeyecektir. Do-while döngüsünde şart sağlanmasa bile listBox1 nesnesine bir tane eleman eklenecektir.

Uygulama-29

Adım 1: Form üzerine 1 adet RichTextBox, 1 adet Button nesnesi ekleyiniz.

Adım 2: Button nesnesine tıklandığında 0'dan 30'a kadar olan sayıları yan yana RichTextBox nesnesine aktaran programı do-while döngüsü kullanarak Görsel 2.21'deki gibi tasarlayıp yazınız.



Görsel 2.21: Metinsel toplama



2.3.6. Döngüyü Kesme (Durdurma)

Döngüyü kesme komutu "break;" döngü tamamlanmadan döngüden çıkmak için kullanılır. "break;" komutundan sonra döngüye ait hiçbir kod çalışmaz ve program, döngü ifadesinden sonraki kod satırı ile çalışmaya devam eder.

For Döngüsü		
for (int a = 0; a < 10; a++)		
{		
MessageBox.Show("Döngü Çalışıyor.");		
if (a == 5)		
{		
MessageBox.Show("Döngü Durdu.");		
break;		
}		
}		
While Döngüsü	Do-While Döngüsü	
int $a = 0;$	int $a = 0;$	
while (a<10)	do	
{	{	
MessageBox.Show("Döngü Çalışıyor.");	MessageBox.Show("Döngü Çalışıyor.");	
if (a == 5)	if (a == 5)	
{	{	
MessageBox.Show("Döngü Durdu.");	MessageBox.Show("Döngü Durdu.");	
break;	break;	
}	}	
a++;	a++;	
}	}while (a<10);	

Not: Döngü türleri için ayrı ayrı verilen örnek kodlara göre "a" döngü değişkeni en son 5 değerini alır, "break;" kesme komutu kullanıldığı için döngü tamamlanmadan durur ve döngü 10 kez çalışması gerekirken 6 kez çalışır.



Uygulama-31



10. SINIF

Adım 1: Form nesnesi üzerine 2 adet Label, 2 adet ListBox, 1 adet TextBox ve 1 adet Button nesnesi ekleyiniz.

Adım 2: ListBox nesnesinin Items özelliğine 10 adetten fazla isim giriniz.

NESNE TABANLI PROGRAMALAMA

•		KARAR VE DÖNGÜ YAPILARI
	ÖĞRENME BİRİMİ	

Adım 3: TextBox nesnesi içine girilen müşteri adını ListBox nesnesi içinde arayarak eşleşen ad varsa "Aradığınız müşteri bulundu!" şeklinde mesaj veren programı Görsel 2.22'de görüldüğü gibi tasarlayıp yazınız.



Görsel 2.22: Müşteri arama

string ad = textBox1.Text;
for (int i = 0; i < listBox1.Items.Count; i++)
{
 if (listBox1.Items[i].ToString() == ad)
 {
 MessageBox.Show(listBox1.Items[i].ToString() + " adlı müşteri bulundu!");
 break;
 }
}</pre>

Not: Uygulamada "....adlı müşteri bulundu!" mesajı verildikten sonra "break;" komutu çalıştığı için döngü kesintiye uğrar ve sonlanır. Döngü içinde "break;" komutu kullanılmasaydı müşterinin adı bulunduktan sonra döngüde yapılan tüm işlemler gereksiz olacaktı. Bu gereksiz işlemler, programın çalışma performansını etkilemektedir. 3000 tane müşteri adının kayıtlı olduğu bir listede aranan müşteri listenin 15. sırasında bulunmuş ise geri kalan 2985 tane kaydı kontrol etmek hem gereksizdir hem de programın performans kaybına sebep olur.

Sıra Sizde >

Yukarıdaki uygulamada for döngüsü ile düzenlenmiş döngü kodlarını **while** ve **do-while** döngülerini kullanarak ayrı ayrı yazınız.





Sıra Sizde

1 ile 100 arasında 3'ün katı olan sayıları toplarken sayıların toplamı 200'ü geçince sadece bir kez "Limit aşıldı." mesajı veren programı yazınız.

2.3.7. Döngüyü Devam Ettirme

Döngü içinde "continue;" komutu çalıştıktan sonra diğer kodlar döngünün o andaki adımı için çalışmaz ve döngü bir sonraki adıma geçer. Böylece döngü isteğe bağlı bir şekilde belli adımları atlayarak çalışır.

for (int a = 1; a < 15; a++)
 {
 if (a==5 || a==10)
 {
 continue;
 }
 //kodlar
 }
</pre>

int a = 1;int a = 1; while (a<15) do{ //kodlar { //kodlar if (a==5 || a==10) if (a==5 || a==10) { continue; { } continue; } a++; }while (a<15); a++;

Yukarıdaki döngü ifadelerinde "a" değişkeninin değeri 5'e veya 10'a eşit olduğunda döngü bir sonraki değerini alır.

Sıra Sizde

Aşağıdaki kodlara göre listBox1 nesnesi içine hangi sayılar eklenir?

for (int i = 1; i <= 10; i++) for (int i = 1; i <= 10; i=i+2) { { listBox1.Items.Add(i); **if** (i < 7) **if** (i >= 7) { { continue; continue; } } listBox1.Items.Add(i); } } Cevap: Cevap:





2.4. HATA AYIKLAMA

Hata Listesi			
Tüm Çözüm	-	🕄 2 Hata	🔒 1 Uyarı

Görsel 2.23: Hata ayıklama

Kod editörü derleyicisi, programın kodları yazım kurallarına uygun olmadığında Hata Listesi panelinde hatalı kodları göstermektedir (Görsel 2.23). Bazı hatalar program çalıştıktan sonra oluşmaktadır ve programı durdurmaktadır. Örneğin, programda TextBox nesnesine girilen sayı ile işlem yapılıyorsa fakat TextBox nesnesine sayı yerine harf girilmişse program hata verip duracaktır. Bu hata, kullanıcının yanlış veri girişinden kaynaklandığı için derleyici bu tip hataları yakalayıp Error List panelinde gösteremez. Bu tip hataların programı durdurmaması için önlem olarak **try-catch-finally** hata ayıklama blokları kullanılmalıdır.

{		Özel Durum İşlenmedi	-= X
<pre>int sayi1,karesi; sayi1 = Convert.ToInt32(textBox1.Text);</pre>	0	System.FormatException: 'Giriş dizesi doğru biçimde değildi.'	
<pre>karesi = sayi1 * sayi1; MessageBox.Show(karesi.ToString());</pre>		Bu özel durum özgün olarak bu çağrı yığınında oluşturuldu: [Dış Kod]	

Görsel 2.24: Hatalı kod-1

Yukarıdaki örnek kodlar çalıştırıldığında textBox1 nesnesine sayı yerine harf girilmişse program Görsel 2.24'teki gibi hata verecek ve durdurulacaktır. Bu tip hatalar özel durumlarda oluşan hatalardır ve program çalıştırılmadan önce derleyici tarafından tespit edilemez.

2.4.1. Try-Catch-Finally Bloku

Program çalıştırıldığında hata meydana gelme olasılığı olan kodlar **try** bloku içinde yazılır. Try bloku içine yazılan kodlarda hata meydana gelirse program **try** blokundan çıkarak **catch** bloku içindeki kodları çalıştırır. Böylece program hata vermez ve çalışmaya devam eder.

Try bloku içine yazılan kodlarda bir hata meydana gelmezse **catch** blokundaki kodlar çalışmaz fakat **try** blokunda hata meydana gelse de gelmese de **finally** blokundaki kodlar çalışır. **Finally** blokunu kullanmak zorunlu değildir, tercihe bağlıdır.

Uygulama-32

Adım 1: Form üzerine Button nesnesi ekleyiniz.Adım 2: Button nesnesinin Click olay metodunu oluşturunuz.Adım 3: Aşağıdaki kodları Click olay metodunun içine yazınız.

```
int sayi1,sayinin_karesi;
```

```
try
{
  sayi1 = Convert.ToInt16(textBox1.Text);
  sayinin_karesi = sayi1 * sayi1;
  MessageBox.Show(sayinin_karesi.ToString());
  }
  catch
  {
    MessageBox.Show("Hatalı giriş yaptınız!");
  }
  finally
  {
    // Hata olsa da olmasa da çalışacak kodlar
  }
```



<u> </u>	/ükle	

Görsel 2.25: Yüzde uygulaması

Özel Durum İşlenmedi 🛛 🗝 🗙
System.ArgumentOutOfRangeException: "128' değeri "Value' için geçerli değil. 'Value' değeri 'minimum' ile 'maximum' arasında olmalıdır. Parametre adı: Value'
Bu özel durum özgün olarak bu çağrı yığınında oluşturuldu: [Dış Kod] WindowsFormsApp10.Form1.button1 Click(object System.EventAras) / Form1.

Not: ProgressBar nesnesi 0 ile 100 arasındaki değerleri alır. Program, hata ayıklama kodları olmadan yazılırsa Görsel 2.26'daki gibi hata meydana gelecek ve program duracaktır.





ÖLÇME VE DEĞERLENDİRME - 2 1. Eğer "a" değişkeni "b" ve "c" değişkeninden büyükse ifadesini kod olarak yazınız. 2. Eğer "a" değişkeni "b" veya "c" değişkeninden kücükse ifadesini kod olarak yazınız. 3. Aşağıdaki cümlelerde parantez içine yargılar doğru ise (D), yanlış ise (Y) yazınız. () $<_{i}>_{i}=_{i}<=_{i}>=_{i}!=$ karakterleri mantıksal operatörlerdir. () Üçten fazla if ifadesi iç içe kullanılmaz. () Şart ifadesi sağlanmadığında çalışacak kodlar "else" blokunun içine yazılabilir. () Karar ifadelerinde ve(&&) operatörü veya(||) operatöründen öncelikli işleme alınır. () Döngü şart ifadesi sağlanmasa da for döngüsü en az bir kez çalışır. 4. Aşağıdaki kodlar çalıştırıldığında comboBox1 nesnesine eklenecek değerler nelerdir? for (int i = 1; $i \le 20$; i = i + 1) { if (i % 3 == 0) { comboBox1.Items.Add(i); if (i == 17) { comboBox1.Items.Add(i); } } 5. Aşağıdaki kodlar çalıştırıldığında comboBox1 nesnesine eklenecek değerler nelerdir? for (int i = 1; $i \le 15$; i = i + 1) { if (i % 3 == 0) { continue; } comboBox1.Items.Add(i); }







NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Nesne tabanlı programlama mantığını kavrayacak,
- Nesne tabanlı programlamanın temel prensiplerini açıklayacak,
- Sınıf ve nesneler oluşturacak,
- Sınıflarda alan, özellik ve metot ögelerini kullanacak,
- Erişim belirleyicilerini sıralayacak,
- Metotlar yazacak ve kullanacak,
- Değer ve referans kavramlarını açıklayacak,
- Sınıflarda kalıtım özelliklerini kullanacak,
- Soyut sınıf, arayüz ve çok biçimlilik kavramlarını açıklayacak,
- Statik, isimsiz, mühürlü ve parçalı sınıfları kullanacak,
- Numaralandırma mantığını öğreneceksiniz.

ANAHTAR KELİMELER

Sınıf, nesne, alan, özellik, metot, kapsülleme, metot, kalıtım, soyut sınıf, arayüz, çok biçimlilik, statik sınıf, isimsiz sınıf, mühürlü sınıf, parçalı sınıf, enums

SINIFLAR	(CLASS)
----------	---------



HAZIRLIK ÇALIŞMALARI

- 1. Yapısal (prosedürel) programlama mantığı ile ilgili neler biliyorsunuz?
- 2. Nesne tabanlı programlamanın en çok tercih edilen yazılım geliştirme yaklaşımı olmasının nedeni

neler olabilir?

3.1. NESNE TABANLI PROGRAMLAMAYA GIRIS

1960'lı yılların sonunda ortaya çıkan Nesne Tabanlı Programlama (NTP) kavramı, günümüzde yazılım geliştirmek, tasarlamak, analiz ve test etmek için kullanılır. NTP, öğrenilmesi gereken en önemli kavramlardan biridir. NTP bazı kaynaklarda "Nesne Yaklaşımlı Programlama", "Nesneye Yönelik Programlama", "Nesne Yönelimli Yaklaşım" olarak da kullanılmaktadır [(OOP) Object Oriented Programming-(OOA) Object Oriented Approach].

NTP, yazılımların boyutları ve karmaşıklığı arttığı icin bazı yazılım gereksinimlerini karşılamak amacıyla doğmustur. NTP, yazılan kodların bakımını ve aynı yazılım üzerinde birden fazla yazılımcının calısmasını kolaylaştırmaktadır. Günümüzde geniş çaplı yazılım projelerinde yaygın olarak NTP kullanılmaktadır. NTP, genel olarak kullanılan fonksiyonel veya yapısal programlama tekniğine yeni bir bakış açısı getirmiştir. Günümüzde iyi bir yazılımcı olmak isteyen herkes NTP mantığını ve prensiplerini detaylı bir şekilde öğrenmelidir. Ayrıca bilinmesi gerekir ki şu an dünyada kullanılan -neredeyse- tüm programlama dilleri NTP'yi desteklemektedir.

3.1.1. NTP Öncesi

NTP öncesinde fonksiyonel programlama yaklaşımı mevcuttu. Fonksiyonel programlama, 1950'li yıllarda ilk üst düzey dillerin ortaya çıkışından bu yana kullanılagelmektedir. Hem yapısal programlama hem de NTP genel amaçlı tüm programlama dillerinde uygulanabilmektedir.

Yapısal programlama, büyük programları kücük parcalara bölerek cözümleme yöntemidir. Modüller ve alt programlar sıralı bir şekilde çalışmaktadır. Bu çalışma tarzı, ek bir programcılık yükü getirmekte ve kodların işleyişinin takibini zorlaştırmaktadır.

3.1.2. NTP Temel Prensipleri

NTP genel olarak dört prensip üzerine kurulmuştur. Herhangi bir programlama dilinin nesne tabanlı olduğundan söz edebilmek için asgari düzeyde şu prensipleri desteklemesi gerekmektedir:

Soyutlama (Abstraction): Karmaşıklığın azaltılması anlamına gelmektedir. Örneğin otomobiller-

de gaz pedalına basıldığında otomobil hızlanır ancak arka planda olan bitenler çoğu kişi için önemsizdir.

Sarmalama veya Kapsülleme (Encapsulation): Sadece istenilen bilgilerin dıs dünyaya acılma-

sı, hassas veya özel bilgilerin gizlenmesi anlamına gelmektedir. "Banka hesabına para yatır." komutu verildikten sonra T.C. kimlik numarası ve şifre bilgilerinin gizlenmesi buna örnek verilebilir.

Kalıtım (Inheritance): Var olan özelliklerin aktarılması anlamına gelmektedir. Örneğin, bütün

kedi türlerinin dört ayaklı olması ortak bir özelliktir. Bir Van kedisi, kedilerin tüm özelliklerini tasırken ayrıca kendine has özellikleri de barındırır.

Cok bicimlilik (Polymorphism): Farklı tiplere ait olan ortak özellikleri tanımlama işlemidir. Örne-

ğin, farklı hayvan türleri farklı sesler çıkarır zira "ses çıkarma" eylemi ortak bir özelliktir.

3.2. SINIFLAR VE NESNELER

Çevremiz ve dünyamız incelendiğinde her şeyin (cisimler, canlılar vb.) belirli özelliklerinin ve işlevlerinin olduğu görülür. Her öğrencinin bir numarası, adı soyadı, aldığı dersler gibi özellikleri ve okula gitme, sınava girme gibi **işlevleri** vardır. Benzer şekilde yine bir cep telefonunun rengi, boyutları, markası, fiyatı gibi özellikleri ve cağrı baslatma, mesaj gönderme, uygulama acma gibi islevleri bulunmaktadır.

NTP, dünyada var olan her sevin yazılım icinde modellenmesini amaclamaktadır. Sınıf (class), NTP'nin en önemli kavramıdır. Sınıf, nesnelerin özelliklerini ve işlevlerini (davranışlarını) tanımlamak için kullanılan bir taslaktır. Bu taslak aracılığıyla nesneler (objects) oluşturulmaktadır.



10, SINIF



Görsel 3.1: Sınıf ve nesne ilişkisi

Görsel 3.1'de bir ev planı görülmektedir. Programlamada bu ev planına sınıf, ev planından yola çıkılarak yapılan gerçek eve ise nesne adı verilebilir.

Sıra Sizde >

Çevrenizde gördüğünüz nesnelerden en az beş tanesinin özelliklerini ve işlevlerini yazınız. Yazdıklarınızı sınıf arkadaşlarınızla paylaşınız.

3.2.1. Sınıf Tanımlama

C#'ta sınıf tanımında class anahtar kelimesi kullanılmaktadır. Aşağıda en basit sınıf tanımı verilmiştir.

class SinifAdi
{
}

Aşağıdaki örnekte bir dikdörtgen sınıfı tanımı yapılmaktadır.

```
class Dikdortgen
{
  private int a, b;
   public Dikdortgen(int a, int b)
  {
     this.a = a;
     this.b = b;
  }
public int AlanHesapla()
  {
     return a * b;
   }
   public int CevreHesapla()
   ł
      return 2 * (a + b);
   }
```

Not: public ve private kavramları 3.4. Erişim Belirleyicileri konusunda açıklanacaktır.

SINIFLAR(CLASS))
-----------------	---

Bir dikdörtgenin iki kenar uzunluğu bilgisi bulunmaktadır. Ayrıca çevre ve alan bilgilerinin hesabı da söz konusudur. Yukarıdaki sınıf tanımında a ve b değişkenleri dikdörtgenin kenar uzunluklarını saklamak için, **AlanHesapla()** ve **CevreHesapla()** işlevleri de dikdörtgenin alan ve çevre hesabının yapılması için tanımlanmıştır.

Not: C#'ta işlevleri gerçekleştiren kod bloklarına "metot" denilmektedir. Dolayısıyla bundan sonra işlev kelimesi yerine metot kelimesi kullanılacaktır.

Yukarıdaki kod blokunda aşağıdaki gibi bir sınıf tanımlaması yapılmıştır.

class Dikdortgen

ÖĞRENME BİRİMİ

Sınıfın adı "Dikdortgen"dir. İsimlendirmelerde genel prensip olarak "ı, İ, ü, Ü, ö, Ö, ğ, Ğ, ş, Ş" gibi alfabemize özel harflerin kullanılmaması uygundur.

Daha genel yazılacak olursa bir sınıf aşağıdaki gibi tanımlanmaktadır.

«Erişim belirleyici» class «Sınıf adı»
{
}

Sıra Sizde >

Dikdortgen sınıfına benzer şekilde Daire sınıfını oluşturunuz.

3.2.2. Nesne Oluşturma

Programlarda sınıfların kullanılabilmesi için bu sınıftan oluşturulan nesnelere (object) gereksinim duyulur. Bu türetme işlemine örnek oluşturma (instance) adı da verilir.

C#'ta önceden tanımlanan bir sınıftan **nesne** türetmek için **new** anahtar kelimesi kullanılmaktadır. Daha önceden oluşturulan Dikdortgen sınıfından bir nesne türetmek ve bu nesnenin ögelerini (özellikler ve metotlar) kullanmak için aşağıdaki gibi bir konsol uygulaması yazılabilir.

```
private static void Main(string[] args)
```

{

Dikdortgen d = new Dikdortgen(3, 4);

Console.WriteLine("Dikdörtgenin alanı: {0}", d.AlanHesapla());

```
Console.WriteLine("Dikdörtgenin çevresi: {0}", d.CevreHesapla());
```

}

Oluşturulan nesnenin adı "**d**"dir. Kenar uzunlukları 3 ve 4 olarak belirlenmiştir. Nesnenin ögelerine erişmek için . (nokta) karakterinin kullanıldığı görülmektedir.

Genel olarak yazılacaksa bir nesne aşağıdaki gibi tanımlanmaktadır.

«Sinif adı» «Nesne adı» = new «Sinif adı»(««Parametre listesi»»);

Sıra Sizde

Dikdortgen nesnesine benzer şekilde yarıçapları farklı iki adet Daire nesnesi oluşturup metotlarını kullanınız.

3.3. KAPSÜLLEME, ALANLAR VE ÖZELLİKLER (ENCAPSULATION, FIELDS, PROPERTIES)

Erişim belirleyicileri ile NTP'nin temel prensiplerinden olan "**kapsülleme**", alanlar ve özellikler aracılığıyla programlarda uygulanabilmektedir. Kapsülleme, hassas veya özel bilgilerin gizlenmesi anlamına gelmektedir.



SINIFLAR(CLASS)

ÖĞRENME BİRİMİ

3.4. ERİŞİM BELİRLEYİCİLER (ACCESS MODIFIERS)

.NET platformunda oluşturulan uygulamalarda güvenliği artırmak amacıyla sınıflara ve/veya sınıf içinde kullanılan ögelere erişimin kısıtlanması gerekmektedir. Dolayısıyla koda dışarıdan erişimin sınırlarını belirlemek amacıyla erişim belirleyicileri kullanılır.

C# programlama dilinde kullanılan erişim belirleyicileri şunlardır:

• **public** (Genel): public olarak tanımlanan ögeler üzerinde herhangi bir kısıtlama yoktur. Her yerden erişilebilirdir.

• **private** (Gizli): En katı erişim belirleyicidir. Ögeler sadece tanımlandığı sınıf içinde erişilebilirdir. Bir başka deyişle ögeler sadece tanımlandığı küme parantezleri arasında kullanılabilir.

• **protected** (Korunumlu): Ögeler, bulunduğu sınıf içinde ya da bu sınıftan türeyen diğer sınıflarda erişilebilirdir.

• internal (Dâhilî): internal olarak tanımlanan ögelere sadece aynı program içinden erişilebilir.

• **protected internal** (Dâhilî+Korumalı): Ögeler hem protected hem de internal erişim belirleyicisine sahip olarak değerlendirilir. Türetilen sınıfın farklı program içinde olması sorun teşkil etmemektedir.



Gizliden genele doğru erişim belirleyicileri Görsel 3.2'deki hiyerarşiye sahiptir.

Görsel 3.2: Erişim belirleyicileri hiyerarşisi

Not: Bir ögeye herhangi bir erişim belirleyicisi tanımlaması yapılmazsa varsayılan olarak private olduğu kabul edilir.

3.5. ALANLAR (FIELDS)

Bir alan, sınıf içinde tanımlanmış herhangi türden (int, string vb.) bir değişkendir. Aşağıda Ucgen sınıfı ve bu sınıfa ait üç adet alan tanımlanmıştır.

```
public class Ucgen
{
    public int a;
    public int b;
    public int c;
}
internal class Program
{
    private static void Main(string[] args)
    {
      Ucgen ucgen = new Ucgen();
      ucgen.a = 3;
      ucgen.b = 4;
      ucgen.c = 5;
      Console.WriteLine("Üçgenin a kenar uzunluğu: {0}", ucgen.a);
      Console.WriteLine("Üçgenin b kenar uzunluğu: {0}", ucgen.c);
    }
}
```

SINIFL	AR(CLASS)
--------	-----------



Bu kod blokunda Ucgen sınıfından ucgen adında bir nesne türetilmiştir (C#'ın büyük / küçük harf duyarlı bir dil olduğu unutulmamalıdır. Dolayısıyla "U" ile "u" farklı karakterlerdir.).

Nesnenin alanlarına erişim için Görsel 3.3'te görüldüğü üzere . (nokta) karakteri kullanılmaktadır. Kod editöründe nesnenin adı yazıldıktan sonra . (nokta) karakterine basıldığında sınıfa ait kullanılabilir ögelerin listesi gelmektedir.

ucgen.	
	a (field) int Ucgen.a
	🤗 b
	🤗 c
	Equals
	GetHashCode
	𝒮 GetType
	⊞ 🖌 🖉

Görsel 3.3: Sınıf ögelerine erişim

Sınıfa ait alanlar tanımlanırken başına "public" erişim belirleyicisi yazılmıştır. Bu erişim belirleyicisi, alan bilgisine sınıf dışından erişim için gereklidir. Alanlar yalnızca özel ve gizli kalması gereken değişkenler için kullanılmalıdır. Sınıf içinde tanımlanmış bir değişkenin başına yazılan "public" erişim belirleyicisi ile alanı dış dünyaya açmak uygun değildir. Bu şekilde yapıldığında değişkene değer atama ya da değişkenin değerinin okunması işlemlerinde kontrol mekanizması işletilememektedir. Ucgen sınıfına ait bir değişkene aşağıdaki değer ataması kolaylıkla yapılabilir.

Ucgen ucgen = new Ucgen(); ucgen.a = -3; ucgen.b = 0; ucgen.c = -12345;

Yukarıda belirtilen kenar uzunluklarına sahip bir üçgeni çizmek mümkün değildir. Bu da programın doğru çalışmayacağı anlamına gelir. Bu durumun önüne geçmek için **özellikler** kullanılır.

Sıra Sizde >

- **1.** Öğrenci sınıfını ve alanlarını oluşturunuz.
- 2. Bilgisayar sınıfını ve alanlarını oluşturunuz.



ÖĞRENME BİRİMİ

3.6. ÖZELLİKLER (PROPERTIES)

Bir değişkeni dış dünyaya açmak (diğer sınıflardan, programlardan vb. erişmek) için bu değişkene ait bir özellik eklenir. Bu sayece NTP'nin temel prensiplerinden "kapsülleme" prensibi sınıfa uygulanmış olur.

Aşağıda Ucgen sınıfının NTP prensiplerine daha uygun hazırlanmış örneği verilmiştir.

```
public class Ucgen
   int a;
   int b;
int c;
   public int A
      get { return a; }
      šet
      {
          if (value <= 0)
   Console.WriteLine("Hatalı bilgi");</pre>
          else
             a = value;
      }
   }
   public int B
      get { return b; }
      set {
          if (value \leq = 0)
             Console.WriteLine("Hatalı bilgi");
          else
             b = value;
      }
   }
   public int C
      get { return c; }
      šet
      {
          if (value \leq = 0)
             Console.WriteLine("Hatalı bilgi");
      else
             c = value;
      2
   }
```

Not: Değişkenlerin başında erişim belirleyicisinin olmadığına dikkat edilmelidir.

Görsel 3.4'teki gibi sınıf adı yazılıp . (nokta) karakterine basıldığında alanlar değil, özellikler görüntülenmektedir.



Görsel 3.4: Sınıf özelliklerine erişim

ÖĞRENME BİRİMİ



Bir özelliğin **get** (almak, elde etmek) ve **set** (düzenlemek, ayarlamak) şeklinde iki ayrı alt metodu bulunur.

• **get Metodu:** Bir değer döndürmek için kullanılır. Özelliklerin get metodunda **return** anahtar kelimesi kullanılarak "return...;" ile bir değerin döndürüleceği belirtilir.

 set Metodu: Değişkene değer atama işlemleri için kullanılır. Burada görülen value anahtar kelimesi dışarıdan bu özelliğe gönderilen değeri temsil etmektedir.

```
if (value <= 0)
Console.WriteLine("Hatalı bilgi");</pre>
```

Yukarıdaki kod satırları ile dışarıdan alınan bilginin kontrolü gerçekleştirilmektedir, sıfır veya negatif bir değer gönderildiğinde kullanıcıya hata mesajı gösterilmektedir.

Özellikler kullanılarak Ucgen sınıfının alanlarına değer atamak için aşağıdaki kod bloku yazılır.

```
Ucgen ucgen = new Ucgen();

ucgen.A = 3;

ucgen.B = 4;

ucgen.C = 5;

Console.WriteLine("Üçgenin a kenar uzunluğu: {0}", ucgen.A);

Console.WriteLine("Üçgenin b kenar uzunluğu: {0}", ucgen.B);

Console.WriteLine("Üçgenin c kenar uzunluğu: {0}", ucgen.C);
```

A kenarına negatif bir değer atanmak istenildiğinde kullanıcıya bir uyarı mesajı gösterilecektir.

```
private static void Main(string[] args)
{
    Ucgen ucgen = new Ucgen();
    ucgen.A = -3;
}
// Ekran çıktısı:
Hatalı bilgi
```

Programlarda özellikler sıklıkla kullanıldığı için kısa yoldan dâhilî bir değişken ve bu değişkene ait özellik tanımlanabilir.

```
public class Ucgen
{
    public int A { get; set; }
    public int B { get; set; }
    public int C { get; set; }
}
```

Nesne oluştururken ilk değer ataması da kısa yoldan yapılabilir.

```
static void Main(string[] args)
{
    Ucgen u = new Ucgen
    {
        A = 3,
        B = 4,
        C = 5
};
```

Yapıcı metotları aşırı yüklemeye (Aşırı yükleme konusu 3.7.5'te incelenecektir.) gerek kalmadan doğrudan özelliklere değer ataması gerçekleştirilebilmektedir. Özelliklerin programcılara sunduğu bazı avantajlar vardır. Her özellik;

- Sadece get metoduna,
- Sadece set metoduna,
- Hem get hem set metoduna sahip olabilir.





Sıra Sizde

- 1. Öğrenci sınıfını ve özelliklerini oluşturunuz.
- 2. Bilgisayar sınıfını ve özelliklerini oluşturunuz.

3.6.1. Sadece Okunabilir Özellikler

Bir özellikte sadece **get** metodu kullanılırsa dışarıdan bu özelliğe değer ataması gerçekleştirilemez. Bu özellik = (eşittir) karakterinin solunda kullanılamaz. Bu özellik "sadece okunabilir" (readonly) bir özellik olur. Aşağıda sadece okunabilir bir özellik tanımı yapılmıştır.

```
public class Ucgen
  // ...
  public int Cevre
     get
        return a + b + c;
     }
  }
internal class Program
ł
  private static void Main(string[] args)
     Ucgen ucgen = new Ucgen();
     ucgen.A = 3;
     ucgen.B = 4;
     ucgen.C = 5;
     Console.WriteLine("Üçgenin çevresi: {0}", ucgen.Cevre);
  }
 Ekran çıktısı:
Üçgenin çevresi: 12
```

Buna karşın Cevre özelliğine değer atanmak istenirse bir hata oluşacaktır.

Görsel 3.5: Salt okunur özelliğe değer atamaya çalışma

Görsel 3.5'te görüldüğü gibi ucgen nesnesinin Cevre özelliğine değer atama işlemi gerçekleştirilememektedir.

3.6.2. Sadece Yazılabilir Özellikler

Bir özellik sadece **set** metoduna sahipse bu özelliğe "sadece yazılabilir" özellik denir. Bu özellik = (eşittir) karakterinin sağında kullanılamaz. Dolayısıyla bu tür özellikler değer döndürmez, sadece dışarıdan değer alabilir.





Daha önceden yazılan Ucgen sınıfının bir özelliği sadece yazılabilir özellik yapılmak istenirse bu özellik aşağıdaki gibi tanımlanmalıdır.

pub	lic class Ucgen
t ii ir p {	nt a; nt b; nt c; ublic int A
}	<pre>set { if (value <= 0) Console.WriteLine("Hatalı bilgi"); else a = value; } /</pre>
} inte	rnal class Program
{	
l p	rivate static void Main(string[] args)
	Ucgen ucgen = new Ucgen(); ucgen.A = 3; ucgen.B = 4; ucgen.C = 5;
۲ ۲	Console.WriteLine("A kenar uzunluğu: {0}", ucgen.A);

Yukarıdaki kod parçasında üçgenin a, b ve c kenarlarına özelliklerin set metodu üzerinden değer ataması yapılabildiği ancak ekrana yazdırma esnasında get metodu olmayan A özelliğinden bir değer döndürülemediği görülmektedir.

Kod editöründe bu hata mesajı Görsel 3.6'daki gibi görülmektedir.



Görsel 3.6: Sadece yazılabilir özelliği okumaya çalışma

3.7. METOTLAR (METHODS)

Bir metot, yalnızca çağrıldığında çalışan ve bir dizi ifade içeren kod blokudur. Yazılım dünyasında bir metot, sınıf içinde yapılacak işlerin veya operasyonların tanımlanmasını sağlar. Metotlara parametreler aracılığıyla ana programdan veriler gönderilebilir ve metotlar çalışmasını bitirdikten sonra ana programa değer döndürülebilir.

Genel olarak bir metot aşağıdaki şekilde tanımlanmaktadır.

```
«Erişim belirleyici» «Dönüş tipi» «Metodun adı» («Parametre listesi»)
{
```



SINIFLAR(CLASS)

BÖĞRENME BİRİMİ

Erişim Belirleyici: Metoda nerelerden erişilebileceğini tanımlar.

• **Dönüş Tipi:** Metotlar değer döndürebilir. Dönüş tipi, metodun döndüreceği değerin tipini tanımlar (int, string vb.). Değer döndürmek için **return** anahtar kelimesi kullanılır. Metot değer döndürmeyecekse dönüş tipi olarak **void** anahtar kelimesi kullanılır.

• Metot Adı: Metodun adını tanımlamada kullanılır.

• Parametre Listesi: Parantez içinde verilen parametreler, bir metoda değer göndermek veya me-

tottan değer almak için kullanılır. Parametrelerin türü, sayısı ve sırası parametre listesi olarak adlandırılır. Bunlardan metodun adı ve parametre listesi **metodun imzası**, küme parantezi { } arasına yazılan

kodlar da **metodun gövdesi** olarak adlandırılır. Metot imzaları, bir sınıf içinde her metotta farklı olmak zorundadır.

Metot adı + Parametre listesi = Metodun imzası

Aşağıda örnek metot tanımları verilmiştir.

public void ProgramiKapat() { ... }

public void DaireCiz(double x, double y, double cap) { ... }

public int KareKokHesapla(int sayi) { ... };

public void SMSGonder(string cepNo, string mesaj) { ... }

public decimal MaasHesapla(int gunSayisi, decimal gundelikUcret) { ... }

Bir metodu ana programdan çağırmak için metodun adı ile () parantez açma ve kapatma karakterleri kullanılmalıdır.

10. SINIF

nesneAdi.ProgramiKapat();

nesneAdi.DaireCiz(3.2, 2.4, 3);

int kareKok = nesneAdi.KareKokHesapla(9);

nesneAdi.SMSGonder("5051234567", "Merhaba, nasılsın?");

decimal maas = nesneAdi.MaasHesapla(24, 90.25M);

Verilen iki sayıdan büyük olanını bulan bir metot aşağıda yazılmıştır.

```
class SayiBulucu
   public int BuyukOlaniBul(int sayi1, int sayi2)
      int sonuc;
     if (sayi1 < sayi2)
             sonuc = sayi2;
      else
             sonuc = sayi1;
      return sonuc;
   }
class Program
ł
   static void Main(string[] args)
      SayiBulucu sb = new SayiBulucu();
      int a = 100;
      int b = 25;
      int sonuc = sb.BuyukOlaniBul(a, b);
      Console.WriteLine("Büyük olan sayı: {0}", sonuc);
   }
}
```

NESNE TABANLI PROGRAMALAMA

SINIFLAR(CLASS)
-----------	--------



SayiBulucu adlı sınıfın içindeki metotta;

- Erişim belirleyicisi "public",
- Dönüş tipi "int",
- Metot adı "BuyukOlaniBul",
- Parametre listesi "(int sayi1, int sayi2)",
- Metot imzası "BuyukOlaniBul(int sayi1, int sayi2)" olduğu söylenebilir. Metottan değer döndürmek için return anahtar kelimesi kullanılır.

Sıra Sizde >

- 1. Parametresinde verilen sayının değeri tek ise true, çift ise false döndüren metodu yazınız.
- 2. Klavyeden 0 (sıfır) girilene kadar bu değerleri toplayıp döndüren metodu yazınız.

3.7.1. Varsayılan Değerli Parametreler (Optional Parameters)

Metot parametreleri tanımlanırken istendiğinde bunlara "varsayılan değerler" atanabilmektedir. Metot çağrılırken bu parametrelere değer ataması yapılmazsa varsayılan değerler kullanılır.



Birden fazla parametre kullanıldığında varsayılan değere sahip parametreler, parametre listesinin en sonunda yer almalıdır. Aksi takdirde derleyici hatası oluşur (Görsel 3.7).

<pre>public void Metodum(int a, string ad = "Ahmet",</pre>	int þ)	
{	🢡 🕶	(parameter) int b
Console.WriteLine("Merhaba " + ad);		
}		CS1737: Optional parameters must appear after all required parameters
ferences		Show potential fixes (Alt+Enter or Ctrl+.)

Görsel 3.7: Varsayılan parametrelerin hatalı kullanımı





3.7.2. İsimlendirilmiş Parametreler (Named Parameters)

Bir metot oluşturulurken tanımlanan parametrelere değer atamak için ana programda parametreler sırasıyla yazılmalıdır. İstenilirse parametre isimleri kullanılarak bu sıralamaya uyulmayabilir.

```
class SayiIslemleri
{
    public int Topla(int sayi1, int sayi2, int sayi3)
    {
        return sayi1 + sayi2 + sayi3;
    }
}
class Program
{
    static void Main(string[] args)
    {
        class SayiIslemleri
            SayiIslemleri si = new SayiIslemleri();
        // int toplam = si.Topla(5, 10, 15);
        int toplam = si.Topla(sayi2: 10, sayi3: 15, sayi1: 5);
        Console.WriteLine("Toplam: {0}", toplam);
    }
}
```

Yukarıdaki kodda metot çağrılırken parametre adının ardından : (iki nokta) karakteri kullanılarak parametre sırasına uymadan atama yapıldığı görülmektedir.

3.7.3. Parametre Dizileri

nımlanır.

Parametre, metodun parametre sayısının bilinmediği durumlarda params anahtar kelimesi ile ta-

```
class SayiIslemleri
Ł
    public int Toplam(params int[] sayilar)
       int toplam = 0;
       foreach (var s in sayilar)
           toplam += s;
       return toplam;
   }
}
class Program
ł
   static void Main(string[] args)
       SayiIslemleri si = new SayiIslemleri();
       Console.WriteLine("Toplam: {0}", si.Toplam(3));
Console.WriteLine("Toplam: {0}", si.Toplam(3, 4, 5));
Console.WriteLine("Toplam: {0}", si.Toplam(5, 1, 7, 3, 4, 5));
   }
}
// Ekran çıktısı:
Toplam: 3
Toplam: 12
Toplam: 25
```



SINIFLAR(CLASS)



params anahtar kelimesi ile bir parametre tanımlanacaksa bu parametre, metodun en son parametresi olmalıdır. Aksi takdirde derleyici hata verecektir. Ayrıca metotlarda sadece bir adet params tipinde parametre tanımlanabilir.

3.7.4. Metodu Sonlandırma

Dönüş tipi **void** olan bir metodun çalıştırılması, istenildiği an sonlandırılabilir. Bunun için **return** anahtar kelimesi kullanılır.

```
class EkranIslem
{
   public void EkranaYaz(params int[] sayilar)
      if (sayilar.Length == 0)
      ł
         Console.WriteLine("Parametre olmadığı için metottan çıkılıyor.");
         return;
      Console.WriteLine("Parametreden gelen değerler:");
      foreach (var s in sayilar)
      ł
         Console.WriteLine(s);
      }
   }
}
class Program
   static void Main(string[] args)
   {
      EkranIslem ei = new EkranIslem();
      ei.EkranaYaz(3, 4, 5);
Console.WriteLine("============"");
      ei.EkranaYaz();
   }
}
// Ekran ciktisi:
Parametreden gelen değerler:
3
4
5
                          _____
Parametre olmadığı için metottan çıkılıyor.
```

Yukarıdaki kod blokunda **EkranaYaz** metodunun içinde **return** ifadesi ile metodun çalıştırılması sonlandırılmıştır.

Sıra Sizde >

Yukarıdaki kod blokunda EkranaYaz metodunda verilen parametrelerden herhangi biri 0 (sıfır) ise programın çalışmasını sonlandıran değişikliği yapınız.





3.7.5. Metot Aşırı Yüklemeleri (Method Overloads)

Metodun adı aynı kalmak şartıyla parametre tipleri ve/veya sayısı değiştirilerek farklı metot imzaları oluşturulabilir. Bu durumda aynı isimli birden fazla metot oluşacaktır. Aşağıdaki sınıfta farklı türden parametreler alan **Topla** isimli metodun birden fazla kez oluşturulduğu görülmektedir.

```
class ToplamaIslemi
  public int Topla(int a, int b)
     Console.WriteLine("int parametreli metot cağrılıyor.");
    return a + b;
  }
  public int Topla(params int[] sayilar)
     Console.WriteLine("params parametreli metot çağrılıyor.");
    int toplam = 0;
    foreach (var s in sayilar)
     ł
       toplam += s;
     return toplam;
  }
  public double Topla(double a, double b)
     Console.WriteLine("double parametreli metot çağrılıyor.");
     return a + b;
  }
  public string Topla(string a, string b)
     Console.WriteLine("string parametreli metot çağrılıyor.");
     return a + b;
  }
class Program
  static void Main(string[] args)
  ToplamaIslemi ti = new ToplamaIslemi();
    Console.WriteLine(ti.Topla(2, 5));
Console.WriteLine(``========
                                  Console.WriteLine(ti.Topla(3.3, 5.1));
    Console.WriteLine(
                                        Console.WriteLine(ti.Topla("Sağlıcakla ", "kalınız."));
    Console.WriteLine(
    Console.WriteLine(ti.Topla(3, 8, 3, 7, 12, 33, 11, 4));
 }
}
// Ekran çıktısı:
int parametreli metot çağrılıyor.
______
double parametreli metot çağrılıyor.
8,4
_____
string parametreli metot çağrılıyor.
Sağlıcakla kalınız.
_____
params parametreli metot çağrılıyor.
81
```



Her bir Topla metodunun imzası farklı olduğu için derleyici hata vermemekte ve program başarılı bir şekilde çalıştırılmaktadır.



Görsel 3.8: Metot aşırı yüklemelerini kullanma

Burada Topla metodunun dört farklı aşırı yüklenmiş hâli olduğu belirtilmektedir. Klavyeden yukarı ve aşağı tuşları ile parametre tipleri incelenebilir.

Sıra Sizde

Yaş hesaplayan bir metodu DateTime (doğum tarihi) ve int (doğum yılı) parametrelerini alacak şekilde aşırı yükleyerek gerçekleştiriniz.

3.8. YAPICI VE YIKICI METOTLAR

Nesneler oluşturulduğunda ve yok edilme anında otomatik olarak çalıştırılan metotlar bulunmaktadır. Nesneler oluşturulurken otomatik olarak çalıştırılan metotlara **yapıcı metot** (constructor), nesnelerin yok edildiği anda otomatik olarak çalıştırılan metotlara **yıkıcı metot** (destructors) denilmektedir.

3.8.1. Yapıcı Metotlar (Constructors)

Yapıcı metotlar, nesnelerin ilk oluşturulduğu anda otomatik olarak çalıştırılır. Yapıcı metotlar genellikle sınıf içinde tanımlanan yerel değişkenlerin ilk değerlerini düzenlemek için kullanılır.

Bir metodun yapıcı metot olabilmesi için şu şartları taşıması gerekir:

- Metot adı, sınıfın adı ile aynı olmalıdır.
- Geri dönüş tipi olmamalıdır (void ya da int gibi).

Nesneleri oluşturmak için **new** operatörü kullanıldığı anda yapıcı metotlar otomatik olarak çalıştırılır. Nesne oluşturulduktan sonra yapıcı metotlar bir daha çağrılamaz.

Not: Sınıf içinde bir yapıcı metot tanımlanmamışsa derleyici arka planda boş bir varsayılan yapıcı metot oluşturmaktadır.

Not: Yapıcı metotlar bazı kaynaklarda "kurucu metot" veya "oluşturucu metot" olarak da geçmektedir.



		SINIFLAR(CLASS)
3.	ÖĞRENME BİRİMİ	

Aşağıda Kisi sınıfında tanımlanan iki adet yerel değişkene yapıcı metot içinde değer ataması yapılmıştır.

```
class Kisi
{
   int yas;
   string ad;
   public Kisi()
    ł
       yas = 19;
ad = "Ahmet";
       Console.WriteLine("Yapıcı metot çalıştı.");
   }
   public int Yas
    {
       get
{
           return yas;
       }
   }
   public string Ad
       get
       Ĭ
          return ad;
       }
   }
}
class Program
ł
   static void Main(string[] args)
    {
       Console.WriteLine("Program başladı.");
       Kisi k = new Kisi();
Console.WriteLine("Adı: {0}, Yaşı: {1}", k.Ad, k.Yas);
Console.WriteLine("Program bitti.");
   }
}
// Ekran çıktısı:
Program başladı.
Yapıcı metot çalıştı.
Adı: Ahmet, Yaşı: 19
Program bitti.
```

Yukarıdaki kod parçasında **new Kisi()** komutu işletildiği anda yapıcı metodun çalıştırıldığı görülmektedir. Yapıcı metotları da aşırı yüklemek mümkündür.



SINIFLAR(CLASS)

ÖĞRENME BİRİMİ

Yukarıdaki **Kisi** sınıfı şu şekilde de tanımlanabilir.

class Kisi int yas = 0; string ad = ""; public Kisi() { yas = 19; ad = "Ahmet"; Console.WriteLine("Yapıcı metot çalıştı."); } public Kisi(int yas) ł this.yas = yas; ad = "Ahmet" Console.WriteLine("int parametreli yapıcı metot çalıştı."); } public Kisi(string ad) yas = 19; this.ad = ad; Console.WriteLine("string parametreli yapıcı metot çalıştı."); } public Kisi(int yas, string ad) this.yas = yas; this.ad = ad; Console.WriteLine("İki parametreli yapıcı metot çalıştı."); } } class Program { static void Main(string[] args) { Console.WriteLine("Program başladı."); Kisi k1 = new Kisi(); Kisi k2 = new Kisi(23); Kisi k3 = new Kisi("Filiz"); Kisi k4 = new Kisi(25, "Süleyman"); Console.WriteLine("Program bitti."); } } // Ekran ciktisi: Program başladı. Yapıcı metot çalıştı. int parametreli yapıcı metot çalıştı. string parametreli yapıcı metot çalıştı. İki parametreli yapıcı metot çalıştı. Program bitti.

Yapıcı metot içinde kullanılan **this** anahtar kelimesi, bu sınıftan türeyen nesneyi temsil etmektedir. Dolayısıyla parametre adı ile sınıf değişkeninin adlarının aynı olması durumunda **this** anahtar kelimesi ile bu sınıftan türetilen nesnenin ilgili değişkenini kullanmak mümkündür.

public Kisi(int yas)
{
 this.yas = yas;
}



Parametreden gelen **yas** bilgisi, **"this.yas"** ifadesi ile bu sınıftan türetilen nesnenin **yas** alanına atanmaktadır.

Sıra Sizde

Daire sınıfını yapıcı metodu ile beraber tanımlayınız (Yarıçap değerini almalıdır.).

Sıra Sizde

Ev sınıfını yapıcı metodu ile beraber tanımlayınız (Oda sayısı ve m² değerlerini almalıdır.).

3.8.2. Yıkıcı Metotlar (Destructors)

Nesne hafızadan atıldığı anda otomatik olarak çalışan yıkıcı metotlar, tıpkı yapıcı metotlar gibi özel metotlardır ve aşağıdaki şu şartları taşımalıdır:

- Metot adı, sınıfın adı ile aynı olmalıdır.
- Metot adının başında ~ (Tilde) karakteri olmalıdır.
- Bir sınıfın yalnızca bir tane yıkıcı metodu olabilir.
- Yıkıcı metotlar aşırı yüklenemez.
- Yıkıcı metotlar parametre alamaz.
- Yıkıcı metotların erişim belirleyicisi olamaz.

Programcının yıkıcı metot üzerinde bir kontrolü bulunmamaktadır. Yıkıcı metotlar genellikle sınıf içinde kullanılan kaynakların (veri tabanı, dosya vb.) kapatılması ve hafizadan atılması amacıyla .**NET Framework** içindeki **Garbage Collector** (Çöp Toplayıcısı) tarafından <u>gerekli görüldüğü</u> zaman çalıştırılır.

Aşağıda yıkıcı metoda sahip bir sınıf örneği verilmiştir.

```
class Otomobil
     string marka = "";
string renk = "";
public Otomobil()
         marka = "TOGG";
          renk = "kırmızı"
          Console.WriteLine("Yapıcı metot çalıştı.");
      }
      ~Otomobil()
      {
          Console.WriteLine("Nesne hafizadan atıldı.");
      }
  }
  class Program
      static void Main(string[] args)
     Console.WriteLine("Program başladı.");
Otomobil oto = new Otomobil();
Console.WriteLine("Program bitti.");
  3
// Ekran çıktısı:
Program başladı.
Yapıcı metot çalıştı.
Program bitti.
Nesne hafızadan atıldı.
```

Yıkıcı metot, ekran çıktısından da görüleceği gibi programın sona ermesinden sonra çalıştırılmaktadır.

Sıra Sizde

Yıkıcı metotların kullanım yerlerini araştırınız, edindiğiniz bilgileri sınıf arkadaşlarınızla paylaşınız.



3.9. DEĞER VE REFERANS TİPLER

ÖĞRENME BİRİMİ

.NET platformunda hafıza yönetiminin nasıl işlediğinin bilinmesi, programcılar için oldukça önemlidir. .NET'te hafıza, **yığın** (stack) ve **öbek** (heap) olmak üzere iki bölgeye ayrılmıştır.

Değişkenler, veri tipine göre atanan değerleri taşıyan veri tutuculardır. .NET platformunda kullanılan her bir veri tipi **değer tipli** ve **referans tipli** olarak ikiye ayrılmaktadır. Bu veri tiplerinden değer veri tipleri hafızanın yığın bölgesinde, referans veri tipleri de hafızanın öbek bölgesinde tutulur.

- Değer Tipleri: int, long, float, double, decimal, char, bool, byte, short, struct, enum
- Referans Tipleri: string, object, class, interface, array, delegate

Not: Bunlardan **string** veri tipi teorikte referans tipli olmasına rağmen program içinde değer tipli olarak işlem görmektedir.

Değer tipleri veriyi bizzat barındıran türlerdir, referans tipleri ise veri yerine verinin bellekteki adresini tutan türlerdir (Görsel 3.9).



Görsel 3.9: Yığın ve öbek ilişkisi

Değer tiplerinden biri kullanılarak bir değişken tanımlandığında değişkenin değeri yığın bellek bölgesinde tutulur. Referans tipte bir değişken tanımlandığında ise değişkenin değeri öbek hafıza bölgesinde tutulur ve bu bölgenin adresini tutan bilgi de yığında saklanır. Yığında öbek bölgesini işaret eden bir "işaretçi" (pointer) oluşturulur.







Aşağıdaki örnekte değer ve referans tiplerin çalışma mantıkları verilmiştir.



Main metodu çalıştırıldığında sırasıyla şu işlemler gerçekleşir:

- int sayi1 = 10; => sayi1 için yığında alan ayrılır ve buraya 10 değeri yazılır.
- int sayi2 = sayi1; => sayi2 için yığında alan ayrılır ve bu alana sayi1'in değeri 10 yazılır.
- **sayi2 = 50;** => sayi2 için ayrılan alana 50 değeri yazılır. sayi1 değişkeninin değeri değişmez.

• **SayiTutucu st1 = new SayiTutucu();** => Oluşturulan nesne için öbekte bir alan ayrılır ve nesnenin ögeleri burada saklanır. Bu alanın adresi st1 değişkeninde tutulur. Bu değişken için de yığında bir yer ayrılır.

• **st1.A = 10;** => st1'in gösterdiği öbek alanındaki nesnenin A özelliğinin değeri 10 olarak güncellenir.

• **SayiTutucu st2 = st1;** => st2 için yığında ayrı bir yer ayrılır ve buraya st1 değişkeninin tuttuğu

adres bilgisi yazılır. Dolayısıyla st1 ve st2 aynı öbek alanının adresini tutmaktadır.

• **st2.A = 50;** => st2'nin gösterdiği öbek alanındaki nesnenin A özelliğinin değeri 50 olarak güncellenir.

st1 ve st2 aynı öbek alanının adresini tuttuğu için hangi değişken üzerinden olduğu fark etmeksizin aynı nesnenin A özelliğinin değeri ekrana yazdırılacaktır (Görsel 3.10).



Görsel 3.10: Aynı referansa sahip nesneler





3.9.1. Metotlarda ref ve out Kullanımı

Değer tipli değişkenler metotlara parametre olarak gönderildiğinde bu değişkenin değeri için yığında farklı bir bellek alanı ayrılır. Dolayısıyla metot içinde bu değişkenin değeri değiştirilse bile değişiklik ana programdan gönderilen değişkeni etkilemeyecektir. Aşağıdaki kod parçasında bu durum verilmiştir.

```
class Matematik
   public void Artir(int x)
      x++:
   }
}
class Program
   static void Main(string[] args)
   {
      Matematik m = new Matematik();
      int a = 100;
      m.Artir(a);
      Console. WriteLine ("a değeri = \{0\}", a);
   }
}
// Ekran çıktısı:
a değeri = 100
```

Artir metodunun içinde x değişkeninin değeri 1 artırılmasına rağmen bu değişikliğin ana metot içinde kullanılan a değişkeninin değerine bir etkisi olmamaktadır çünkü ana metot içindeki a değişkeni ile Artir metodunda kullanılan x değişkeni için yığında farklı hafıza alanları ayrılır ve bunlar birbirlerinden tamamen ayrı iki değişken olarak düşünülmelidir.

Parametre olarak gönderilen değişkenin değeri Artir metodunun içinde değiştirilmek istenirse bu durumda **ref** veya **out** anahtar kelimeleri kullanılmalıdır.

Bir önceki kod parçacığı ref anahtar kelimesi kullanılarak tekrar yazılırsa ana metot içindeki değişkenin değierinin değiştiği görülecektir.

```
class Matematik
{
    public void Artir(ref int x)
    {
        x++;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Matematik m = new Matematik();
        int a = 100;
        m.Artir(ref a);
        Console.WriteLine("a değeri = {0}", a);
    }
}
// Ekran çıktısı:
    a değeri = 101
```

Yukarıdaki kodda Artir metodu yazılırken ve bu metot çağrılırken ref anahtar kelimesinin kullanıldığına dikkat edilmelidir. ref veya out anahtar kelimeleri ile metoda parametre gönderildiğinde aslında değişkenin değeri değil, değişkenin yığında bulunduğu hafıza adresi gönderilmektedir. Dolayısıyla metot içinde değişken üzerinde yapılan değişiklikler aslında ana metotta kullanılan değişken üzerinde gerçekleşmektedir.



SINIFLAR(CLASS)

ÖĞRENME BİRİMİ

out anahtar kelimesinin kullanımı, ref anahtar kelimesinin kullanımından biraz farklıdır. ref kullanılmak istendiğinde parametre olarak göndermeden önce değişkene mutlaka bir değer ataması yapılmalıdır. out kullanıldığında ise değişkenin değeri artırma veya azaltma gibi bir işleme tabi tutulmadan önce değişkene mutlaka değer ataması yapılmalıdır.

Yukarıdaki kod parçası out anahtar kelimesi ile tekrar yazılmıştır.

```
class Matematik
{
   public void Artir(out int x)
      x = 123;
   }
}
class Program
   static void Main(string[] args)
   {
      Matematik m = new Matematik();
      int a;
      m.Artir(out a);
      Console.WriteLine("a değeri = \{0\}", a);
   }
}
// Ekran çıktısı:
a değeri = 123
```

Tablo 3.1'de ref ve out anahtar kelimeleri arasındaki farklar listelenmiştir.

ref	out
Metodu tanımlarken parametrenin önüne	Metodu tanımlarken parametrenin önüne "out"
"ref" yazılmalıdır.	yazılmalıdır.
Metodu çağırırken değişkenin önüne "ref"	Metodu çağırırken değişkenin önüne "out" yazılma-
yazılmalıdır.	lıdır.
Metoda göndermeden önce değişken başlan-	Metoda göndermeden önce değişken başlangıç
gıç değeri almak zorundadır.	değeri almak zorunda değildir.
Metot içinde istenildiği gibi kullanılabilir.	Metot içinde mutlaka bir değer ataması gerçekleş- tirilmelidir.

Tablo 3.1: ref ile out Arasındaki Farklar

Sıra Sizde >

Siz de ref veya out kullanarak ad, soyad ve yaş bilgilerini döndüren metodu yazınız.

3.10. KALITIM (INHERITANCE)

Kalıtım, NTP'deki en önemli kavramlardan biridir ve bir sınıfın özelliklerinin farklı sınıflar tarafından da kullanılabilmesini sağlar. Buna "miras alma" da denilmektedir. Bu durumda miras alınan sınıfa üst veya temel (parent) sınıf, miras alan sınıfa da türetilmiş (derived) sınıf denir. C#'ta bir sınıf sadece bir sınıftan türetilebilir.

Bu durumun tek istisnası **Object** sınıfıdır. Bir sınıf, başka bir sınıftan türesin veya türemesin, varsayılan olarak **Object** sınıfından türer. Buna "örtük devralma" denir. Boş bir sınıftan oluşturulan nesnelerin sahip olduğu metotlar **Object** sınıfından gelmektedir (Görsel 3.11).





Görsel 3.11: Object sınıfından miras alınan ögeler

Üst sınıf ile türetilmiş sınıf arasında bir üst / alt ilişkisi vardır. Türetilmiş sınıf, üst sınıf ögelerine erişebilir ancak bu durumun tersi doğru değildir.

Bir sınıfı bir başka sınıftan türetmek için : (iki nokta) karakteri kullanılır.

«Erişim belirleyici» class «Sınıf adı» : «Üst sınıf adı» { }

Aşağıda OkulPersoneli sınıfı ve bu sınıftan türetilen bir Ogretmen sınıfı tanımlanmıştır.

```
public class OkulPersoneli
   public string Ad { get; set; }
   public string Soyad { get; set; }
}
public class Ogretmen : OkulPersoneli
   public string Brans { get; set; }
}
class Program
{
   static void Main(string[] args)
   {
      Ogretmen ogr = new Ogretmen
      ł
        Ad = "Ahmet",
        Soyad = Oz'',
        Brans = "Matematik"
      };
      // ..
   }
```

Bu kod parçasında Ad ve Soyad özellikleri Ogretmen sınıfına aktarılmıştır. Ogretmen sınıfı, Ad ve Soyad bilgilerini OkulPersoneli sınıfından miras almıştır.

Miras alma işlemlerinde "Her ... bir ... dır." mantığı bulunmaktadır. Yukarıdaki kod parçası için şu ifade kurulabilir: "Her öğretmen bir okul personelidir."





Her öğretmen, bir okul personeli olduğuna göre aşağıdaki kod satırı hata vermeyecektir.



Hafızanın öbek bölgesinde Ogretmen nesnesi bulunmasına rağmen per değişkeni üzerinden sadece Ad ve Soyad özelliklerine erişim mümkündür (Görsel 3.12).



Görsel 3.12: Referansın tipine bağlı nesneye erişim

Sıra Sizde

Çevrenizdeki nesneleri kalıtım açısından inceleyiniz ve en az üç tane kalıtım örneği gerçekleştiriniz.

3.10.1. Hiyerarşik Kalıtım

Bir sınıf, türetilmiş sınıflardan kalıtım yoluyla aynı şekilde türetilebilir. Bir anlamda hiyerarşik kalıtım mümkündür.

```
public class Canli
{
//...
public class Hayvan : Canli
{
//...
public class Kopek : Hayvan
{
//...
public class Kangal : Kopek
{
//...
}
```



	SINIFLAR(CLASS)
ÖĞRENME	
Do BİRİMİ	

Yukarıdaki sınıf tanımlamaları geçerlidir ve hiyerarşik kalıtıma bir örnektir. "Her ... bir ... dır." mantığı her bir sınıf için geçerlidir.

3.10.2. new Operatörüyle Metot Gölgeleme (Shadowing)

Bir sınıftan başka bir sınıf türetildiğinde özel bir durum ortaya çıkar. Üst sınıfta bulunan bir metot, alt sınıfta da tanımlanmışsa derleyici bir "uyarı" verir. Bu durumda üst sınıfta yer alan metot gölgelenerek erişilemez duruma gelir.

```
public class OkulPersoneli
{
    public string Ad { get; set; }
    public string Soyad { get; set; }

    public void AdSoyadYazdir()
    {
        Console.WriteLine("Benim adım soyadım : " + Ad + " " + Soyad);
    }

public class Ogretmen : OkulPersoneli
{
    public string Brans { get; set; }

    public void AdSoyadYazdir()
    {
        Console.WriteLine("Benim adım soyadım : " + Ad + " " + Soyad);
    }
}
```

Yukarıdaki kod parçasında **Ogretmen** sınıfındaki **AdSoyadYazdir()** metodu, üst sınıftaki aynı isimli metodu gölgelemektedir. Hem üst sınıfta hem de türetilmiş sınıfta aynı isimli metot bulunduğu için derleyici, alt sınıftaki metodun **new** operatörü ile tanımlanması gerektiğini belirten bir "uyarı" (hata değil) verir (Görsel 3.13).

public class Ogre	etmen : OkulPersoneli
{	
0 references	
public string	g Brans { get; set; }
1 reference	
public void /	\dSoyadYazdir()
{	void Ogretmen.AdSoyadYazdir()
}	CS0108: 'Ogretmen.AdSoyadYazdir()' hides inherited member 'OkulPersoneli.AdSoyadYazdir()'. Use the new keyword if hiding was intended.
}	Show potential fixes (Alt+Enter or Ctrl+.)

Görsel 3.13: Derleyici uyarı mesajı

Bu bir hata olmadığı için program çalışacaktır. Ancak bu uyarı mesajını ortadan kaldırmak için bu metodun kasıtlı olarak yazıldığını derleyiciye bildirmek gerekmektedir. Bunun için **new** anahtar kelimesi kullanılır.





SINIFLAR(CLASS) ÖĞRENME

Bu durumda derleyici herhangi bir uyarı ya da hata mesajı vermeyecektir. Ancak bu kullanımın NTP prensiplerine uygun olduğu söylenemez. O nedenle bu gibi durumlarda "sanal metot"ların kullanılması uygun olacaktır.

3.10.3. Sanal Metotlar (Virtual Methods)

Temel sınıftan türetilmiş alt sınıflara aktarılan metotlar her zaman olduğu gibi kullanılmayabilir. İstenilen metotlar alt sınıflarda tekrardan yazılabilir. Böyle bir durumda bu metotlar, üst sınıfta **virtual** (sanal), alt sınıflarda da **override** (geçersiz kılma veya ezme) anahtar kelimeleri kullanılarak tanımlanmalıdır.

Sanal olarak tanımlanan metotlar, alt sınıflarda geçersiz kılınmak zorunda değildir. Sanal metotlar geçersiz kılınmazsa metodun kendisi, sanal metotlar geçersiz kılınırsa alt sınıfın metodu çağrılır.

```
public class Sekil
ł
   public const double pi = 3.14;
   protected double x, y;
   public Sekil()
  public Sekil(double x, double y)
     this.x = x;
     this.y = y;
public virtual double AlanHesapla()
      return x * y;
  }
  public virtual void BilgiYazdir()
      Console.WriteLine("x = " + x + " ve y = " + y);
  }
}
public class Daire : Sekil
   public Daire(double r) : base(r, 0)
   public override double AlanHesapla()
     return pi * x * x;
  }
}
class Program
ł
  static void Main(string[] args)
   {
     Sekil s = new Sekil(3, 4);
     s.BilgiYazdir();
     Console.WriteLine("Sekil alanı: " + s.AlanHesapla());
     Console.WriteLine("=========================");
     Daire d = new Daire(1.3);
     d.BilgiYazdir();
     Console.WriteLine("Daire alan: {0:N2}", d.AlanHesapla());
  }
}
```

SINIFLAR(CLASS)



// Ekran çıktısı:
x = 3 ve y = 4
Şekil alanı: 12
=======================================
x= 1,3 ve y= 0
Daire alani: 5,31

Not: public Daire(double r) : base(r, 0)

Daire sınıfının alan hesabı, Sekil sınıfının alan hesabından farklıdır. Yukarıdaki programda da görüldüğü gibi AlanHesapla() metodu alt sınıfta geçersiz kılınmış ancak BilgiYazdir() metodu alt sınıfta geçersiz kılınmamıştır.

Yukarıdaki kod blokunda Daire sınıfının yapıcı metodu tanımlanırken nesne oluşturulduğu anda aynı zamanda üst sınıfın yapıcı metodunun da çağrılması sağlanmıştır. Bunun için base anahtar kelimesi kullanılmaktadır. this anahtar kelimesi sınıfı, base anahtar kelimesi ise üst sınıfı temsil etmektedir.

Not: Sınıf içinden üst sınıfın ögelerine base anahtar kelimesi ile erişim mümkündür. Yukarıdaki örnek için "base.x" ifadesi ile Sekil sınıfının x alanına erişilebilir.

Sıra Sizde >

{

Kitap, dergi ve ansiklopediler için bir üst sınıf yazarak Oku() sanal metotlarını ihtiva eden sınıflar tanımlayınız.

3.11. SOYUT SINIFLAR (ABSTRACT CLASSES)

NTP'nin bir diğer önemli kavramı da soyutlamadır. Soyutlama genellikle ortak özellikleri olan sınıfları bir çatı altında toplamak için kullanılır. Soyut sınıfların klasik sınıflardan en önemli farkı, **new** anahtar kelimesi ile nesnelerinin oluşturulamamasıdır.

Soyut sınıflar **abstract** anahtar kelimesi ile tanımlanmalı ve en az bir tane **abstract** ile tanımlanmış metodu olmalıdır. Bu metodun sadece imzası bulunur, gövdesi bulunmaz. Ayrıca soyut olarak tanımlanmış metotlar, bu sınıftan türeyen alt sınıflarda mutlaka geçersiz kılınmalıdır (override).

Not: Sanal metotlar geçersiz kılınmak zorunda değildir. Buna karşın soyut metotlar <u>mutlaka</u> geçersiz kılınmalıdır.




```
public abstract class MotorluArac
   public int MotorHacmi { get; set; }
  public int ModelYili { get; set; }
  public abstract void Calis();
  public abstract void Dur();
}
public class Otomobil : MotorluArac
ł
  public bool OtomatikVites { get; set; }
  public override void Calis()
     Console.WriteLine("Otomobil calisti.");
  }
  public override void Dur()
  ł
     Console.WriteLine("Otomobil durdu.");
  }
class Program
Ł
   static void Main(string[] args)
{
     // ** Alttaki satır hata verir.
     // ** Sanal sınıflardan nesne türetilemez.
     // MotorluArac ma = new MotorluArac();
   Otomobil oto = new Otomobil
     {
        ModelYili = 2020,
        MotorHacmi = 1600,
        OtomatikVites = true
     };
     oto.Calis();
     oto.Dur();
  }
}
// Ekran çıktısı:
Otomobil çalıştı.
Otomobil durdu.
```

Her motorlu araçta bulunan ortak özellikler MotorluAraclar sınıfında, otomobillere özgü özellikler de Otomobil sınıfında tanımlanmıştır.

MotorluAraclar sınıfındaki iki soyut metot, Otomobil sınıfında geçersiz kılınmıştır (Geçersiz kılınmak zorundadır.).





	SINIFLAR(CLASS)
ÖĞRENME BİRİMİ	

"Her ... bir ... dır." mantığı burada da geçerlidir. "Her otomobil bir motorlu araçtır." ifadesi doğru olduğuna göre aşağıdaki kod hata vermeyecektir.

```
Otomobil oto = new Otomobil
{
	ModelYili = 2020,
	MotorHacmi = 1600,
	OtomatikVites = true
};
	MotorluArac ma = oto;
	ma.Calis();
	ma.Dur();
```

Sıra Sizde >

Kitap, dergi ve ansiklopediler için bir soyut üst sınıf yazarak Oku() metotlarını ihtiva etmek <u>zorunda</u> <u>olan</u> sınıfları tanımlayınız.

3.12. ARAYÜZLER (INTERFACES)

NTP'de soyutlamanın bir başka yolu da **arayüzler** (interfaces) aracılığıyla mümkündür. Bir arayüz, tüm ögeleri soyut olan bir sınıfa benzetilebilir ancak burada arayüzlerin ve sınıfların farklı kavramlar olması önemli bir husustur. Bir sınıf sadece bir sınıftan türetilebiliyor iken birden fazla arayüzden türetilebilir.

Arayüzün içinde tanımlanan metotların sadece imzaları bulunur, gövdeleri bulunmaz. Ayrıca arayüzde bulunan tüm metotlar varsayılan olarak soyuttur (abstract) ve genel (public) erişim belirleyicisine sahiptir.

Arayüz bir sınıf türü olmadığından içinde kod bloku bulunamaz. Arayüzde tanımlanan ögeler, kendisinden türetilen sınıfta mutlaka uygulanmak (implement) zorundadır.

Not: Soyut sınıflarda "geçersiz kılma" (override) kavramı, arayüzler için de "uygulamak" (implement) kavramı kullanılır.

Not: Arayüz isimlerinin sınıf olmadığını belirtmek için "I" (büyük ı) harfi ile başlatılması gelenektir.





Aşağıda türetildiği arayüzleri uygulayan bir sınıf tanımı görülmektedir.

.

•

interface IHayvan
void SesCikar();
}
{
<pre>> Void Besien(); }</pre>
public class Kedi : IHayvan, IBeslen
ຼີ public void SesCikar()
Console.WriteLine("Kedi: miyav"); }
public void Beslen()
Console.WriteLine("Kedi süt içti.");
public class Kopek : IHayvan, IBeslen
¹ public void SesCikar()
Console.WriteLine("Köpek: hav hav"); }
public void Beslen()
Console.WriteLine("Köpek et yedi.");
class Program
<pre>static void Main(string[] args) {</pre>
Kedi kedi = new Kedi(); kedi.SesCikar(); kedi.Beslen();
Kopek kopek = new Kopek(); kopek.SesCikar(); kopek.Beslen();
Console.WriteLine("======""); IHayvan hayvan1 = kedi; IHayvan hayvan2 = kopek; hayvan1.SesCikar(); hayvan2.SesCikar(); Console.WriteLine("======""); IBeslen beslen1 = kedi; IBeslen beslen2 = kopek; beslen1.Beslen(); beslen2.Beslen(); }

ÖĞRENME BİRİMİ	• •	
// Ekran çıktısı: Kedi: miyav Kedi süt içti. Köpek: hav hav Köpek et yedi.		
======================================		
====================================		

Hem Kedi hem de Kopek sınıfları, IHayvan ve IBeslen arayüzlerinden türetilmiştir. Dolayısıyla her iki sınıf da arayüzlerin içindeki metotları uygulamak zorundadır. Arayüzden türetilen sınıflar için soyut sınıflarda olduğu gibi "Her ... bir ... dır." ifadesi yine kullanılmaktadır ("Her kedi bir hayvandır.", "Her köpek bir hayvandır."). Tablo 3.2'de arayüzler ve soyut sınıflar arasındaki farklar listelenmiştir.

Tablo 3.2: Arayüzler ve Soyut Sınıflar Arasındaki Farklar

Arayüzler	Soyut Sınıflar
Bir sınıf birden fazla arayüzden türetilebilir.	Bir sınıf sadece tek bir soyut sınıftan türetilebilir.
Sadece boş (gövdesi olmayan) metotlar tanımla- nabilir.	Hem normal metot hem de boş metotlar tanımla- nabilir.
Çoklu kalıtım özelliği sağlar.	Çoklu kalıtım özelliği sağlamaz.
Tüm ögeler public olarak kabul edilir.	Ögeler public olmak zorunda değildir.
Yapıcı metot içeremez.	Yapıcı metot içerebilir.
Statik ögeler barındıramaz.	Statik ögeler barındırabilir.

Sıra Sizde 🔸

Kitap, dergi ve ansiklopediler için bir arayüz aracılığıyla Oku() metotlarını ihtiva etmek <u>zorunda olan</u> sınıfları tanımlayınız.

3.13. ÇOK BİÇİMLİLİK (POLYMORPHISM)

Çok biçimlilik NTP'deki bir diğer önemli prensiptir. Çok biçimlilik, aynı temel sınıftan veya arayüzden türetilmiş alt sınıflardaki metotların farklı şekillerde davranabilmesidir.Türetilen alt sınıflarda aşağıdaki durumlardan birinin sağlanması gerekir:

- Üst sınıftaki sanal ögeler geçersiz kılınır (virtual / override).
- Soyut sınıflarda soyut tanımlanan ögeler geçersiz kılınır (abstract / override).
- Arayüzlerdeki ögeler uygulanır (implementation).

Her üç yolla da çok biçimlilik sağlanabilir.





Aşağıdaki sınıflar için üç yolla da çok biçimliliğin sağlandığı görülmektedir.



```
class Sekil
ł
   public virtual void Ciz()
      Console.WriteLine("Şekil çizildi.");
  }
class Kare : Sekil
{
   public override void Ciz()
      Console.WriteLine("Kare çizildi.");
   }
class Daire : Sekil
{
   public override void Ciz()
    Console.WriteLine("Daire çizildi.");
  }
}
class Ucgen : Sekil
{
  public override void Ciz()
  {
      Console.WriteLine("Üçgen çizildi.");
  }
```

abstract / override





interface

```
interface Sekil
{
   void Ciz();
}
class Kare : Sekil
{
   Public void Ciz()
   {
       Console.WriteLine("Kare çizildi.");
    }
class Daire : Sekil
    public void Ciz()
       Console.WriteLine("Daire çizildi.");
   }
}
class Ucgen : Sekil
ł
    public void Ciz()
      Console.WriteLine("Üçgen çizildi.");
   }
 ì
```

Aşağıdaki programda çok biçimlilik uygulanmıştır.

class Program

```
{
  static void Main(string[] args)
   {
      List<Sekil> sekiller = new List<Sekil>
      {
         new Daire(),
         new Kare(),
         new Ucgen()
      };
      foreach (var sekil in sekiller)
      {
         sekil.Ciz();
      }
  }
}
// Ekran çıktısı:
Daire çizildi.
Kare çizildi.
Üçgen çizildi.
```

Burada önemli olan husus, "sekil.Ciz();" ifadesi ile oluşturulan farklı nesnelerin aynı isimli metodunun çağrılarak ilgili nesneye ait işlemlerin gerçekleştirilmesidir.

Sıra Sizde

Kitap, dergi ve ansiklopedileri tutan bir koleksiyon nesnesi oluşturunuz ve bu sınıfların içindeki Oku metodunu döngü kullanarak çağırınız.

SINIFL	.AR(C	LASS)
--------	-------	-------



3.14. STATİK SINIFLAR (STATIC CLASSES)

Statik (static) sınıflar temel olarak statik olmayan sınıflarla aynıdır ancak statik sınıflardan new anahtar kelimesi ile nesne türetilemez. Nesne türetilemediği için bu sınıfların ögelerine nesne adı üzerinden değil, doğrudan sınıf adı üzerinden erişilir. Sınıfın kendisi ya da içindeki bazı ögeler statik olarak tanımlanabilir. Sınıfın kendisi statik olarak tanımlanırsa sınıf içindeki tüm ögelerin statik olması zorunludur.

Statik ögeler genellikle nesnelerin durumuna göre değişmeyen verileri temsil etmede veya hesaplamaları yapmada kullanılır. Buna en güzel örnek, .NET Sınıf Kütüphanesi'ndeki Math sınıfıdır.

Math m1 = new Math(); // Hata Math m2 = new Math(); // Hata Console.WriteLine(m1.Sqrt(9)); Console.WriteLine(m2.Sqrt(9));

Bir sayının karekökü yukarıdaki kod yazılarak hesaplanmak istendiğinde **m1** ve **m2** nesneleri üzerinden hesaplamada farklılık olmayacağı için **Math** sınıfından bir nesne türetilmesine gerek yoktur. Dolayısıyla **Math** sınıfı **static** olarak tanımlanmıştır ve bu sınıftan nesne türetilemez. Bu yüzden yukarıdaki kod parçası hata verecektir. Karekök hesaplamak için kullanılan **Sqrt()** metoduna nesne oluşturmadan sınıf adı üzerinden erişilmektedir.

```
double sayi = Math.Sqrt(9);
Console.WriteLine(sayi);
```

Statik bir sınıf, herhangi bir ögesi ilk kullanıldığı anda hafızaya yüklenir ve programın çalışması sonlanana kadar hafızada durur. Bu yüzden statik sınıfın dikkatli kullanılması önerilir.

Bir sınıf **static** olarak tanımlanmadan içindeki herhangi bir öge **static** olarak tanımlanabilir. Ayrıca statik bir yapıcı oluşturmak da mümkündür. Aşağıdaki örnekte hem normal bir sınıf yapıcısı hem de statik yapıcı bir arada kullanılmıştır.

```
public class Ogrenci
  public int Numara { get; set; }
  public string AdSoyad { get; set; }
}
public class OgrenciIslem
  public List<Ogrenci> ogrenciler;
  public static int OgrenciSayisi { get; set; }
  static OgrenciIslem()
  {
     OgrenciSayisi = 0;
     Console.WriteLine("Statik yapıcı çalıştı.");
  }
  public OgrenciIslem()
     ogrenciler = new List<Ogrenci>();
     Console.WriteLine("Yapıcı çalıştı.");
}
  public void OgrenciEkle(Ogrenci ogr)
     ogrenciler.Add(ogr);
     OgrenciSayisi++;
     Console.WriteLine("Öğrenci eklendi.");
}
```





```
public void OgrenciSil(int numara)
  {
    var ogr = ogrenciler.FirstOrDefault(x => x.Numara == numara);
    if (ogr != null)
    {
      ogrenciler.Remove(ogr);
      OgrenciSayisi--;
      Console.WriteLine("Öğrenci silindi.");
    }
  }
}
internal class Program
{
  private static void Main(string[] args)
    OgrenciIslem oi = new OgrenciIslem();
    oi.OgrenciEkle(new Ogrenci
    {
      Numara = 100,
      AdSoyad = "Nihal Öz"
    });
    Console.WriteLine("Öğrenci sayısı: " + OgrenciIslem.OgrenciSayisi);
    oi.OgrenciEkle(new Ogrenci
    ł
      Numara = 200,
      AdSoyad = "İbrahim Yurt"
    });
    Console.WriteLine("Öğrenci sayısı: " + OgrenciIslem.OgrenciSayisi);
    Console.WriteLine("3) =========================");
    oi.OgrenciSil(100);
    Console.WriteLine("Öğrenci sayısı: " + OgrenciIslem.OgrenciSayisi);
  }
}
// Ekran çıktısı:
Statik yapıcı çalıştı.
Yapıcı çalıştı.
1) =======
            _____
Oğrenci eklendi.
Oğrenci sayısı: 1
Öğrenci eklendi.
Öğrenci sayısı: 2
Öğrenci silindi.
Öğrenci sayısı: 1
```





Sıra Sizde

Bir kütüphane sınıfını aşağıdaki istekleri karşılayacak şekilde oluşturunuz.

- Eklenen ögelerin adedini tutan gizli bir özelliği olsun.
- Kitap, dergi veya ansiklopedi ekleme metodu olsun.
- Eklenen öge adedini ekrana yazdıran bir metodu olsun.

3.15. İSİMSİZ SINIFLAR (ANONYMOUS CLASSES)

İsimsiz sınıflar yalnızca salt okunur özellikleri içeren ve adı olmayan sınıflardır. Alan, metot gibi diğer ögeleri barındırar Sınıf özelliklerinin veri tipleri, aldığı değere göre otomatik olarak belirlenir.

Bir isimsiz sınıf, **var** anahtar kelimesi ile tanımlanır ve **new** anahtar kelimesi ile oluşturulur.

var ogrenci = new
{
 Numara = 35,
 Ad = "Yasin",
 Ortalama = 80.5
};
Console.WriteLine("Öğrencinin adı: " + ogrenci.Ad);
//ogrenci.Ortalama = 90.2; HATA!!!

Ayrıca bir isimsiz sınıf içinde bir başka isimsiz sınıf oluşturulabilir.

```
var ogrenci = new
{
    Numara = 35,
    Ad = "Yasin",
    Ortalama = 80.5,
    Adres = new
    {
        Il = "Malatya",
        Ilce = "Yeşilyurt"
};
Console.WriteLine("Öğrencinin yaşadığı il: " + ogrenci.Adres.II);
```

```
//ogrenci.Ortalama = 90.2; HATA!!!
```

İstenirse isimsiz sınıf dizisi de oluşturulabilir.

```
var ogrenciler = new[]
{
    new { Numara = 100, Ad = "Yasin", Ortalama = 80},
    new { Numara = 200, Ad = "İsmail", Ortalama = 75},
    new { Numara = 300, Ad = "Ömer", Ortalama = 60}
};
Console.WriteLine("2. öğrencinin adı: " + ogrenciler[1].Ad);
Console.WriteLine("==========================");
foreach (var ogrenci in ogrenciler)
{
    Console.WriteLine("Adı: {0}, Ortalaması: {1}", ogrenci.Ad, ogrenci.Ortalama);
}
```

// Ekran çıktısı:		
2.öğrencinin adı: İsmail		
Adı: Yasin, Ortalaması: 80		
Adı: İsmail, Ortalaması: 75		
Adi: Omer, Ortalamasi: 60		

Sıra Sizde >

ÖĞRENME BİRİMİ

Bir isimsiz sınıfı parametre olarak alan metodu nasıl tanımlayabileceğinizi araştırınız ve edindiğiniz bilgileri sınıf arkadaşlarınızla paylaşınız.

3.16. MÜHÜRLÜ SINIFLAR (SEALED CLASSES)

Bir sınıftan bir başka sınıf türetilmek istenmediğinde bu sınıfı **sealed** anahtar kelimesiyle mühürlü tanımlamak gerekir.

```
public sealed class UstSinif
{
    // ..
}
public class AltSinif : UstSinif // HATA !!!
{
    // ...
}
```

SINIFLAR(CLASS)

Yukarıdaki kod parçası hata verecektir (Görsel 3.14).



Görsel 3.14: Derleyici hata mesajı

3.17. PARÇALI SINIFLAR (PARTIAL CLASSES)

Büyük projelerde oluşturulan sınıfları birden fazla dosyaya yaymak mümkündür. Parçalı sınıflar;

- Büyük sınıfları parçalamak, okunmasını kolaylaştırmak,
- Mantıksal olarak katmanlara ayırmak (veri tabanı işlemlerinin ayrı bir dosyada olması gibi),
- Sınıf ögelerini ayrıştırmak (özellikler bir dosyada, metotlar başka bir dosyada vb.),
- Aynı sınıf üzerinde birden fazla programcının çalışması gibi durumlar için kullanılabilir.

SINIFLAR(CLASS) ÖĞRENME BİRİMİ

Parçalı sınıflar oluşturmak için partial anahtar kelimesi kullanılır. Parçalı sınıfların isimleri aynı olma-

lıdır.

```
// ParcaliSinif1.cs
public partial class ParcaliSinif
   public int Ozellik1 { get; set; }
  // ...
}
// ParcaliSinif2.cs
public partial class ParcaliSinif
   public void Metot1() { }
  //...
}
// Program.cs
internal class Program
Ł
   private static void Main(string[] args)
      ParcaliSinif ps = new ParcaliSinif();
      Console.WriteLine(ps.Ozellik1);
      ps.Metot1();
  }
```

Program derlendiğinde tüm parçalar birleştirilir ve tek bir sınıf tanımlanmış gibi çalıştırılır.

3.18. ENUMS (NUMARALANDIRMALAR)

Bir **enum** (enumerations kelimesinin kısaltması) sadece **int** tipindeki sabitlerden oluşan özel bir sınıftır. Bu değerler sadece okunabilirdir ve değiştirilemez. Enum genellikle programların okunmasını kolaylaştırmak için kullanılır.

Bir numaralandırma oluşturabilmek için enum anahtar kelimesi ve değerleri birbirinden ayırmak için , (virgül) karakteri kullanılır.

```
enum Seviyeler
ł
  Çok_Düşük, // 0
  Düşük,
              // 1
  Orta,
              // 2
  Yüksek,
            // 3
  Çok Yüksek // 4
internal class Program
ł
  private static void Main(string[] args)
  {
     Console.WriteLine(Seviyeler.Düşük);
     Console.WriteLine((int)Seviyeler.Düşük);
  }
}
// Ekran çıktısı:
Düşük
1
```

enum içindeki değerler 0'dan (sıfır) başlayarak birer birer artar. İstenirse farklı tam sayı değerleri de verilebilir.

enum Kategoriler { Bilgisayar = 3, Mobilya = 10, Kırtasiye = 7, Hırdavat, // 8 Otomobil // 9 }

BİRİMİ

Yukarıdaki örnekte bilgisayar, mobilya ve kırtasiye kategorilerine istenilen değerler atanmıştır. Değer ataması yapılmayan hırdavat ve otomobil kategorilerine ise 7'den sonra gelen 8 ve 9 değerleri otomatik olarak atanmıştır.

Program içinde örnek kullanım aşağıda verilmiştir.

```
// enum tanımlama
Kategoriler kat = Kategoriler.Kırtasiye;
// değer adının kullanımı
Console.WriteLine(kat);
// değerin sayısal değerinin kullanımı
Console.WriteLine((int)kat);
// enuma sayısal değer atama
kat = (Kategoriler)8;
// if ile kullanımı
if (kat == Kategoriler.Kırtasiye)
   Console.WriteLine("Hırdavat kategorisi");
// switch ile kullanımı
switch (kat)
{
   case Kategoriler.Bilgisayar: // ..
     break;
   case Kategoriler.Mobilya: // ..
     break;
   case Kategoriler.Kırtasiye: // ..
      break;
   case Kategoriler.Hırdavat: // ..
      break;
   case Kategoriler.Otomobil: // ..
     break;
   default:
      break;
```

Sıra Sizde

Ayları 1'den başlayarak numaralandıran tanımlamayı yazınız.

Sıra Sizde

Numaralandırmayı nerelerde kullanabileceğinizi araştırınız ve edindiğiniz bilgileri sınıf arkadaşlarınızla paylaşınız.





ÖLÇME VE DEĞERLENDİRME - 3

1. "Ses seviyesi", "ekran boyutu" ve "görüntü teknolojisi" <u>alanlarına</u> sahip bir "Televizyon" sınıfı yazınız.

2. "RAM bellek kapasitesi", "CPU", "HD kapasitesi" <u>alanlarına</u> sahip bir "Bilgisayar" sınıfı yazınız.

3. "Televizyon" sınıfını <u>özellikler</u> kullanarak tekrar yazınız.

4. "Bilgisayar" sınıfını <u>özellikler kullanarak tekrar yazınız.</u>

5. "Televizyon" sınıfına "Güç aç / kapat", "Kanal değiştir" ,"Ses seviyesi oku" metotlarını ekleyiniz (Gerekli alanları sınıfa ekleyiniz.).

-KanalNoArtir() => Kanal numarasını bir artırmalı.

-KanalNoArtir(int) => Kanal numarasını parametrede verilen değer kadar artırmalı.

-KanalNoAzalt() => Kanal numarasını bir azaltmalı.

-KanalNoAzalt(int) => Kanal numarasını parametrede verilen değer kadar azaltmalı.

7. "Televizyon" sınıfında kullandığınız alanların, özelliklerin ve metotların erişim türlerini açıklayınız.

8. "Televizyon" sınıfından "İşletim sistemi" özelliğine sahip bir "Akıllı televizyon" sınıfını aşağıdaki hususlara dikkat ederek türetiniz.

- "Televizyon" sınıfındaki "Güç aç / kapat" metodu bu sınıf içinde tekrar yazılmalıdır.
- "Televizyon" sınıfındaki "ses seviyesi" bilgisi sadece bu iki sınıf içinde kullanılabilir olmalıdır.

9. "Televizyon" sınıfından türetilen tüm nesneler için "Marka" bilgisinin aynı olması istenirse bu sınıf üzerinde nasıl bir değişiklik yaparsınız?

10. "Televizyon" sınıfından başka bir sınıf türetilemesin istenirse bu sınıf üzerinde nasıl bir değişiklik yaparsınız?

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

11. Sınıf içindeki bir değişkeni dış dünyaya kapatıp sadece sınıf içinde kullanılabilir kılmak için bu özellikşeklinde tanımlanmalıdır.

12. Sınıf içindeki bir değişkeni dış dünyaya kapatıp sadece sınıf içinde ve bu sınıftan türetilen alt sınıflarda kullanılabilir kılmak için bu özellik.....

13. Sınıf içindeki bir değişkeni her yerden erişilebilir kılmak için bu özellik şeklinde tanımlanmalıdır.





static void Hesapla(int sayi, int kare, ref int kup)

}

ł

kare = sayi * sayi ; kup = kare * sayi;

17. Statik sınıflar hangi durumlarda kullanılır?

18. Statik yapıcı metotlar ne zaman çalıştırılır?

19. Soyut sınıf ile arayüzler arasındaki benzerlikler ve farklılıklar nelerdir?

20. "Güç aç" ve "Güç kapat" metotlarını tanımlayan bir arayüz yazarak "Televizyon" ve "Bilgisayar" sınıflarına bu metotları uygulayınız.





DİZİLER (ARRAYS)

KOLEKSIYONLAR (COLLECTIONS)

NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Dizi kavramını açıklayacak,
- Dizi tanımlaması yapacak,
- Dizilere değer verme ve dizilerden değer alma işlemlerini yapacak,
- Çok boyutlu dizi kavramını açıklayacak,
- Çok boyutlu dizi tanımlaması yapacak,
- Çok boyutlu dizilere değer verme ve dizilerden değer alma işlemlerini yapacak,
- İhtiyaca uygun olarak dizileri kullanacak,
- Koleksiyon kavramını açıklayacak,
- Koleksiyon tanımlaması yapacak,
- Koleksiyonlara değer verme ve koleksiyonlardan değer alma işlemlerini yapacak,
- İhtiyaca uygun koleksiyon kullanımını öğreneceksiniz.

ANAHTAR KELİMELER

Dizi, index, veri tipi, for döngüsü, foreach döngüsü, koleksiyon



10, SINIF





HAZIRLIK ÇALIŞMALARI

1. Projenizde girilen 10 adet sayının ortalamasını değişkenler kullanarak nasıl alırsınız? Sınıf arkadaşlarınızla paylaşınız.

2. Size verilen 100 kişinin isimlerini düzenlemede yapacağınız ilk işlem nedir? Nedeniyle birlikte açıklayınız?

4.1. DİZİLER

Dizi, aynı tipte birden çok değeri bellek üzerinde tutabilecek yapıdır. Programlama yaparken dizileri kullanmak; dizilerin verdiği avantajlardan yararlanarak değerler üzerinde seçme, silme, değiştirme, sıralama vb. işlemlerin kolayca gerçekleştirilmesini sağlar.

4.1.1. Bir Boyutlu Diziler

Bir boyutlu veya tek boyutlu diziler, verileri saklamak için bir satırdan oluşan dizilerdir. Bu diziler, art arda sıralanmış bellek alanları gibi düşünülebilir. Aynı tipten değerler olmak şartıyla belirlenen adet kadar veri, dizi içinde sıralanmış bellek alanlarında saklanır. Burada dikkat edilmesi gereken noktaların başında diziyi oluştururken içinde kaç adet veri olacağı ve dizilerde hangi tip verilerin saklanacağı (int, string, char, double vb.) gelir. Belirtilen tipin sınırı dışında veya belirtilen adet sayısından daha fazla veri saklamaya çalışıldığında derleyici hata verir.

4.1.2. Bir Boyutlu Dizilerin Oluşturulması

Dizi oluştururken temelde üç noktaya dikkat edilir.

1. Dizinin Tipi: Dizide hangi tip verilerin saklanacağı (int, string, char, byte, double vb.)

2. Dizi Adı: Dizide saklanacak verilerle anlamlandırılan değişken isimlendirme kurallarına uygun hangi isimlerin diziye verileceği (Anlamlı isimler vermek, yazılan kodların okunabilirliğini artıracağı için her zaman tavsiye edilen bir yöntemdir. Örneğin, okul numaraları saklanacak bir dizi için diziOkulNo kullanılabilir.)

3. Dizilerin Kapasitesi: Dizi içinde kaç adet veri saklanacağı



Görsel 4.1: Bir boyutlu dizi tanımlaması

Görsel 4.1'de sayilar isminde, integer tipinde 10 adet veri saklama kapasitesine sahip bir dizi tanımlaması yapılmıştır. Derleyicinin bir diziyi tanıması için başlangıçta veri tipi belirtildikten sonra içi boş köşeli parantezler kullanılmalıdır. İçi boş köşeli parantezler, bu ifadenin bir boyutlu dizi olduğu anlamına gelir. Tanımlamadaki ikinci köşeli parantez ise dizide saklanacak değer sayısını belirtmektedir. Aşağıda farklı veri tiplerine sahip dizi tanımlama örnekleri verilmiştir.

<pre>string[] isimler = new string[5]; // String tipinde 5 elemanlı dizidir.</pre>
<pre>byte[] siralar = new byte[6]; // Byte tipinde 6 elemanlı dizidir.</pre>
bool[] durumlar = new bool[4]; // Boolean tipinde 4 elemanlı dizidir.
<pre>float[] uzunluklar = new float[8]; // Float tipinde 8 elemanlı dizidir.</pre>

Bir dizi tanımlaması yapıldığında derleyici, dizinin her elemanına temel veri tipleri için varsayılan değerleri ilk değer olarak verir. İlk değerler, dizi içine veri eklenmeden verilir. Bunlar; string tipi için null, sayısal tipler için 0, bool tipi için ise false değerleridir. Verilen bu ilk değerler, dizilere değer aktarımı yapıldıkça yeni değerlerle değiştirilir.



10, SINIF





4.1.3. Bir Boyutlu Dizilere Değer Aktarma

Dizilere değer aktarmanın farklı yöntemleri vardır. Dizilere ilk olarak tanımlamasının yapıldığı satırda değer verilebilir.

string[] kisiler = new string[5] {"Ayse","Fatma","Yusuf","Merve","Ahmet"};

Yukarıda tanımlanan ve aynı satırda değer aktarımı yapılan dizide kodlar derlendiğinde bellekte **5** elemanlı bir dizi oluşturulur. Oluşturulan bu diziye küme parantezi **{ }** içindeki değerler sırasıyla verilir.



Görsel 4.2: Diziye değer aktarma

Günlük hayatta sıralama işlemlerine hep 1'den başlanır fakat programlama dillerinin çoğunda sıralama 0'dan başlar. Görsel 4.2'de eklenen ilk elemanın sıra numarası 0'dır. Dizilerin her değerinin bir sıra numarası vardır. Sıra numaraları index, indis veya indeks olarak adlandırılır.

Tanımlandıkları satırda dizilere değer aktarma işlemi farklı şekillerde de yapılabilir. Aşağıdaki örnekte diziye değer aktarım işlemi, dizinin eleman sayısı belirtilmeden veya new sözcüğü kullanılmadan gerçekleştirilmiştir. Bu durumda derleyici hata vermez ve dizinin eleman sayısı derleyici tarafından belirlenir.

string[] kisiler = new string[] {""Ayse","Fatma","Yusuf","Merve","Mehmet"};

string[] kisiler = {"Ayşe","Fatma","Yusuf","Merve","Mehmet"};

Dizilere değer aktarımının bir diğer yöntemi, dizinin index numaralarının kullanılarak yapılmasıdır.

int[] sayilar = new int[6];

Yukarıdaki kodda **integer** veri tipine sahip, **6** elemanlı, **sayilar** adında bir dizi tanımlandı. Derleyici, bu kod satırına geldiğinde bellekte eleman sayısı kadar yer ayırır ve bu yerlere ilk değer olarak **0** (sıfır) sayısını aktarır.



Görsel 4.3: Dizinin ilk değerlerinin verilmesi

Bellek üzerinde dizi oluşturulduktan sonra index numaraları kullanılarak değer aktarımı gerçekleştirilebilir. Görsel 4.4'te **sayilar** ismindeki dizinin **2** numaralı index elemanına (dizinin üçüncü elemanına) 45 değerinin aktarımı yapılmıştır.



Görsel 4.4: Diziye index numarası ile değer aktarılması







Sıra Sizde

BİRİMİ

Aşağıdaki kodlamaları yaparak dizi elemanlarına değer aktarma işlemini gerçekleştiriniz.

```
int[] sayilar = new int[6];
sayilar[0] = 10; //sayilar dizisinin 0 index numaralı elemanı 10 oldu.
sayilar[1] = 25; //sayilar dizisinin 1 index numaralı elemanı 25 oldu.
sayilar[2] = 45; //sayilar dizisinin 2 index numaralı elemanı 45 oldu.
sayilar[3] = 5; //sayilar dizisinin 3 index numaralı elemanı 5 oldu.
sayilar[4] = -30; //sayilar dizisinin 4 index numaralı elemanı -30 oldu.
sayilar[5] = -50; //sayilar dizisinin 5 index numaralı elemanı -50 oldu.
```

Görsel 4.5'te değer aktarma işlemi bittikten sonra derleme işleminde bellek üzerindeki dizinin son hâli verilmiştir.



Görsel 4.5: Değer atandıktan sonra dizinin bellek üzerindeki durumu

Değer aktarım işleminde **index** numaralarına göre dizinin index numarası **0**'dan başlayarak dizideki eleman sayısının bir eksiğine kadar istenilen alana değer aktarılabilir. Tanımlanan dizide **0**'dan küçük ve eleman sayısının bir eksiğinden büyük bir index numarası ile diziye değer aktarmaya veya dizi elemanına erişmeye çalışıldığında derleyici tarafından hata mesajı gönderilir. Hata mesajı, girilen index numarasının dizinin sınırları dışında olduğunu bildirmektedir.

sayilar[6] = -30;

Yukarıdaki kod yazıldığında derleyici Görsel 4.6'daki hata mesajını vererek kullanıcıyı uyarır.



Görsel 4.6: Dizi sınır aşımı hata mesajı

Bir Boyutlu Diziler

Uygulama-1

Dizilere değer aktarım işlemi, kodlama sırasında değil de uygulamanın çalışması esnasında olabilir. Aşağıdaki uygulamada çalışma esnasında dizilere değer aktarım işlemi yapılmıştır.



10, SINIF



Adım 1: Görsel 4.7'deki form tasarımını yaptıktan sonra form içindeki kontrollere name değerlerini (mavi yazı ile belirtilen) veriniz.

🖳 Bir Boyutlu Diziler Uygulama 1	_		\times
Ad Soyad		Ekle	
txtAdSoyad	b	tnEkle	

Görsel 4.7: Dizi uygulamaları form tasarımı-1

Adım 2: Ekle butonu Click olayında butona her tıkladığınızda TextBox içine girilen değerleri **isimler** adındaki 5 elemanlı bir diziye aktaracak kodlamaları yapınız.

```
string[] isimler = new string[5]; //Global Dizi
int index = 0; //Global Değişken
private void btnEkle_Click(object sender, EventArgs e)
{
    isimler[index] = txtAdSoyad.Text;
    index++;
}
```

Sıra Sizde >

1. Kodlarda isimler dizisi neden btnEkle_Click içinde değil de global olarak tanımlanmıştır? Arkadaşlarınızla paylaşınız.

2. Adım 2'de verilen kodda tanımlanan index ismindeki değişken neden kullanılmıştır? Açıklayınız.

3. Uygulamada 6. kişi eklenmeye çalışıldığında nasıl bir hatayla karşılaşılır? Karşılaşılan bu hatanın cözümü için neler yapılabileceğini sınıf arkadaşlarınızla paylaşınız.

4.1.4. Bir Boyutlu Dizi Elemanlarına Erişim

Dizinin elemanlarına erişim, dizilere değer aktarmada olduğu gibi index numaraları kullanılarak sağlanır. Erişim sağlanan dizi elemanı; değişkenlere aktarma, hesaplamalar yapma, ekrana yazdırma, nesneye aktarma gibi işlemlerde kullanılır.

```
int[] dizi = new int[5] { 46, 41, 34, -10, 55 };
// 5 elemanlı bir dizi tanımlaması yapıldı. Aynı satırda değerler verildi.
int sayi1 = dizi[0];
// Dizinin 0 index numaralı değeri sayi1 ismindeki int tipindeki değişkene aktarıldı.
int toplam = dizi[0] + dizi[1] + dizi[2];
// Dizinin ilk üç elemanıyla toplama işlemi yapıldı.
Console.WriteLine(dizi[1]);
// Dizinin 1 index numaralı değeri (41 değeri) Console ekranına yazdırıldı.
dizi[2] = dizi[3];
// Dizinin 3 index numaralı değeri 2 index numaralı alana aktarıldı.
label1.Text = dizi[4].ToString();
// Dizinin 4 index numaralı değeri label1 nesnesinin text özelliğine aktarıldı.
```











Bir Boyutlu Diziler

Bu işlem, Uygulama 1'deki 5 elemanlı isimler dizisinde yer alan değerleri, Listele butonuna tıklayarak ListBox nesnesinin içinde göstermek olacaktır.

Adım 1: Görsel 4.8'de verilen formun tasarımını yapınız. Form kontrol nesnelerine name değerlerini veriniz.

Ad Soyad Ekle txtAdSoyad btnEkle Listele btnListe bl iste

Görsel 4.8: Dizi uygulamaları form tasarımı-2

Adım 2: Listele butonu Click olayına dizi içindeki her bir elemanı for döngüsü yardımıyla ListBox içinde göstermek için Items.Add() metodu kullanarak listeleme işlemi yapan kodlamaları yazınız.

```
private void btnListele_Click(object sender, EventArgs e)
{
  for (int i = 0; i < isimler.Length; i++)
    {
        IbListe.Items.Add(isimler[i]);
    }
}</pre>
```

Sıra Sizde

- 1. Kodlarda dizi isminden sonra Length ifadesi hangi amaç için kullanılmıştır?
- **2.** Uygulamada 5 elemanlı dizinin tamamına değer aktarmadan listele butonuna tıklandığında nasıl bir hata ile karşılaşılır? Karşılaşılan bu hatanın çözümü için ne yapılabilir?
 - 3. Dizi değerlerini listeleme işlemi, for döngüsü yerine while döngüsü kullanılarak nasıl yapılır?

4.1.5. Dizilerde Foreach Döngüsü Kullanımı

Birçok programlama dili, diziler üzerinde işlem yapılmasını kolaylaştıran bir döngü sunar. Bu döngü, foreach döngüsüdür. Dizilerde kullanılan foreach, dizi elemanlarını ilk elemandan başlayıp dizinin son elemanına kadar her elemanı tek tek dolaşarak belirlenen bir değişkene aktarır. Örneğin, 10 elemanlı bir dizide foreach döngüsü kullanıldığında döngü 10 defa tekrarlama işlemi yapar. Döngü her seferinde dizi içindeki değeri alarak aynı tipte olan bir değişkene aktarır. Döngünün yapısı aşağıda verilmiştir.

```
foreach (Tip Değişken in Dizi)
{
// Döngü içindeki işlemler
```



DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS)



Foreach döngüsünün yapısındaki ögeler aşağıda sıralanmıştır:

• **Tip:** Dizi içindeki veri tipleri ile aynı olmalıdır (Dizi içindeki değerler string ise Tip de string, double ise Tip de double olmalıdır.). Bazı durumlarda Tip olarak **var** kullanılır. Var tipi, kendisine atanan değer ne ise o değerin tipini alır.

- **Değişken:** Döngü, dizi içindeki değeri her dönme işleminde belirtilen bir değişkene aktarır.
- in: Bir anahtar kelimedir, foreach döngülerinde değişken adlarından sonra kullanılır.
- Dizi: Üzerinde işlem yapılacak dizinin adıdır.

<pre>int[] sayilar = { 20, 30, 40, 50 }; foreach (int sayi in sayilar)</pre>	<pre>int[] sayilar = { 20, 30, 40, 50 }; for (int i = 0; i < sayilar.Length; i++)</pre>
{	{
Console.WriteLine(sayi);	Console.WriteLine(sayilar[i]);
}	}

Yukarıdaki örneklerde iki döngü de aynı görevi yerine getirmektedir. Dizilerde for döngüsüne göre daha az kod yazarak sonuca ulaşıldığı için genellikle foreach döngüsü tercih edilmektedir. Foreach döngüsünde dizi içindeki değerler döngü tamamlanıncaya kadar sırasıyla sayi ismindeki değişkene aktarılır. For döngüsünde i değişkeni, dizinin index numarası olarak dizi elemanlarına erişim için kullanılır.

Sıra Sizde

Bir Boyutlu Diziler Uygulama 2'deki dizi elemanlarını listele butonuna tıklama olayında ListBox içine foreach döngüsü kullanarak listeleyiniz.

Uygulama-3

Bir Boyutlu Diziler

Bu işlemde Bir Boyutlu Diziler Uygulama 2'deki 5 elemanlı isimler dizisiyle birlikte integer tipindeki değerleri saklayan 5 elemanlı **notlar** isimli dizi kullanılacaktır. **Listele** butonuna tıklandığında her iki dizideki değerlerin ListBox nesnesinin içinde gösterilmesi sağlanacaktır.

Adım 1: Görsel 4.9'da verilen formun tasarımını yapınız. Form kontrol nesnelerine name değerlerini veriniz.

😼 Bir Boyutlu Diziler Uygulama 3		—		×
Ad Soyad txtAdSoyad Ders Notu txtDersNotu	Ekle btnEkle	Liste	Listel	e

Görsel 4.9: Dizi uygulamaları form tasarımı-3



<pre>string[] isimler = new strin</pre>	g[5]; //Global dizi
<pre>int[] notlar = new int[5];</pre>	//Global dizi
int index = $0;$	//Global değişken





Adım 3: Ekle butonunun Click olayına aşağıdaki kodları yazarak dizilere ad, soyad ve ders notu bilgilerini aktarınız.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    if(index<isimler.Length)
    {
        isimler[index] = txtAdSoyad.Text;
        notlar[index] = int.Parse(txtDersNotu.Text);
        index++;
        txtAdSoyad.Text = "";
        txtDersNotu.Text = "";
    }
}</pre>
```

Adım 4: Listele butonu Click olayında aşağıdaki kodları yazarak dizideki değerleri ListBox kontrolü içinde gösteriniz.

```
private void btnListele_Click(object sender, EventArgs e)
{
  for (int i = 0; i < isimler.Length; i++)
  {
    if(isimler[i] != null)
    {
        lbListe.Items.Add(isimler[i]+" > "+notlar[i]);
    }
}
```

Sıra Sizde >

Uygulama-4

Bir Boyutlu Diziler

- 1. Adım 3'te karşılaştırma ifadesinin kullanım amacı nedir?
- 2. Adım 4'te karşılaştırma ifadesinin kullanım amacı nedir?
- 3. Adım 3'te for döngüsü yerine foreach döngüsü kullanılabilir mi? Neden?



BİRİMİ

----- Ř

ttp://kitap.eba.gov.tr/ lodSor.php?KOD=21077

Bu işlem, Bir Boyutlu Diziler Uygulama 3'ün geliştirilmiş hâlidir. Bir önceki uygulamada isimler ve notlar dizilerine değer aktarımı sağlanmıştı. Bu uygulamada ise girilen notlar içinde en yüksek ve en düşük not ile tüm notların ortalamasını hesaplama işlemi yapılacaktır.







Adım 1: Görsel 4.10'da verilen formun tasarımını yapınız. Form kontrol nesnelerine name değerlerini veriniz.

🖳 Bir Boyutlu Diziler Uygulama 4		_		×
Ad Soyad	txtAdSoyad			
Ders Notu	txtDersNotu			
btnEkle Ekle		Lis	stele	
btnEnYuksek En Yüksek	txtEnYuksek	btnl	istele	
btnEnDusuk En Düşük	txtEnDusuk			
btnOrtalama Ortalama	txtOrtalama lbListe			

Görsel 4.10: Dizi uygulamaları form tasarımı-4

Adım 2: En Yüksek butonu Click olayında notlar dizisi içindeki en yüksek notu bulduran kodlamayı yapınız.

```
private void btnEnYuksek_Click(object sender, EventArgs e)
{
    int enyuksek = notlar[0];
    for (int i = 0; i < notlar.Length; i++)
    {
        if (notlar[i] > enyuksek)
        {
        enyuksek = notlar[i];
        }
    }
    txtEnYuksek.Text = enyuksek.ToString();
}
```

Adım 3: En Düşük butonu Click olayında notlar dizisi içindeki en düşük notu bulduran kodlamayı yapınız.

```
private void btnEnDusuk_Click(object sender, EventArgs e)
{
    int endusuk = notlar[0];
    for (int i = 0; i < notlar.Length; i++)
    {
        if (notlar[i] < endusuk)
        endusuk = notlar[i];
    }
    txtEnDusuk.Text = endusuk.ToString();
}</pre>
```





Adım 4: Ortalama butonu Click olayında notlar dizisi içindeki notların ortalamasını bulduran kodlamayı yapınız.

```
private void btnOrtalama_Click(object sender, EventArgs e)
{
    int toplam=0;
    double ortalama=0;
    for (int i = 0; i < notlar.Length; i++)
    {
        toplam += notlar[i];
     }
        ortalama = toplam / notlar.Length;
    txtOrtalama.Text = ortalama.ToString();
}</pre>
```

Sıra Sizde

- 1. Adım 2'nin çalışma mantığının nasıl olduğunu açıklayınız.
- 2. Adım 2'de kullandığınız kodda neden dizinin ilk elemanı bir değişkene aktarıldı?
- 3. Adım 2 ve 3'teki karşılaştırma ifadelerinin kullanım amacı nedir?
- 4. Adım 4'te toplam ve ortalama değişkenlerine neden başlangıç değeri olarak 0 verildi?
- 5. Adım 4'te ortalama değişkeninin tipi neden double olarak seçildi?

Uygulama-5

Bir Boyutlu Diziler

Bu uygulamada bir dizinin elemanları başka bir diziye kopyalanacaktır.

Adım 1: Görsel 4.11'de verilen formun tasarımını yapınız. Form kontrol nesnelerine name değerlerini

veriniz.



Görsel 4.11: Dizi uygulamaları form tasarımı-5

Adım 2: Formda kaynak dizi ve kopyalanacak diziyi 100 elemanlı ve integer tipinde sayıları saklayacak şekilde global olarak oluşturunuz.

```
int[] diziKaynak = new int[100];
int[] diziKopya = new int[100];
```



DİZİLER (ARRAYS) VE KOLEKSİYONLAR (COLLECTIONS)

Adım 3: Dizi Oluştur butonu Click olayında kaynak diziye 0 ile 100 arasında rastgele sayılardan oluşan değerlerin aktarımı yapılacaktır. Bu işlem için **Random** sınıfından üretilen **rastgele** isimli nesneye Next metodu ile birlikte 0 ile 100 arasında üretilen sayıları diziye aktaran ve dizi değerlerini listeleyen kodlamayı yapınız.

```
private void btnOlustur_Click(object sender, EventArgs e)
{
   Random rastgele = new Random();
   for (int i = 0; i < diziKaynak.Length; i++)
   {
      diziKaynak[i] = rastgele.Next(0, 101);
   }
   for (int i = 0; i < diziKaynak.Length; i++)
   {
      listeAsil.Items.Add(diziKaynak[i]);
   }
}</pre>
```

Adım 4: Dizi Kopyala butonu Click olayında ise kaynak dizideki değerleri kopyalanacak diziye aktarım işlemini yaptıktan sonra kopyalanan diziyi listeleyen kodlamaları yapınız.

```
private void btnKopya_Click(object sender, EventArgs e)
{
    for (int i = 0; i < diziKopya.Length; i++)
    {
        diziKopya[i] = diziKaynak[i];
    }
    for (int i = 0; i < diziKopya.Length; i++)
    {
        listeKopya.Items.Add(diziKopya[i]);
    }
}</pre>
```

Sıra Sizde >

ÖĞRENME

1. Listeleme işlemlerini foreach döngüsü ile gerçekleştiriniz.

2. Dizi kopyalama işleminin daha kısa yolu var mıdır? Oluşturacağınız küçük gruplarla dizi kopyalama işleminin kısa yolunun olup olmadığını tartışınız. Sonuçları sınıfla paylaşınız.

4.1.6. Bir Boyutlu Dizilerde Kullanılan Özellikler ve Metotlar

Dizilerle işlem yaparken işleri kolaylaştıracak bazı metotlar mevcuttur. Bu metotlar, dizilerin daha etkin kullanılabilmesine olanak sağlar.

- Length Özelliği: Dizideki eleman sayısını verir. Bu özelliğin kullanımı diziadi.Length şeklindedir.
- Rank Özelliği: Dizideki boyut sayısını verir. Bu özelliğin kullanımı diziadi.Rank şeklindedir.
- Max Metodu: Dizideki en büyük sayıyı verir. Bu metodun kullanımı diziadi.Max() şeklindedir.
- Min Metodu: Dizideki en küçük sayıyı verir. Bu metodun kullanımı diziadi.Min() şeklindedir.
- Sum Metodu: Dizideki sayıların toplamını verir. Bu metodun kullanımı diziadi.Sum() şeklindedir.

Average Metodu: Dizideki sayıların ortalamasını verir. Bu metodun kullanımı diziadi.Average() şeklindedir.

- First Metodu: Dizideki ilk elemanı verir. Bu metodun kullanımı diziadi.First() şeklindedir.
- Last Metodu: Dizideki son elemanı verir. Bu metodun kullanımı diziadi.Last() şeklindedir.



10, SINIF





Sıra Sizde

Bir form tasarımı yapınız. Formun içine bir adet Button ve bir adet ListBox ekleyiniz. Butona tıklandığında oluşturduğunuz sayısal tipteki diziye 0 ile 10 arasında rastgele değerler aktarınız. Değer aktarımı yapılan dizi için hangi metot ve özelliklerin kullanılabileceğini sınıfla paylaşınız. Elde edilen sonuçları ListBox içinde gösteren kodlamayı yapınız.

4.1.7. Çok Boyutlu Diziler

Tek boyutlu diziler, aynı veri tipinden değerler içeren ve tek satırdan oluşan dizilerdir. Çok boyutlu diziler ise tek boyutlu dizilerin genişletilmiş biçimidir. Çok boyutlu diziler genellikle matematiksel hesaplamalar, görüntü işleme ve kayıt işlemlerinde kullanılır. Çok boyutlu diziler içinde en çok iki boyutlu ve üç boyutlu diziler kullanılmaktadır. Bu öğrenme biriminde iki boyutlu diziler açıklanacaktır.

İki boyutlu diziler, tek boyutlu dizilerin birden çok satırdan oluşmuş hâlidir. Satranç tahtası, iki boyutlu dizilere örnek olarak verilebilir. Satranç tahtasında 8 satır ve 8 sütundan oluşan kareler ile her karenin satır ve sütunlarının kesişmesinden oluşan bir adresi vardır. Bu adresler kullanılarak karenin tahta üzerindeki konumu belirlenir. Tahtanın her karesinde bir veri saklanacağı varsayılırsa adresler kullanılarak karelerin içindeki verilerle işlem yapılabilir.

4.1.8. İki Boyutlu Dizi Tanımlama

Çok boyutlu dizilerin tanımlanmasında boyut sayısı, veri tipi belirtildikten sonra köşeli parantez içine yazılan virgül adediyle belirlenir. Örneğin; hiç virgül yoksa tek boyutlu dizi, bir virgül varsa iki boyutlu dizi, iki virgül varsa üç boyutlu dizi tanımlanacaktır.

int[,] dizi2d;	// İki boyutlu dizi	
int[, ,] dizi3d;	// Üç boyutlu dizi	
int[, , ,] dizi4d ;	// Dört boyutlu dizi	
int[, , , ,] dizi5d;	// Beş boyutlu dizi	

Görsel 4.12'de double tipinde sayılar saklayan **notlar** isminde **3** satır ve **4** sütunlu iki boyutlu bir dizi tanımlaması yapılmıştır. Derleyici, kodu derlediğinde bellek üzerinde 3*4=12 adet sayının saklanacağı bir dizi oluşturur.



Görsel 4.12: İki boyutlu dizi tanımlaması

Derleyici, tanımlanan diziyi derlediğinde bellek üzerindeki bu alanlara varsayılan olarak 0 değerini verir. Görsel 4.13'te sağ alt köşede kırmızı ile belirtilen adresler kullanılarak dizideki verilerle işlem yapılır. Tek boyutlu dizilerde olduğu gibi iki boyutlu dizilerde de hem satır hem sütun sıralaması daima 0'dan başlar.



Görsel 4.13: İki boyutlu dizinin satır ve sütunları











İki boyutlu dizilere iki şekilde değer aktarılır.

• İki boyutlu diziye tanımlandığı satırda değer aktarımı yapılabilir.

double[,] notlar = new double[3,4] {{45,55,60,65},{75,80,85,90},{10,20,30,40}};

• Dizinin index numaraları kullanılarak değer aktarımı yapılabilir.

double[,] notlar = new double[3, 4]; notlar[0, 0] = 45; notlar[0, 1] = 55; notlar[0, 2] = 60; notlar[0, 3] = 65; notlar[1, 0] = 75; notlar[1, 1] = 80; notlar[1, 2] = 85; notlar[1, 3] = 90; notlar[2, 0] = 10; notlar[2, 1] = 20; notlar[2, 2] = 30; notlar[2, 3] = 40;

ÖĞRENME BİRİMİ

Diziye değer aktarımı yapıldıktan sonra dizinin değerleri Görsel 4.14'teki gibi olur.

	Sütun	Sütun	Sütun	Sütun
	0	4	2	ω
Satır Ø	45 _{0,0}	55 _{0,1}	60 0,2	65 0,3
Satır 1	75	80 1,1	85	90
Satır 2	10 2,0	20 2,1	30 2,2	40 2,3

Görsel 4.14: İki boyutlu diziye değer aktarma





Sıra Sizde

Sınıfınızda üçer kişilik dört takım kuracağınızı düşününüz. İki boyutlu dizi oluşturarak her satırda bir takım bulunacak şekilde arkadaşlarınızın isimlerini bu diziye aktarma işlemini yapınız.

Uygulama-1			

İki Boyutlu Diziler

Bu uygulamada üç öğrenciye ait 4 adet ders notu girişi yapılacaktır. Öğrenci adları tek boyutlu diziye, öğrenciye ait ders notları iki boyutlu diziye aktarılacaktır.

Adım 1: Görsel 4.15'te verilen form tasarımını yapınız. Form üzerindeki kontrollerin name değerlerini veriniz.

🖳 İki Boyutlu	ı Diziler Uyg	ulama 1		—		×
Ad Soyad			txtAdSoyad			
Yazılı 1 txtNot1	Yazılı 2 txtNot2	Sözlü 1 txtNot3	Sözlü 2 txtNot4		Ekle btnEkle	

Görsel 4.15: İki boyutlu dizi form tasarımı uygulaması-1

Adım 2: Öğrenci adı soyadı değerlerinde tek boyutlu dizi, not değerlerinde iki boyutlu dizi kullanılacağı için dizi tanımlamalarını global olarak yapınız.

string[] isimler = new string[3]; //Tek boyutlu global dizi				
<pre>int[,] notlar = new int[3,4];</pre>	//İki boyutlu global dizi			
int index = 0;	//Global değişken			

Adım 3: Ekle butonu Click olayında TextBox'lara girilen değerleri ilgili dizilere aktarma işlemini gerçekleştiriniz.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    isimler[index] = txtAdSoyad.Text;
    notlar[index, 0] = int.Parse(txtNot1.Text);
    notlar[index, 1] = int.Parse(txtNot2.Text);
    notlar[index, 2] = int.Parse(txtNot3.Text);
    notlar[index, 3] = int.Parse(txtNot4.Text);
    index++;
}
```

Sıra Sizde >

• Görsel 4.15'teki uygulamanın çalışma mantığını açıklayınız.

• Görsel 4.15'teki uygulamada notları girilecek öğrenci sayılarını form içindeki bir TextBox'tan alacak şekilde kodları düzenleyiniz.







4.1.10. İki Boyutlu Dizi Elemanlarına Erişim

Bir boyutlu dizilerde olduğu gibi iki boyutlu dizi elemanlarına erişmek için de index numaraları kullanılır. Erişim sağlanan dizi elemanı; değişkenlere aktarma, hesaplamalar yapma, ekrana yazdırma, nesneye aktarma gibi işlemlerde kullanılır.

double[,] notlar = new double[3,4] {{45,55,60,65},{75,80,85,90},{10,20,30,40}}; // 3x4 elemanlı iki boyutlu dizi tanımlaması yapıldı. Aynı satırda değerler verildi.
int sayi1 = notlar[0,1]; // Dizinin 0,1 index numaralı değeri sayi1 ismindeki integer değişkenine aktarıldı.
int toplam = notlar[1,0] + notlar[1,1] + notlar[1,2] + notlar[1,3]; // Dizisinin 1. index numaralı satırındaki elemanlarla toplama yapıldı.
Console.WriteLine(notlar[2,1]); // Dizisinin 2,1 index numaralı değeri Console ekranına yazdırıldı.
notlar[2,0] = notlar[1,3]; // Dizisinin 1,3 index numaralı değeri 2,0 index numaralı alanına aktarıldı.
label1.Text = notlar[2,2].ToString(); // Dizinin 2,2 index numaralı değeri label1 nesnesinin text özelliğine aktarıldı.
1

Uygulama-2

İki Boyutlu Diziler

İki Boyutlu Diziler Uygulama 1'de diziye aktarılan not bilgilerinin ortalaması hesaplanarak ListBox içinde gösterilecektir.

Adım 1: Uygulama 1 formuna Görsel 4.16'da olduğu gibi ListBox ve Button kontrollerini ekleyiniz.

🔜 İki Boyutlu Diziler Uygulama 2			_	
Ad Soyad	txtAdSoyad			
Yazılı 1 Yazılı 2 Sözlü 1 Image: Solution of the second state of the se	Sözlü 2 txtNot4	Ekle btnEkle		Listele btnListele
			listeNotlar	

Görsel 4.16: İki boyutlu dizi form tasarımı uygulaması-2





Adım 2: Listele butonu Click olayında isimlerle birlikte girilen notların ortalamasını hesaplayarak ListBox içinde gösteren kodlamaları yapınız.



Sıra Sizde >

Adım 2'de yazılan kodların çalışma mantığını açıklayınız.

Uygulama-3

İki Boyutlu Diziler

Bu uygulamada iki boyutlu diziler kullanılarak basit bir oyun programı yapılacaktır. Oyun programının amacı, 4 satır ve 4 sütundan oluşan iki boyutlu dizide rastgele bir konuma 1 değerini aktarmaktır. Dizinin geri kalan değerleri 0 olarak bırakılacaktır. Satır ve sütun bilgileri, form üzerinde bulunan iki adet TextBox'tan alınacaktır. Alınan satır ve sütun bilgileri dizinin satır ve sütun konumundaki değerle karşılaştırılarak dizi içindeki değer 1 ise oyun kaybedilecek ve ilgili PictureBox'ın rengi kırmızı olacak, dizi içindeki değer 1 değil ise ilgili PictureBox'ın rengi yeşil olacak şekilde kodlama yapınız.

Adım 1: Yeni bir form oluşturarak 16 adet PictureBox, 2 adet TextBox ve 2 adet Button kontrolünü Görsel 4.17'deki gibi ekleyiniz. Kontrollere name değerlerini veriniz.



Görsel 4.17: İki boyutlu dizi form tasarımı uygulaması-3



http://kitap.eba.gov.tr/ KodSor.php?KOD=21078





Adım 2: Byte tipinde 4 satır ve 4 sütundan oluşan iki boyutlu diziyi global olarak tanımlayınız.

byte[,] dizi = new byte[4, 4];

Adım 3: Yeni Oyun butonu Click olayında dizi içinde rastgele bir konuma 1 değerini aktarmak için Random sınıfından üretilen rastgele isminde bir nesne kullanınız. Random sınıfından üretilen bu nesne Next metodu ile birlikte 0 ile 4 arasında, 4 dâhil olmayacak şekilde bir sayı vermektedir. Üretilen bu sayılar, dizinin herhangi bir konumuna 1 değerini aktarmak için kullanılmaktadır. Dizinin belirtilen konumundaki sayı 1 olurken diğer konumlardaki değerler 0 olarak kalmaktadır.

private void btnYeni_Click(object sender, EventArgs e)
{
 Random rastgele = new Random(); // rastgele isminde bir Random nesnesi oluşturdu.
 int satirRastgele = rastgele.Next(4);// 0-4 arası (4 dâhil değil) üretilen sayıdır.
 int sutunRastgele = rastgele.Next(4); // 0-4 arası (4 dâhil değil) üretilen sayıdır.
 dizi[satirRastgele, sutunRastgele] = 1; // Dizi içinde rastgele bir konuma 1 değeri aktarıldı.
}

Adım 4: Gönder butonu Click olayında satır ve sütun TextBox'larına girilen bilgileri satir ve sutun isimli değişkenlere aktarınız. Bu değişkenleri kullanarak önce hangi PictureBox'ın seçileceğini belirleyiniz. Örneğin; satır değeri 1, sütun değeri 3 ise formdaki PictureBox kontrollerinden name değeri "p13" seçilecektir. Dizi içinde [1,3] konumundaki değer 1 ise kutu rengi kırmızı, değilse kutu rengi yeşil olmaktadır.

```
private void btnGonder_Click(object sender, EventArgs e)
{
    byte satir = byte.Parse(txtSatir.Text);
    byte sutun = byte.Parse(txtSutun.Text);
    PictureBox kutu=this.Controls.Find("p"+satir+sutun, true)[0]as PictureBox;
    byte durum = dizi[satir, sutun];
    if (durum == 0)
    {
        kutu.BackColor = Color.Green;
    }
    else
    {
        kutu.BackColor = Color.Red;
    }
}
```





10, SINIF





İki Boyutlu Diziler

Bu uygulamada satır ve sütun sayıları kullanıcı tarafından verilen iki boyutlu bir diziye form üzerinden girilen belirli aralıktaki rastgele sayıları aktarma işlemi yapılacaktır. Sonraki işlem, bu dizinin değerlerini ListBox nesnesi içinde göstermektir.

Adım 1: Görsel 4.18'deki form tasarımını yapınız ve form üzerindeki kontrollerin name değerlerini veriniz.

🔛 İki Boyutlu Diziler Uygulama 4	– 🗆 🗙
Dizi Satır Sayısı bxtSatirSayisi Dizi Sütun Sayısı bxtSutunSayisi Rastgele En Düşük Sayı bxtRastgeleMin Rastgele En Yüksek Sayı bxtRastgeleMax Dizi Oluştur btnDiziOlustur	Dizi Göster btnListele
	listeDizi

Görsel 4.18: İki boyutlu dizi form tasarımı uygulaması-4

Adım 2: Dizinin satır ve sütun sayıları birden çok yerde kullanılacağı için diziyi global olarak tanımlayınız.

int[,] dizi;
int satirSayisi;
int sutunSayisi;

Adım 3: Dizi Oluştur butonu Click olayında formdan gelen satır ve sütun bilgileri ile bir dizi oluşturunuz. Oluşturulan bu diziye aktarılacak rastgele sayıların hangi aralıkta olacağı bilgisi kullanılarak iki boyutlu bir dizi elde edilecektir. Bu dizinin her elemanına içi içe döngü kullanılarak rastgele sayılar atanacaktır.



Adım 4: Dizi Göster butonu Click olayında rastgele sayılarla oluşturulan dizinin ListBox nesnesi içinde ait olduğu satır ve sütun bilgisi ile satır ve sütuna ait değerleri listeleyiniz.

```
private void btnlistele_Click(object sender, EventArgs e)
{
  for (int x = 0; x < satirSayisi; x++)
     {
     for (int y = 0; y < sutunSayisi; y++)
      {
        listeDizi.Items.Add(x+","+y+" => "+dizi[x,y]);
     }
}
```

Sıra Sizde

- Aynı uygulamayı üç boyutlu dizi kullanarak yeniden düzenleyiniz.
- Üç boyutlu dizi oluşturmak ve listelemek için kaç adet iç içe döngü kullanmak gereklidir? Neden?

4.2. KOLEKSİYONLAR

Diziler, programlama dillerinde çokça kullanılan yapılardır. Dizilerde çok sayıda değer tutulabilir ve onlara erişim sağlanabilir ancak dizilerle işlem yaparken iki sınırlama ile karşılaşılır. Bu sınırlamalar şunlardır:

- 1. Dizilere aktarılacak değerler dizi ile aynı tipte olmalıdır.
- 2. Dizilerin eleman sayıları önceden belirlenmelidir.

Programlamada bazen aynı veri tipine sahip olmayan değerlerle işlem yapmak zorunda kalınabilir. Nesne Tabanlı Programlama bu durumda koleksiyonları (collections) sunar. Koleksiyonları kullanabilmek için projeye **System.Collections** isim uzayı (namespace) dâhil edilmelidir. Koleksiyonların isim uzayına eklenmesi aşağıda verilmiştir.

using System.Collections;

Dizi ve koleksiyon arasındaki temel farklar Tablo 4.1'de belirtilmiştir.

Tablo 4.1: Dizi ve Koleksiyon Karşılaştırması

Diziler	Koleksiyonlar
Diziler aynı tip verileri saklar.	Koleksiyonlar aynı veya farklı tipteki verileri saklar.
Dizilerin eleman sayısı başlangıçta belirlenir ve sonradan değiştirilemez.	Koleksiyonların eleman sayılarını başlangıçta belirt- meye gerek yoktur.
Dizilerin eleman sayıları sabittir. Bu nedenle ihti- yaç durumunda eleman sayılarının artırılmasına ve azaltılmasına izin vermez.	Koleksiyonların eleman sayıları değiştirilebilir. İhtiyaca göre eleman sayıları artırılabilir ve azaltıla- bilir.
Diziler performans açısından koleksiyonlardan daha hızlıdır.	Koleksiyonlar performans açısından dizilerden daha yavaştır.
Bellek açısından dizilerin kullanılması tavsiye edil- mez.	Bellek açısından koleksiyonların kullanılması tavsiye edilir.
Dizilerin elemanları üzerinde işlem yapmak için ko- leksiyonlardan daha az hazır metodu vardır.	Koleksiyonların elemanları üzerinde işlem yapmak için dizilerden daha fazla hazır metodu vardır.







Kodlama yaparken özel isteğe uyarlanmış koleksiyon oluşturulabileceği gibi Nesne Tabanlı Programlama dili ile gelen hazır koleksiyonlar da kullanılabilir. Hazır koleksiyonlardan bazıları şunlardır:

- ArrayList
- List

ÖĞRENME BİRİMİ

- Queue-Stack
- Dictionary
- HashTable
- SortedList

4.2.1. Boxing (Kutulama)-Unboxing (Kutu Açma)

Nesne Tabanlı Programlama dilinde **Value Type** (değer tipi) ve **Reference Type** (referans tipi) olmak üzere iki veri tipi vardır. Değer tipleri; int, char, byte, double vb. belleğinin **Stack** adı verilen kısmında tutulur. Referans tipleri ise object, string, class vb. belleğin **Heap** adı verilen kısmında tutulur.

Boxing, herhangi value (değer) tipideki değişkenin object (nesne) tipindeki değişkene dönüştürülmesidir.

int x = 1234; object obj; obj = x;

Unboxing, object (nesne) tipindeki değişkenin value (değer) tipindeki değişkene dönüştürülmesidir. Burada dikkat edilmesi gereken nokta, dönüştürülmek istenen nesne hangi tipte boxing yapılmış ise aynı tipte **Casting** yapılması gerekmektedir. Aksi durumda veri tipleri aynı olmadığı için dönüştürme hatası (InvalidCastException) alınacaktır.

```
int y = (int)obj;
```

4.2.2. ArrayList Koleksiyonu

ArrayList, dizilerde olduğu gibi veri saklama amacıyla kullanılan bir koleksiyondur. Eleman sayıları dinamik olarak değişir ve farklı tiplerde veri saklama imkânı sunar. ArrayList elemanlarına erişim için index numaraları kullanılır. ArrayList tanımlanırken eleman sayısını belirtmeye gerek yoktur. Dinamik yapısı sayesinde kodlama sırasında veya çalışma anında ekleme, silme, araya ekleme, değerleri değiştirme gibi işlemler yapılabilir. ArrayList içine eklenen elemanlar object tipinde olacağı için veri eklerken boxing (kutulama), veri alırken de unboxing (kutu açma) işlemi yapılmaktadır.

Koleksiyon nesnesi ArrayList'in oluşturulması aşağıda verilmiştir.

ArrayList liste = new ArrayList();

Yukarıdaki komut derlendiğinde ArrayList sınıfından liste adında bir nesne üretilmiş olur.

• ArrayList Veri Ekleme

Koleksiyonlara veri eklemek için **Add** metodu kullanılır. Koleksiyonlar da diziler gibi index numaralarına sahiptir. ArrayList'e ilk eklenen elemanın index numarası 0 olur, diğer elemanlar da 0'dan başlayarak sıralanır.

ArrayList liste = new ArrayList(); liste.Add("Bilişim"); // Metinsel liste.Add(100); // Tam sayı liste.Add('m'); // Karakter liste.Add(3.14); // Ondalık sayı liste.Add(true); // Mantıksal









Şehir isimlerinin saklanacağı bir ArrayList oluşturarak formdan girilen şehir isimlerini tanımlanan ArrayList içine ekleme işlemi yapılacaktır.

Adım 1: Görsel 4.19'da verilen form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

🖶 ArrayList Uygulama 1		_		×
Şehir Adı			Ekle]
	txt SehirAdi		btnEkle	

Görsel 4.19: ArrayList form tasarımı uygulaması-1

Adım 2: Ekle butonu Click olayında global olarak tanımlanan sehirler ismindeki ArrayList'e Text-Box'tan gelen verileri aktarınız.



• ArrayList Elemanlarına Erişim

ArrrayList içindeki elemanlara erişim için dizilerde olduğu gibi index numaraları kullanılır.

sehirler[0] // 0 index numaralı ilk eleman
sehirler[1] // 1 index numaralı ikinci eleman
sehirler[2] // 2 index numaralı üçüncü eleman

Yukarıda gösterildiği gibi index numaraları 0'dan başlayarak eleman sayısının 1 eksiğine kadar istenilen değerlere erişim sağlanabilir. Dizilerde olduğu gibi ArrayList elemanlarına sınırları dışında bir index numarası ile ulaşılmaya çalışıldığında hata ile karşılaşılır.

sehirler[1]="Mardin"; // 1 index numaralı elemana erişim sağlanarak değeri değiştirildi.

Label1.Text=(string)sehirler[1]; // 1 index numaralı elemana erişim sağlanarak unboxing işlemiyle // Label kontrolü içinde gösterilmiştir.





ArrayList

Uygulama-2

ArrayList Uygulama 1 biraz daha geliştirilerek ArrayList nesnesine eklenen veriler index numaraları ile birlikte ListBox kontrolü içinde gösterilecektir.

Adım 1: Görsel 4.20'de verilen form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

🖳 ArrayList Uygulama 2				_		×
Şehir Adı	txt SehirAdi	Ekle btnEkle			Listele btnListe	
			lbListe			

Görsel 4.20: ArrayList form tasarımı uygulaması-2

Adım 2: Listele butonu Click olayında aşağıdaki kodları yazarak listeleme işlemini gerçekleştiriniz.

```
private void btnListe_Click(object sender, EventArgs e)
{
    for (int i = 0; i < sehirler.Count; i++)
    {
        lbListe.Items.Add(sehirler[i]);
    }
}</pre>
```

Uygulamada ArrayList koleksiyonu içindeki değerlere ulaşmak için bir döngü kullanıldı. Bu döngüyle O'dan başlayarak koleksiyonun eleman sayısına kadar dönme işlemi yapıldı. Koleksiyonlarda eleman sayısı Count özelliği ile alınıp döngü değişkeni koleksiyon index numarası olarak kullanıldı ve her elemana erişim sağlandı.

Sıra Sizde >

1. Adım 2'de kullanılan for döngüsü yerine foreach döngüsünü kullanarak aynı işlemleri yapınız.

2. Adım 2'de koleksiyon değerlerinin ListBox içinde gösterilmesinde niçin unboxing yapılmamıştır? Açıklayınız.

Insert metodu, koleksiyonlara değer eklemenin bir diğer yoludur. Bu metodun add metodundan farkı, eklenecek değerin hangi sıraya atanacağının belirtilmesidir. Insert metodu kullanımından sonra koleksiyonun bellek üzerindeki durumu Görsel 4.21'de verilmiştir.

```
ArrayList isimler = new ArrayList();
isimler.Add("Fatma");
isimler.Add("Ayşe");
isimler.Add("Merve");
isimler.Add("Ahmet");
isimler.Add("Kaan");
isimler.Insert(2,"Zeynep");
```


isimler.Insert(2, "Zeynep");

Görsel 4.21: ArrayList insert metodu kullanımı

• ArrayList Veri Silme

ArrayList'lerde bir veriyi silmek için **Remove** ve **RemoveAt** metotları vardır. **Remove** metodunda silinecek nesnenin değeri, **RemoveAt** metodunda ise silinecek nesnenin index numarası kullanılır. Veri silme metodlarının kullanımından sonra koleksiyonun bellek üzerindeki durumu Görsel 4.22'de verilmiştir.



Görsel 4.22: ArrayList Remove ve RemoveAt metodu kullanımı

ArrayList içindeki tüm verileri silmek için Clear() metodu kullanılır.

Sıra Sizde

ArrayList içinde aynı değerlere sahip birden çok eleman varsa Remove metodu aynı değerlere sahip elemanlardan ilk bulunanı siler. Diğer elemanları da silmek için ne yapılmalıdır?

• ArrayList Veri Arama

Bir ArrayList içinde bir verinin aranması için birkaç yöntem bulunmaktadır. Bunlardan bir tanesi **Contains** metodudur. Contains metodu; aranan veri koleksiyonda varsa **true**, yoksa **false** değerini geriye döndürür.

```
if(isimler.Contains("Ahmet"))
{
    label1.Text = "Aranan veri bulundu.";
}
else
{
    label1.Text = "Bulunamadı.";
}
```







ArrayList icinde bir diğer arama yöntemi IndexOf metodudur. Bu metot, Contains metodundan farklı olarak koleksiyon içinde aranan veri bulunursa index numarasını, bulunamazsa -1 değerini geriye döndürür.



Uygulama-3

ÖĞRENME BİRİMİ

Uygulamada sehirler isminde global tanımlanan bir ArrayList üzerinde ekleme, araya ekleme, güncelleme, silme ve arama işlemleri yapılacaktır.

Adım 1: Görsel 4.23'teki form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

ygulama-3							Kodse
Jygulamada s e, güncelleme	ehirler ismin e, silme ve a	de global tanır rama işlemleri	mlanan bir A yapılacaktır.	rrayList üzeri	nde ekleme	e, ara-	The second second second second second second second second second second second second second second second se
Adım 1: Görs iniz.	sel 4.23'teki	form tasarımır	nı yapınız ve	form içindeki	kontrollere	name (değer-
🖳 ArrayList	Uygulama 3				—		^ک ×
Şehirler		txtSehirler					
Ekle	Araya Ekle	Güncelle	Sil	Ara			
btnEkle	btnAraEkle	btnGuncelle	btn Sil	btn.Ara			
Durum							
aberburum							
					liste	Sehirler	

http://kitap.eba.gov.tr/

Görsel 4.23: ArrayList form tasarımı uygulaması-3

Adım 2: Ekle butonu Click olayında global olarak tanımlanan sehirler ismindeki ArrayList'e Text-Box'tan gelen verileri aktarınız.

ArrayList sehirler = new ArrayList();
private void btnEkle_Click(object sender, EventArgs e)
{
sehirler.Add(txtSehirler.Text);
Listele();
}

Adım 3: Ekle butonu içinde Listele isminde bir metot kullanınız. Bu metodun görevi; eklenen, silinen veya değeri değiştirilen ArrayList elemanlarını ListBox kontrolü içinde göstermektir.

```
private void Listele()
{
  listeSehirler.Items.Clear();
  foreach (string sehir in sehirler)
  ł
     listeSehirler.Items.Add(sehir);
  }
```

> **Adım 4:** ListBox içindeki elemanın index numarası, ArrayList elemanları ile aynıdır. Araya Ekle butonu Click olayında ListBox içinden seçilen elemanın index numarası, **SelectedIndex** özelliği kullanılarak bir değişkene aktarılır. Bu değişkeni **insert** metodu içinde kullanarak araya ekleme işlemini gerçekleştiriniz.

```
private void btnAraEkle_Click(object sender, EventArgs e)
{
    int indexNo = listeSehirler.SelectedIndex;
    sehirler.Insert(indexNo, txtSehirler.Text);
    Listele();
}
```

Adım 5: Güncelle butonu Click olayında ListBox içinden seçilen elemanın index numarasını kullanarak ArrayList elemanlarının değerlerini değiştiriniz.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    int indexNo = listeSehirler.SelectedIndex;
    sehirler[indexNo] = txtSehirler.Text;
    Listele();
}
```

Adım 6: Sil butonu Click olayında ListBox üzerinden index numarası alınan ArrayList elemanını silme işlemini yapınız.

```
private void btnSil_Click(object sender, EventArgs e)
{
    int indexNo = listeSehirler.SelectedIndex;
    sehirler.RemoveAt(indexNo);
    Listele();
}
```

Adım 7: Ara butonu Click olayında TextBox'a yazılan değeri ArrayList içinde arama işlemini yapınız. Aranan değer bulunursa labelDurum'a "Aranan Değer Bulundu." mesajı verilecek ve **IndexOf** metodu kullanılarak ListBox içindeki konumuna gidecektir. Aranan değer bulunamazsa "Aranan Değer Bulunamadı." mesajı verilecektir.

```
private void btnAra_Click(object sender, EventArgs e)
{
    if (sehirler.Contains(txtSehirler.Text))
    {
        labelDurum.Text = "Aranan Değer Bulundu.";
        listeSehirler.SelectedIndex = sehirler.IndexOf(txtSehirler.Text);
    }
    else
    {
        labelDurum.Text = "Aranan Değer Bulunamadı.";
    }
}
```







ArrayList içindeki veriler, eklenme sırasına göre 0 index numarasından başlayarak devam eder. **Reverse** metodu bu sıralamayı tamamen tersine çevirmektedir.

sehirler.Reverse();

ÖĞRENME BİRİMİ

Bir diğer sıralama ise **Sort** metodudur. Sort metodu, ArrayList içindeki değerleri artan bir sıra hâlinde yeniden düzenlemektedir. Sort metodu; koleksiyondaki elemanlar yazı tipinde ise a'dan z'ye, sayısal tipte ise küçükten büyüğe doğru sıralar.

Sıra Sizde

ArrayList Uygulama 3'teki forma iki adet buton ekleyiniz. Butonlardan birinin görevi sıralama, diğerinin görevi ise tersten sıralama olsun. Bu butonlara tıklandığında ListBox içinde ArrayList'in sıralanmış veya tersten sıralanmış hâllerini gösteriniz.

4.2.3. List Koleksiyonu

List koleksiyonu ile ArrayList koleksiyonu benzerlik gösterir. Aralarındaki fark; List koleksiyonunun generic, ArrayList koleksiyonunun ise non-generic yapıda olmasıdır. Generic koleksiyonlarda mutlaka içinde saklanacak verinin tipi belirtilmelidir çünkü belirlenen tipin dışında veri saklanmaya çalışıldığında hata ile karşılaşılır. Non-generic koleksiyonlarda ise bu işlem yapılmaz. Non-generic koleksiyonlarda veri tipi obje olarak belirlenir ve bu koleksiyonlarda her tipten veri saklanabilir. Generic koleksiyonlarda boxing-unboxing işlemleri yapılmazken non-generic koleksiyonlarda yapılır.

Uy	gulama-1							odSor.php
Bu laştırılacak	ı uygulamada Array tır.	List koleks	iyonu ile Lis	st koleksiyonu işler	m süresi olara	ak karşı-		KOD=210
Ac	dım 1: Görsel 4.24 gulama 1	'teki form t	asarımını y	apınız ve form için	deki kontrolle	ere name –	değerle	× 8
btnAEkle btnLEkle	ArrayList Değer Ekle List Değer Ekle	label1 IbIAEk	ile btnAYaz le btnLYaz	ArrayList Değer Yaz List Değer Yaz	label1 lblAYaz label1 lblLYaz			
btnATopla btnLTopla	ArrayList Değer Topla List Değer Topla	label1 IbIATo	pla btnABul pla btnLBul	ArrayList Değer Bul List Değer Bul	label1 lblABul label1 lblLBul			
						listeA	liste	eL

Görsel 4.24: List form tasarımı uygulaması-1

Adım 2: Bu uygulamada global olarak bir ArrayList, bir List koleksiyonu, her iki koleksiyonun eleman sayılarını belirleyecek integer değişken ve işlemlerde geçen süreyi hesaplamak için Stopwatch nesnesi kullanılmıştır. Stopwatch nesnesi için "using System.Diagnostics" isim uzayını ekleyiniz.

ArrayList kolA = new ArrayList();	// ArrayList koleksiyonu
List <int> kolL = new List<int>();</int></int>	// List koleksiyonu
Stopwatch km = new Stopwatch();	// Geçen süreyi hesaplama için nesne
int elemanSayisi = 100000;	// Test için eleman sayısı

:/



private void btnAEkle_Click(object sender, EventArgs e)	private void btnLEkle_Click(object sender, EventArgs e)
{	{
km.Start();	km.Start();
for (int i = 0; i < elemanSayisi; i++)	for (int i = 0; i < elemanSayisi; i++)
{	{
kolA.Add(i);	kolL.Add(i);
}	}
km.Stop();	km.Stop();
IbIAEkle.Text = "= " +	lbILEkle.Text = "= " +
km.Elapsed.TotalMilliseconds;	km.Elapsed.TotalMilliseconds;
km.Reset();	km.Reset();
}	}

private void btnATopla_Click(object sender, EventArgs e)	private void btnLTopla_Click(object sender, EventArgs e)
{	{
km.Start();	km.Start();
int toplam = 0;	int toplam = 0;
for (int i = 0; i < kolA.Count; i++)	for (int i = 0; i < kolL.Count; i++)
{	{
toplam +=(int)kolA[i];	toplam += kolL[i];
}	}
km.Stop();	km.Stop();
lbIATopla.Text = "= " +	lblLTopla.Text = "= " +
km.Elapsed.TotalMilliseconds;	km.Elapsed.TotalMilliseconds;
km.Reset();	km.Reset();
}	}

private void btnAYaz_Click(object sender, EventArgs	private void btnLYaz_Click(object sender, EventArgs
e)	e)
{	{
km.Start();	km.Start();
for (int i = 0; i < elemanSayisi; i++)	for (int i = 0; i < kolL.Count; i++)
{	{
listeA.Items.Add(kolA[i]);	listeL.Items.Add(kolL[i]);
}	}
km.Stop();	km.Stop();
lbIAYaz.Text= "= " +	lbILYaz.Text = "= " +
km.Elapsed.TotalMilliseconds;	km.Elapsed.TotalMilliseconds;
km.Reset();	km.Reset();
}	}

private void btnABul_Click(object sender, EventArgs	private void btnLBul_Click(object sender, EventArgs
e)	e)
{	{
km.Start();	km.Start();
if(kolA.Contains(9999))	if (kolL.Contains(9999))
{	{
km.Stop();	km.Stop();
}	}
lbIABul.Text = "= " +	lblLBul.Text = "= " +
km.Elapsed.TotalMilliseconds;	km.Elapsed.TotalMilliseconds;
km.Reset();	km.Reset();
}	}









ÖĞRENME BİRİMİ

ArrayList ile List arasındaki işlem süresi sonuçlarını karşılaştırıp yorumlayınız.

4.2.4. Queue-Stack Koleksiyonları

Queue kelimesi kuyruk anlamına gelir. İlk giren eleman ilk çıkar işleyişine sahip bir koleksiyondur (FIFO-First In First Out). Koleksiyondan bir eleman çıkarılmak istendiğinde kuyruğa ilk eklenen eleman çıkartılacaktır. Yeni eklenecek eleman ise kuyruğun en sonuna getirilir.

Aslında bu yapıyla günlük hayatta çokça karşılaşılır. Örneğin, sıra numarasına göre işlem yapılan bir işletmede önce bir sıra numarası alınır. Alınan bu sıra numarası kuyruğun en sonundadır. İşlem yapılan numaralar sıradan çıkar ve işlem sırası diğer numaralara gelir.

Queue ekleme ve çıkarma işleminin iki metodu vardır:

- 1. Enqueue() metodu kuyruğun sonuna bir eleman ekler.
- 2. Dequeue() metodu kuyruğun başındaki elemanı çıkarır.



Queue Koleksiyonu

Uygulamada Queue sınıfından üretilen nesne ile iki Button bir ListBox kontrolü kullanılarak kuyruğa alma ve kuyruktan çıkarma işlemleri yapılacaktır.

Adım 1: Görsel 4.25'teki form tasarımını yapınız ve form üzerindeki kontrollerin name değerlerini veriniz.



Görsel 4.25: Queue form tasarımı uygulaması-1

Adım 2: Queue sınıfından üretilecek nesneyi ve bu nesneye aktarılacak sıra numaraları için global değişkenleri oluşturunuz.

Queue kuyruk = new Queue(); int sira = 0;



Adım 3: Sıra Al butonu Click olayında sira değişkeninin değerini bir artırarak oluşturulan Queue nesnesine aktarınız.

```
private void btnSiraAl_Click(object sender, EventArgs e)
{
    sira++;
    kuyruk.Enqueue(sira);
    Listele();
}
```

Adım 4: Queue nesnesindeki değerleri ListBox içinde göstermek için Listele adında bir metot oluşturarak sıra durumunu bu metot içinde gösteriniz.

```
private void Listele()
{
    listeSiralar.Items.Clear();
    foreach (int sira in kuyruk)
    {
        listeSiralar.Items.Add(sira);
    }
}
```

ÖĞRENME BİRİMİ

Adım 5: Sıra Çıkart butonu Click olayında oluşturduğunuz Queue nesnesine verilen ilk değeri kuyruktan çıkarma işlemini gerçekleştiriniz. Listele metodunu kullanarak en güncel kuyruk değerlerini ListBox içinde gösteriniz.

```
private void btnSiraCikart_Click(object sender, EventArgs e)
{
    kuyruk.Dequeue();
    Listele();
}
```

Sıra Sizde

Adım 5'te kuyrukta hiç sıra yokken çıkartma işlemi yapıldığında nasıl bir hata ile karşılaşılır? Bu hatanın çözümü için ne yapılabilir? Önerilerinizi sınıfınızla paylaşınız.

Stack kelimesi yığın anlamına gelir. Son giren eleman ilk çıkar işleyişine sahip bir koleksiyondur (LIFO-Last In First Out). Koleksiyondan bir eleman çıkarılmak istendiğinde yığına son eklenen eleman çıkartılacaktır. Örneğin, ofis uygulamalarında yapılan işlemler uygulama tarafından hafızaya alınır. Uygulamada geri alma işlemi yapıldığında en son işlem gerçekleştirilir. Geri alma işlemine devam edildiğinde sondan başa doğru hafızdaki işlemler gerçekleştirilir.

Stack içine ekleme ve çıkarma işleminin iki metodu vardır:

- 1. Push() metodu Stack içine bir değer ekler.
- 2. Pop() metodu Stack içine eklenen son değeri çıkarır.



10, SINIF



Uygulama-2

Stack Uygulama

Uygulamada Stack sınıfından üretilen nesne ile iki Button bir ListBox kontrolü kullanarak yığına alma ve yığından çıkarma işlemleri yapılacaktır.

Adım 1: Görsel 4.26'daki form tasarımını yapınız ve form üzerindeki kontrollerin name değerlerini veriniz.

•	Stack Uygulama 2		_		×
	Ad S	oyad dSoyad			
	Ekle	Çıkart			
	btnEkle	btnCikart			
				listeStack	

Görsel 4.26: Stack form tasarımı uygulaması-2

Adım 2: Stack sınıfından üretilecek nesneyi global olarak oluşturunuz.

```
Stack yigin = new Stack();
```

Adım 3: Ekle butonu Click olayında TextBox içinden alınan değerleri Stack içine aktarınız.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    yigin.Push(txtAdSoyad.Text);
    Listele();
}
```

Adım 4: Stack içindeki değerleri ListBox içinde göstermek için Listele adında bir metot oluşturarak listeleme işlemlerini yapınız.

```
private void Listele()
{
    listeStack.Items.Clear();
    foreach (string eleman in yigin)
    {
        listeStack.Items.Add(eleman);
    }
}
```





Adım 5: Çıkart butonu Click olayında Stack içinden son değeri çıkarınız. Listele metodunu kullanarak en güncel Stack değerlerini ListBox içinde gösteriniz.

```
private void btnCikart_Click(object sender, EventArgs e)
{
    yigin.Pop();
    Listele();
}
```

4.2.5. Dictionary Koleksiyonu

Dizi ve ArrayList içine eklenen elemanlar bellek üzerine sıralı olarak yerleşir. Dizi ile ArrayList elemanlarına 0'dan başlayarak index numarası verilir ve elemanlara erişim için bu index numaraları kullanılır. Dictionary koleksiyonunda index-değer ilişkisi Key (Anahtar) - Value (Değer) olarak kullanılmaktadır. Key dizilerdeki index numarası, Value ise index numarası ile belirtilen değer olarak düşünülebilir. Key benzersiz olmak şartıyla int, string, byte, object vb. olabilir.

Aşağıda Dictionary koleksiyonunun kullanımı verilmiştir.

Dictionary<KeyTipi,ValueTipi> koleksiyon_adı = new Dictioanary<KeyTipi,ValueTipi>();

Aşağıdaki kodlamalarda Dictionary nesnesi tanımlaması ve değer aktarımında Key-Value tiplerinin farklı kullanımları gösterilmiştir.

```
Dictionary<int,string> sehirler = new Dictionary<int,string>();
sehirler.Add(1,"Adana");
sehirler.Add(2,"Adıyaman");
```

```
Dictionary<string, int> sehirler = new Dictionary<string, int>();
sehirler.Add(``Adana'', 1);
sehirler.Add(``Adıyaman'', 2);
```



ÖĞRENME BİRİMİ

Uygulama-1

Dictionary Koleksiyonu

Bu uygulamada okul öğrencilerinin numaralarını ve isimlerini bir Dictionary koleksiyonuna ekleme, silme, güncelleme ve arama işlemleri gerçekleştirilecektir.









Adım 1: Görsel 4.27'deki form tasarımını yapınız ve form üzerindeki kontrollerin name değerlerini

veriniz.

🖳 Dictionary Uygulama	1		-		Х
Okul No Ad Soyad	txtOkulNo	txtAdSoyad			
btnAra btnEkle	Ara Ekle				
btnGuncelle	Güncelle	, 			
btnSil	51				
			listeO	grenciler	

Görsel 4.27: Dictionary form tasarımı uygulaması-1

Adım 2: Bu uygulamada Dictionary nesnesinin anahtarını okul numarası ve integer tipinde, değerini ise öğrencilerin adı soyadı ve string tipinde tanımlayınız.

```
Dictionary<int,string> ogrenciler = new Dictionary<int, string>();
int anahtar;
string deger;
```

Adım 3: Ekle butonu Click olayında okul numaralarını Key, öğrencilerin adı soyadını Value olarak koleksiyona ekleyiniz.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    anahtar = int.Parse(txtOkulNo.Text);
    deger = txtAdSoyad.Text;
    ogrenciler.Add(anahtar,deger);
    Listele();
}
```



Adım 4: Dictionary koleksiyonundaki Key ve Value bilgilerini ListBox içinde göstermek için Listele isminde bir metot kullanınız. Bu metot, foreach döngüsü içinde ogrenciler koleksiyonundaki tüm Key ve Value bilgilerini var tipinde ismi ogrenci olan nesneye sırasıyla aktaracaktır. Bu metotla döngü her döndüğünde ogrenci nesnesini kullanarak ListBox içine Key ve Value bilgileri eklenecektir.

```
private void Listele()
{
    listeOgrenciler.Items.Clear();
    foreach (var ogrenci in ogrenciler)
    {
        listeOgrenciler.Items.Add(ogrenci.Key+"-"+ ogrenci.Value);
    }
```

ÖĞRENME BİRİMİ

Adım 5: Güncelle butonu Click olayında okul numarasına göre öğrenci isimlerini güncelleme işlemini gerçekleştiriniz.

```
private void btnGuncelle_Click(object sender, EventArgs e)
{
    anahtar = int.Parse(txtOkulNo.Text);
    deger = txtAdSoyad.Text;
    ogrenciler[anahtar] = deger;
    Listele();
}
```

Adım 6: Sil butonu Click olayında okul numarasına göre Dictionary koleksiyonundan öğrenci bilgilerini silme işlemini gerçekleştiriniz.

```
private void btnSil_Click(object sender, EventArgs e)
{
    anahtar = int.Parse(txtOkulNo.Text);
    ogrenciler.Remove(anahtar);
    Listele();
}
```



Adım 7: Ara butonu Click olayında okul numarasının veya öğrenci adının koleksiyon içinde bulunup bulunmadığını kontrol ediniz, okul numarası veya öğrenci adı varsa mesaj olarak gösterme işlemini yapınız.

```
private void btnAra_Click(object sender, EventArgs e)
{
  bool durum=false;
  if(txtOkulNo.Text!="")
   {
    anahtar = int.Parse(txtOkulNo.Text);
    durum = ogrenciler.ContainsKey(anahtar);
  }
  else
   {
    deger = txtAdSoyad.Text;
    durum = ogrenciler.ContainsValue(deger);
   }
  if(durum==true)
   {
    MessageBox.Show("Öğrenci Kayıtlıdır.");
   }
   else
   {
    MessageBox.Show("Öğrenci Kayıtlı Değildir.");
   }
```

Sıra Sizde >

ÖĞRENME BİRİMİ

Dictionary Koleksiyonu Uygulama 1'de koleksiyon tanımlanırken öğrencinin adı soyadı Key olarak kullanılabilir mi? Açıklayınız.

4.2.6. Hashtable Koleksiyonu

Hashtable, non-generic bir koleksiyondur. Dictionary koleksiyonunda olduğu gibi Key-Value tipleri belirtilmez. Hashtable, karışık tablo anlamına gelmektedir. Bu koleksiyona yalnızca bir defa kullanılmak şartıyla Key değeri olarak istenilen herhangi bir değer atanabilir.

Hashtable koleksiyonunda kullanılan metot ve özellikler aşağıda verilmiştir.

1. Hashtable nesnesi oluşturma

Hashtable ogrenciler = new Hashtable();



2. Hashtable koleksiyonuna veri ekleme

ÖĞRENME BİRİMİ

> ogrenciler.Add(368, "Ahmet"); ogrenciler.Add("Ahmet",368);

3. Hashtable koleksiyonundan veri silme

ogrenciler.Remove(368);
ogrenciler.Remove("Ahmet");

4. Hashtable koleksiyonunda veri güncelleme

ogrenciler[368] = "Mehmet";

5. Hashtable koleksiyonunda Key içeriğini listeleme

foreach (var anahtar in ogrenciler.Keys)
{
 Console.WriteLine(anahtar);

6. Hashtable koleksiyonunda Value içeriğini listeleme

```
foreach (var deger in ogrenciler.Values)
{
    Console.WriteLine(deger);
}
```

7. Hashtable koleksiyonu Key ve Value içeriğini listelemek için DictionaryEntry isimde bir sınıf kullanılarak koleksiyon içindeki elemanların hem Key hem de Value değerlerine erişim sağlanabilir.

```
foreach (DictionaryEntry eleman in ogrenciler)
{
   Console.WriteLine(eleman.Key + " - " + eleman.Value);
}
```

Sıra Sizde

Dictionary Koleksiyonu Uygulama 1'de hazırladığınız formu Hashtable koleksiyonunu kullanarak yeniden yapınız.

4.2.7. SortedList Koleksiyonu

SortedList koleksiyonu, Hashtable koleksiyonuna benzer. SortedList koleksiyonunun en önemli özelliği, içindeki verileri anahtarın içeriğine göre sıralamasıdır. Karışık olarak verilen anahtar değerlerini bile kendi içinde artan şekilde bir sıralama yaparak koleksiyon içine ekler.

SortedList koleksiyonunda kullanılan metot ve özellikler aşağıda verilmiştir.

SortedList nesnesi oluşturma

```
SortedList ogrenciler = new SortedList();
```



10, SINIF



2. SortedList koleksiyonuna veri ekleme

ogrenciler.Add(368, "Ahmet");

3. SortedList koleksiyonundan veri silme

ogrenciler.Remove(368);

4. SortedList koleksiyonunda veri güncelleme

ogrenciler[368] = "Mehmet";

5. SortedList koleksiyonu Key ve Value içeriğini listelemek için DictionaryEntry isimde bir sınıf kullanılarak koleksiyon içindeki elemanların hem Key hem de Value değerlerine erişim sağlanabilir.

foreach (DictionaryEntry eleman in ogrenciler)
{
 Console.WriteLine(eleman.Key+" - "+eleman.Value);
}

Sıra Sizde

BİRİMİ

Dictionary Koleksiyonu Uygulama 1'de hazırladığınız formu SortedList koleksiyonunu kullanarak yeniden yapınız.





ÖLÇME VE DEĞERLENDİRME - 4

1. Aşağıdaki kod satırında integer tipinde numaralar isminde bir dizi oluşturmak için boşluklara doğru ifadeleri yazınız.

_={10,20,30,40};

2. Aşağıdaki kod satırlarında string tipinde sehirler isminde 81 elemanlı bir dizi oluşturmak için boşluklara doğru ifadeleri yazınız.

= new

3. Aşağıda tanımlanan diziye göre x değişkenin değeri ne olur?

byte[] sayilar = new byte[]{ 4,3,2,1};

byte x=sayilar[3];

4. int[] sayilar = {10, 32, 60, 100, 90, 5}; dizisinde sayilar.Length komutunun değeri aşağıdakilerden hangisidir?

A) 7 **B)** 6 **C)** Hata verir. **D)** 10 **E)** 5

5. Aşağıdakilerden hangisi 4 elemanlı integer tipinde tanımlanmış bir dizidir?

A) int[] dizi = new int[4];
 B) int[4] dizi;
 C) int[] dizi = 4;
 D) int dizi[4];
 E) int[] dizi=new byte[4];

6. Dizilerle ilgili aşağıdaki ifadelerden hangisi doğrudur?

A) Dizi elemanları sadece integer tipinde olur.

B) Dizinin Rank özelliği dizideki eleman sayısını verir.

C) Dizinin Length özelliği dizinin boyutunu verir.

D) Sayısal dizilerde ön tanımlı değeri 0'dır.

E) Dizilere istenilen tipte değer verilebilir.

7. Aşağıdakilerden hangisi 4 satır ve 5 sütundan oluşan iki boyutlu dizi tanımlamasıdır?

A) int[4, 5] sayilar = new int[,];
B) int [4][5] sayilar = new int[];
C) int [,] sayilar = new int[4, 5];
D) int [,] sayilar = new int[5, 4];
E) int[5,4] sayilar = new int[];
8. Aşağıdakilerden hangisi 2 satır ve 3 sütunlu bir dizinin doğru tanımlanmış hâlidir?

A) int[,] sayi= new int[2, 3]{{1, 2},{10,11},{ 5, 6}};
B) int[,] sayi = new int[2, 3]{};
C) int[,] a = new int[1, 2];
D) int[,] a = new int[1, 2]{{7, 1, 9}, {2, 5, 6}};
E) int[,] sayi= new int[2, 3]{{7, 1, 9},{2, 5, 6}};



int[] sayila	ar = { 1, 4, 8, 2]	};			
int toplam	1 = 0;				
for (int in {	idex=0; index <	sayilar.Length; ii	ndex++)		
toplam = }	toplam + sayilaı	r[index] ;			
Yukarıda	verilen kodlar o	çalıştırıldığında	a toplam değişker	nin değeri aşağıdal	kilerden hangisi
A) 15	B) 13	C) 5	D) 1 4 8 2	E) 0	
10. Aşağı	ıdakilerden har	ngisi bir Arrayl	.ist tanımlamasıd	ır?	
A) ArrayLis	st[] liste=new Arı	rayList();			
B) ArrayLis	st liste=new Arra	yList();			
C) ArrayLis	st liste=ArrayList((); wlist[10]·			
E) ArrayLis	st() liste=new Arr	rayList();			
	U	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
11. Aşağı	dakilerden han	gisi bir ArrayL	ist içine veri ekle	me metodudur?	
11. Aşağı A) liste.Add	dakilerden han d(``abc″)	igisi bir ArrayL B) liste.Set	ist içine veri ekle :(``abc")	C) liste.Size("abc")
11. Aşağı A) liste.Ado	dakilerden han d(``abc") D) liste.Cou	gisi bir ArrayL B) liste.Set unt("abc")	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
11. Aşağı A) liste.Add 12.	dakilerden han d("abc") D) liste.Cou	gisi bir ArrayL B) liste.Set unt("abc")	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 Aşağı A) liste.Add 12. ArrayList I 	dakilerden han d("abc") D) liste.Cou	gisi bir ArrayL B) liste.Set unt("abc") /List();	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 Aşağı A) liste.Add Iz. ArrayList I liste.add(`` 	dakilerden han d("abc") D) liste.Cou liste = new Array 'Ali");	gisi bir ArrayL B) liste.Set unt("abc") /List();	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` 	dakilerden han d(``abc") D) liste.Cou liste = new Array `Ali"); `Elif");	gisi bir ArrayL B) liste.Set unt("abc") /List();	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` 	dakilerden han d(``abc") D) liste.Cod liste = new Array `Ali"); `Elif"); `Can");	gisi bir ArrayL B) liste.Set unt("abc") /List();	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` liste.add(`` 	dakilerden han d(``abc") D) liste.Cod liste = new Array `Ali"); `Elif"); `Can"); `Fatih");	gisi bir ArrayL B) liste.Set unt("abc") /List();	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc"))
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` liste.add(`` Yukarıda mesi için 	dakilerden han d(``abc") D) liste.Cou liste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc") erden sonra aşağı) Idaki listeye ber
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` Vukarıda mesi için I Ali 	dakilerden han d(``abc") D) liste.Cou liste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc") E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc")) Idaki listeye ber
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` liste.add(`` Yukarıda mesi için I Ali Elif 	dakilerden han d(``abc") D) liste.Cou liste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc")) Idaki listeye bel
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` Vukarıda mesi için I Ali Elif Mehmet 	dakilerden han d(``abc") D) liste.Cou iste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc")) Idaki listeye ber
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` Vukarıda mesi için I Ali Elif Mehmet Can 	dakilerden han d(``abc") D) liste.Cou iste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc")) Idaki listeye bel
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` liste.add(`` Yukarıda mesi için I Ali Elif Mehmet Can Fatih 	dakilerden han d(``abc") D) liste.Cou liste = new Array 'Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pilmıştır?	ist içine veri ekle :(``abc'') E) liste.Get(``a	me metodudur? C) liste.Size("abc" bc")) Idaki listeye bel
 11. Aşağı A) liste.Add 12. ArrayList I liste.add(`` liste.add(`` liste.add(`` Yukarıda mesi için I Ali Elif Mehmet Can Fatih A) liste.Add 	dakilerden han d(``abc") D) liste.Cou liste = new Array `Ali"); `Elif"); `Can"); `Fatih"); yer alan kodla hangi işlem ya	rda koleksiyor pılmıştır?	ist içine veri ekle :(``abc'') E) liste.Get(``a na eklenen değer	tr") C) liste.) Idaki listeye ber



liste.add(``Ali"); liste.add(``Elif"); liste.add(``Can"); liste.add(``Fatih") liste.Insert(0,"Me	new ArrayList(); ; hmet");			
13. Yukarıda ye A) Ali	r alan kodlamala B) Can	ra göre liste[2] deg C) Mehmet	jeri aşağıdakiler D) Elif	den hangisidir? E) Fatih
ArrayList liste = liste.add(``Ali''); liste.add(``Elif''); liste.add(``Can'') liste.add(``Fatih' liste.add(``Mehn	new ArrayList(); ; ;; ;; ;; ;; ;; ;; ;; ;;			
değiştirir?				isi Can ismini Ayşe olar
D) liste(2)="Ayşe";	E) liste.	Set("Can","Ayşe");	, <u> </u>
ArrayList liste = r liste.add(``Ali"); liste.add(``Elif"); liste.add(``Can");	new ArrayList(); ; ;et");			
liste.add("Fatih") liste.add("Mehme				
liste.add(``Fatih'') liste.add(``Mehme Yukarıda yer ala mesi için hangi Ali Elif Fatih Mehmet	ın kodlamada ko işlem yapılmıştır	leksiyona eklenen d ?	leğerlerden sonr	a aşağıdaki listeye ben
liste.add("Fatih") liste.add("Mehme Yukarıda yer ala mesi için hangi Ali Elif Fatih Mehmet A) liste.RemoveAt	i n kodlamada ko işlem yapılmıştır (2); B) I) liste.Remove(2);	leksiyona eklenen d ? iste.Remove(2,"Can"); E	eğerlerden sonr C) liste.Rei liste.RemoveAt(a aşağıdaki listeye ben move(``Can",2) `Can");

:./



\int								
16.								
Arra	ArrayList liste = new ArrayList();							
liste liste liste liste liste	liste.add("Ali"); liste.add("Elif"); liste.add("Can"); liste.add("Fatih"); liste.add("Mehmet");							
for (int index = 0; ind	lex < liste	; ind	ex++)				
{								
lbi	Liste.Items.Add(lis	ste[]);						
}								
Yuka ru ol	ırıda yer alan ko arak verilmiştir	odlamalarda bo ?	şluklar	a gelecek ifade	eler aşağıda	kilerder	n hangisinde do	
A) L	ength, index	B) Count, inde	х	C) Size, liste	D) Count,	liste	E) Size, index	
17.0)ueue koleksivo	onu icin asağıdı	aki ifadı	elerden hangis	i doărudur?	,		
Δ) Η	er eklenen elemar	n koleksivon tarat	findan of	omatik olarak su	alanır			
B) İII		n cıkar			alarini			
	va Push() ma	totları kullanılır						
	p() ve Push() me							
א (ט	hantar (Key)-Dege	er (value) ikilisine	e saniptir					
E) İlk	giren eleman ilk	çıkar.						
18. <i>A</i>	şağıdakilerden	hangisinde Die	ctionary	nesnesinin ta	nımlaması	doğru ol	arak yapılmıştır	
A) Di	ctionary[int,string] nesne = new [Dictionar	y[int, string]();				
B) Di	ctionary <int,strin< td=""><th>g> nesne = new</th><td>Dictiona</td><td>ary(int, string);</td><td></td><td></td><td></td></int,strin<>	g> nesne = new	Dictiona	ary(int, string);				
C) Di	ctionary(int,string	g) nesne = new [Dictionar	y <int, string="">();</int,>				
D) Di	ctionary <int,strin< td=""><th>ig> nesne = new</th><td>Dictiona</td><td>ary<int, string="">(</int,></td><td>);</td><td></td><td></td></int,strin<>	ig> nesne = new	Dictiona	ary <int, string="">(</int,>);			
F) Di	ctionary <int strin<="" td=""><th>a> nesne = new</th><td>Dictiona</td><td>arv<int strina="">'</int></td><td></td><td></td><td></td></int>	a> nesne = new	Dictiona	arv <int strina="">'</int>				
10 L	lashtable kolek	sivonu ile Dicti	onaryk	oleksivonu ara	sındaki far	k asağıd	laki coconoklor-	
den hangisinde verilmiştir?								
	Hashtable Kol	eksiyonu	Dictio	onary Koleksiy	onu			
A)	İlk giren eleman	ilk çıkar.	Son g	iren eleman ilk ç	kar.			
B)	Sıralama yapar.		Sıralaı	ma yapmaz.				
C)	Veri tipleri belirti	ilmez.	Veri ti	pleri belirtilir.				
D)	Anahtar değeri s	sayı olmalıdır.	Anaht	ar değeri sayı olı	nayabilir.			
E)	E) Aynı Key değeri verilebilir. Aynı Key değeri verilemez.							







FORM UYGULAMALARI

NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Form sınıfı ve kontrol sınıfı kavramlarını açıklayacak, •
- Form uygulamalarını yapacak, •
- Masaüstü uygulamaları için menü oluşturacak, •
- Veri doğrulama işlemlerini yapacak, •
- Kontrollere veri bağlama işlemlerini öğreneceksiniz. •

ANAHTAR KELİMELER

Windows form uygulamaları, form sınıfı, kontrol sınıfı, menüler, veri doğrulama, veri bağlama







HAZIRLIK ÇALIŞMALARI

- 1. Windows ortamında çalışan uygulamalardan hatırladıklarınızı arkadaşlarınızla paylaşınız.
- 2. Windows uygulamalarında bulunan nesneler neler olabilir?

5.1. FORMLAR

Herhangi bir etkileşimli uygulamada en az bir tane kullanıcı arabirimi bulunmaktadır. C# programlama dili ile geliştirilen Windows form uygulamalarında bu arabirimler için Form sınıfından üretilen nesneler kullanılmaktadır. Formlar; TextBox, Button, Label gibi kontrolleri üzerinde tutarak bir arabirim oluşturmak için kullanılır. Kod editörü ile Windows form uygulama projesi hazırlandığında Form1 ismindeki form otomatik olarak oluşturulmaktadır.

Tüm uygulamalar, belirlenen bir noktadan itibaren çalışmaya başlar. Windows form uygulamalarında başlangıç noktası, Solution Explorer penceresinde **Program.cs** dosyası içindeki **Main** isimli fonksiyon ile belirlenmektedir.

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

Program.cs dosyasında bulunan yukarıdaki kodlamalarda **Application.Run** metodu, uygulamanın hangi form ile başlayacağını belirtmektedir.

Windows form uygulamalarında kullanılan formlar ve kontroller, **Form** sınıfından oluşturulan nesnelerdir. Bu nesnelerin ait oldukları sınıfın özellikleri (properties), olayları (events) ve metotları (methods) uygulama içinde ihtiyaca göre kullanılmaktadır. Form ile kontrollerin bazı özellik, olay ve metotları aynı, bazıları ise farklıdır.

5.1.1. Form Sınıfı

Windows form uygulamalarında System.Windows.Forms isim uzayındaki Form sınıfından üretilen form nesneleri kullanılır. Projeye bir form eklendiğinde Form sınıfının özeliklerini miras alan yeni Form sınıfı, kod editörü tarafından oluşturulmaktadır.

public partial class Form1 : Form

Oluşturulan form sınıfının içinde sınıf ismiyle aynı olan **Yapıcı Metot** (Contructor) bulunmaktadır. Yapıcı Metot içinden **InitializeComponent** isminde bir metot çağrılmaktadır. InitializeComponent metodu form ile birlikte oluşturulan **Form_İsmi.Designer.cs** dosyası içinde bulunmaktadır. Bu metot, form tasarımı sırasında form içine kontroller eklendiğinde veya kontrollerin özellikleri değiştirildiğinde arka planda kodlar üzerinde güncelleme yapmaktadır. Form açıldığında form ve forma eklenen kontrollerin özellikleri ve olayları bu metot içindeki kodlara göre düzenlenmektedir. Örneğin; formun boyutları görsel olarak değiştirildiğinde, forma bir kontrol eklendiğinde veya özelliklere değer verildiğinde kod editörü tarafından InitializeComponent metodu otomatik olarak güncellenmektedir.





FORM UYGULAMALARI

ÖĞRENME BİRİMİ

Aşağıda Tablo 5.1, Tablo 5.2 ve Tablo 5.3'te Form sınıfında kullanılan temel özellikler, metotlar ve olaylar verilmiştir.

AcceptButton	Form aktif iken enter tuşuna basıldığında belirlenen bir butona tıklama işlemi yapar.				
CancelButton	Form aktif iken esc tuşuna basıldığında belirlenen bir butona tıklama işlemi yapar.				
ControlBox Formun sağ üst köşesindeki kontrol butonlarını gösterir veya gizler.					
FormBorderStyle	Formun kenarlık stilini ayarlar.				
MinimizeBox	Formun sağ üst köşesindeki küçültme butonunu aktif veya pasif yapar.				
MaximizeBox	Formun sağ üst köşesindeki büyültme butonunu aktif veya pasif yapar.				
ShowInTaskbar	Görev çubuğunda formun görünüp görünmeyeceğini belirler.				
StartPosition	Çalışma anında formun başlangıç konumunu belirler.				
WindowState	Başlangıçta formun nasıl görüntüleneceğini ayarlar.				

Tablo 5.1: Form Sınıfının Temel Özellikleri

Tablo 5.2: Form Sınıfının Temel Metotları

Activate()	Form aktif hâle gelir.
Close()	Formu kapatır. Form, başlangıç formu ise uygulamayı sonlandırır.
CenterToScreen()	Formu ekranın merkezine yerleştirir.
Show()	Formu gösterir.
ShowDialog()	Formu diyalog kutusu olarak gösterir. Gösterilen form kapatılmadan diğer forma geçiş yapılamaz.
Hide()	Formu gizler.

Tablo 5.3: Form Sınıfının Temel Olayları

Load Form açılırken gerçekleşir.	
Click Form üzerine tıklandığı zaman gerçekleşir.	
FormClosing	Form kapanmadan hemen önce gerçekleşir.
FormClosed Form kapandıktan sonra gerçekleşir.	

Uygulama-1

Form Sınıfı

Bu uygulamada Form sınıfı için bazı olayların ve özelliklerin kullanılması gösterilecektir.

Adım 1: Yeni bir proje oluşturarak FormA ve FormB isminde iki form hazırlayınız.



http://kitap.eba.gov.tr/ KodSor.php?KOD=21081



Adım 2: FormA için Görsel 5.1'deki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖶 FormA	- 🗆	×	
Accept	btnAccept		
Cancel	btnCancel		
ControlBox	btnControlBox		
CenterToScreen	btnCenter		
Show	btnShow		
ShowDialog	btn Show Dialog		

Görsel 5.1: Form sınıfı uygulaması birinci form tasarımı

Adım 3: FormB için Görsel 5.2'deki form tasarımını yapınız ve kontrollere name değerlerini veriniz.



Görsel 5.2: Form sınıfı uygulaması ikinci form tasarımı

Adım 4: Birinci formdaki Accept ve Cancel butonları Click olaylarına aşağıdaki kodlamaları yapınız. Kodlamaları yaptıktan sonra Properties penceresinden formun AcceptButton ve CancelButton özelliğine ilgili butonları atayınız.



Adım 5: ControlBox butonu Click olayına formun sağ üst köşesindeki butonların gösterilmesi veya gizlenmesi işlemini gerçekleştiren kodlamayı yapınız.

```
private void btnControlBox_Click(object sender, EventArgs e)
{
    if (this.ControlBox == true)
        this.ControlBox = false;
    else
        this.ControlBox = true;
}
```

FORM UYGULAMALARI

ÖĞRENME BİRİMİ

Adım 6: CenterToScreen butonu Click olayına formu ekranın merkezine yerleştirme işlemini gerçekleştiren kodlamayı yapınız.

```
private void btnCenter_Click(object sender, EventArgs e)
{
    this.CenterToScreen();
}
```

Adım 7: Show butonu Click olayına oluşturulan ikinci formun gösterilmesini sağlayan kodlamayı yapınız.

```
private void btnShow_Click(object sender, EventArgs e)
{
    FormB formB = new FormB();
    formB.Show();
}
```

Adım 8: ShowDialog butonu Click olayına oluşturulan ikinci formun diyalog penceresi olarak gösterilmesini sağlayan kodlamayı yapınız.

```
private void btnShowDialog_Click(object sender, EventArgs e)
{
   FormB formB = new FormB();
   formB.ShowDialog();
}
```

Adım 9: İkinci formun Load olayına form yüklendiğinde çalışan kodlamaları yapınız.

```
private void FormB_Load(object sender, EventArgs e)
{
    MessageBox.Show("FormB yükleniyor.»);
}
```

Adım 10: İkinci formun FormClosing olayına form kapanırken çalışan kodlamaları yapınız.

```
private void FormB_FormClosing(object sender, FormClosingEventArgs e)
{
    MessageBox.Show("FormB kapaniyor.");
}
```

Adım 11: İkinci formun FormClosed olayına form kapandıktan sonra çalışan kodlamaları yapınız.

```
private void FormB_FormClosed(object sender, FormClosedEventArgs e)
{
    MessageBox.Show("FormB kapandı.");
}
```





FORM UYGULAMALARI



Adım 12: İkinci form içindeki Hide butonu Click olayına formun gizlenmesini sağlayan kodlamaları ya-

pınız.

```
private void btnHide_Click(object sender, EventArgs e)
{
    this.Hide();
}
```

Adım 13: İkinci form içindeki Close butonu Click olayına formun kapanmasını sağlayan kodlamaları yapınız.

```
private void btnClose_Click(object sender, EventArgs e)
{
   this.Close();
}
```

Adım 14: İkinci formun Click olayına form üzerinde herhangi bir yere fare ile tıklandığında çalışan kodlamaları yapınız.

```
private void FormB_Click(object sender, EventArgs e)
{
    MessageBox.Show("FormB click olayı çalışıyor.");
}
```

5.1.2. Kontrol Sınıfı

Kontrol sınıfı, System.Windows.Forms isim uzayında yer alan bir sınıftır. Form içine eklenen Button, TextBox, ListBox gibi görsel Windows kontrollerinin temel sınıfıdır. Aşağıda Tablo 5.4 ve Tablo 5.5'te kontrol sınıfında kullanılan temel özellikler ve olaylar verilmiştir.

Tablo 5.4: Kontrollerin Temel Özellikleri

BackColor ForeColor Font BackgroundImage Cursor	Kullanıcı arabirim görünüm ayarlarıdır.
Top Left Bottom Right Bounds Height Width	Kontrolün boyutlarını ve konumunu belirler.
Enabled Visible	Kontrolün durumunu belirler.
TabIndex TabStop	Kontrolün sekme sırasını belirler.
Opacity	Kontrolün şeffaflığını belirler (0.0 Şeffaf -1.0 Mat).
Text	Kontrolde metin gösterilmesini sağlar.





Tablo 5.5: Kontrollerin Temel Olayları

Click DoubleClick MouseEnter MouseLeave MouseDown MouseUp MouseMove MouseHover MouseWheel	Fare ile etkileşimi gerçekleştirir.
KeyPress KeyUp KeyDown	Klavye ile etkileşimi gerçekleştirir.
DragDrop DragEnter DragLeave DragOver	Sürükle bırak olaylarını gerçekleştirir.



Uygulama-2 Form Uygulamaları

Bazen form özelliklerine değer verilmesinde veya kontrollerin oluşturulmasında görsel kullanılarak değil, çalışma anında dinamik olarak kodlarla oluşturma ihtiyacı gerekebilir. Bu uygulamada birer adet Text-Box, Label ve Button kontrolünü form içine görsel olarak Toolbox penceresinde eklemek yerine çalışma anında dinamik olarak ekleme işlemi yapılacaktır. Dinamik olarak eklenen kontrole özellik değerleri verme ve olay metotları oluşturma gibi işlemler gerçekleştirilecektir.

Adım 1: Yeni bir form uygulama projesi oluşturunuz. Projedeki formu kod görünümünde açınız.

Adım 2: Form içine eklenecek kontrolleri tanımlamak için Form sınıfının başlangıcına aşağıdaki kodlamaları yapınız.

```
public partial class Form1 : Form
{
    Label lblAdSoyad = new Label();
    TextBox txtAdSoyad = new TextBox();
```

Button btnGiris = new Button();

Adım 3: Form Load olayına form özelliklerini değiştirmek için gereken kodlamaları yazınız.

```
private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "Form Sinifi Uygulama 2";
    this.BackColor = Color.Green;
    this.ForeColor = Color.Black;
    this.Size = new Size(300, 150);
}
```





FORM UYGULAMALARI



Adım 4: Forma eklenecek Label kontrolünün özellikleri için Form Load olayına kodları yazınız.

. .

lblAdSoyad.Text = "Adınız"; lblAdSoyad.Location = new Point(10, 10); lblAdSoyad.Size = new Size(65, 15); lblAdSoyad.ForeColor = Color.White;

Adım 5: Forma eklenecek TextBox kontrolünün özellikleri için Form Load olayına kodları yazınız.

txtAdSoyad.Location = new Point(75, 10); txtAdSoyad.Size = new Size(150, 15); txtAdSoyad.TabIndex = 1;

Adım 6: Forma eklenecek Button kontrolünün özellikleri için Form Load olayına kodları yazınız.

btnGiris.Text = "Tıkla"; btnGiris.Location = new Point(100, 60); btnGiris.Size = new Size(100, 30); btnGiris.ForeColor = Color.White; btnGiris.BackColor = Color.Black;

btnGiris.TabIndex = 2;

Adım 7: Buton için tıklama olayı ve TextBox için klavyeden bir tuşa basma olayı metotları oluşturulacaktır. Bu işlem için kontrole atanan olaydan sonra += operatörü ile olay metotlarının isimlerini vererek Form Load olayı içine ekleyiniz (+= operatöründen sonra tab tuşuna basıldığında metot otomatik olarak oluşturulacaktır.).

txtAdSoyad.KeyPress += TxtAdSoyad_KeyPress;

btnGiris.Click += BtnGiris_Click;

Adım 8: Dinamik oluşturulan, özellikleri verilen ve olay metotları ataması yapılan kontrollerin forma eklenmesi için Form Load olayına kodlamaları yapınız.

this.Controls.Add(lblAdSoyad);

this.Controls.Add(txtAdSoyad);

this.Controls.Add(btnGiris);

Adım 9: TextBox için oluşturulan Keypress olayında klavyeden girilen her harfi büyük harfe çeviren kodlamaları yapınız.

```
private void TxtAdSoyad_KeyPress(object sender, KeyPressEventArgs e)
{
    e.KeyChar = Char.ToUpper(e.KeyChar);
}
```



10, SINIF



Adım 10: Buton için oluşturulan Click olayında TextBox içine girilen bilgileri mesaj kutusunda gösteren kodlamaları yapınız.

private void BtnGiris_Click(object sender, EventArgs e)
{
MessageBox.Show("Merhaba " + txtAdSoyad.Text);
}

Sıra Sizde

Kodlamalarda neden this ifadesi kullanılmıştır? Açıklayınız.

5.1.3. Konteyner Kontrolleri

Bu kontroller, diğer kontroller için özelleştirilebilir konteyner görevi gören özel kontrollerdir. Form uygulaması geliştirilirken eklenen kontrollerin fazlalığı, formun görüntüsünü karmaşık hâle getirebilmektedir. Formun karmaşıklığını önlemek ve birbirleriyle ilişkili kontrolleri bir arada tutmak için konteyner kontroller kullanılır. Bunlar; Toolbox penceresinde Containers sekmesi içinde bulunan Panel, Groupbox, TabControl, FlowLayoutPanel kontrolleridir. Örneğin; öğrenci ve veli bilgileri kaydedilen bir arabirim formu tasarlandığında öğrenci bilgilerine ait kontroller bir konteyner içinde, veli bilgilerine ait kontroller ise başka bir konteyner içinde tutulabilmektedir.

Konteyner kontrolünün belirli özellikleri değiştirildiğinde içinde bulunan kontrollerin özellikleri de bu durumdan etkilenmektedir. Konteyner kontrolün **Enable** veya **Visible** özellikleri değiştirildiğinde konteyner içindeki tüm kontroller aynı şekilde etkilenmektedir.

• **GroupBox Kontrolü:** Formu bölümlere ayırarak ilişkili kontrolleri bir arada tutmak için kullanılan bir konteyner kontrolüdür.

S Uygulama-1

Bu uygulamada form üzerindeki ilişkili kontrolleri GroupBox içine alarak basit bir bilgisayar fiyat hesaplama uygulaması yapılacaktır.

Adım 1: Görsel 5.3'teki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🛃 GroupBox Kontrolü Uygulama 1 🛛 🚽 🦳 🗙					
işlemciler Intel Core i7 Intel Core i5 rbCpul5 Intel Core i3 rbCpul3 AMD Ryzen 5 rbCpuR5 AMD Ryzen 3 rbCpuR3 	Ram Bellek 16 GB 8 GB 4 GB	rbRam16 rbRam8 rbRam4	Sabit Disk () 1 TB () 500 GB () 320 GB	rbHdd1000 rbHdd500 rbHdd320	
Ek Donanımlar DVD RW cbDvd Web Cam cbWebCam Wireless Kart cbWifi			Hesapla btnHesapla		

Görsel 5.3: GroupBox uygulaması form tasarımı-1



```
FORM UYGULAMALARI
```

ÖĞRENME BİRİMİ



```
private void btnHesapla_Click(object sender, EventArgs e)
{
 decimal tabanFiyat = 500;
 // İşlemci fiyat hesaplaması
 decimal cpuFiyat = 0;
 if (rbCpuI7.Checked)
    cpuFiyat = 300;
 else if (rbCpuI5.Checked)
    cpuFiyat = 200;
 else if (rbCpuI3.Checked)
    cpuFiyat = 100;
 else if (rbCpuR5.Checked)
    cpuFiyat = 250;
 else if (rbCpuR3.Checked)
    cpuFiyat = 150;
 tabanFiyat += cpuFiyat;
 // Ram bellek fiyat hesaplaması
 decimal ramFiyat = 0;
 if (rbRam16.Checked)
    ramFiyat = 125;
 else if (rbRam8.Checked)
    ramFiyat = 75;
 else if (rbRam4.Checked)
    ramFiyat = 45;
 tabanFiyat += ramFiyat;
 MessageBox.Show(string.Format("Toplam Fiyat ={0:C}", tabanFiyat));
```

Sıra Sizde

GroupBox Uygulama 1'deki diğer donanım özellikleri için fiyat hesaplamasını yapınız.

 Panel Kontrolü: Bu kontrol, GroupBox gibi diğer kontrolleri barındıran formun alt bölümüdür. GroupBox'tan farklı olarak bu kontrolde kaydırma çubukları bulunur. AutoScroll özelliği true yapıldığında sınırları dışındaki alanlar için yatay veya dikey kaydırma çubukları görünür hâle gelmektedir.

> Uygulama-2

Bu uygulamada kod editörü içindeki Toolbox penceresinde bulunan WebBrowser ve Panel kontrollerini ekleyerek Internet Explorer'a ait özellikleri kullanan web tarayıcı uygulaması yapılacaktır.

10, SINIF



Adım 1: Yeni bir Windows form uygulaması oluşturunuz. Projenizin formuna üç adet Panel kontrolü ekleyiniz. Panellerin Dock özelliğini Görsel 5.4'teki gibi Top, Fill ve Right olarak düzenleyiniz.

🖳 Panel Kontrolü Uygulama 2	- • •
	000000000788888888888888888888888888888

Görsel 5.4: Panel kontrollerinin form üzerinde yerleşimi

Adım 2: Üst panel için Görsel 5.5'teki tasarımı yapınız.

Adres		Git	Ana Sayfa	Geri	İleri	Yenile
	txt Adres	btnGit	btn Ana Sayfa	btnGeri	btnlleri	btn Yenile

Görsel 5.5: Web tarayıcı gezinti bölümü panel tasarımı

Adım 3: Toolbox penceresinden WebBrowser kontrolünü ortadaki panel içine sürükleyerek bırakınız. Properties penceresinden ScriptErrorsSuppressed özelliğini true yapınız.

Adım 4: Toolbox penceresinden ListBox kontrolünü sağdaki panel içine sürükleyerek bırakınız. ListBox kontrolünün Dock özelliğini Fill yapınız.

Adım 5: Üst panelde bulunan Git ve Ana Sayfa butonlarının Click olaylarına aşağıdaki kodlamaları yapınız.

```
private void btnGit_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate(txtAdres.Text);
}
private void btnAnaSayfa_Click(object sender, EventArgs e)
{
    webBrowser1.GoHome();
}
```

Adım 6: Gezilen sayfaların listesini ListBox kontrolü içinde göstermek için WebBrowser kontrolünün Navigated olayına aşağıdaki kodlamaları yapınız.

```
private void webBrowser1_Navigated(object sender, WebBrowserNavigatedEventArgs e)
{
    listBox1.Items.Add(webBrowser1.Url);
}
```





Sıra Sizde

1. Geri, İleri ve Yenile butonlarına tıklandığında WebBrowser kontrolünün ziyaret edilen siteler arasında gezinti yapmasını ve mevcut siteyi yenilemesini sağlayan kodlamaları yapınız.

2. Üstteki panel içine Geçmiş adında bir buton ekleyerek ListBox kontrolünün bulunduğu paneli gösterip gizleme işlemini gerçekleştiren kodlamaları yapınız.

• **TabControl Kontrolü:** Bu kontrol, gruplar hâlindeki kontrolleri sekme sayfalarında göstermek için kullanılır. TabControl sekmelerine tıklanarak sekme sayfaları arasında geçiş yapılabilmektedir.

Uygulama-3

Bu uygulamada TabControl kullanılarak müşteri bilgileri girilecek ve verilen siparişlere göre hesap oluşturulacaktır.

Adım 1: Yeni bir form uygulama projesi oluşturunuz. Projenizin formuna TabControl ekleyiniz.

Adım 2: TabControl nesnesini seçtikten sonra Properties penceresinden TabPages özelliğine tıklayarak açılan pencereden üç adet sekme sayfası oluşturunuz. Oluşturulan sekme sayfalarının Text özelliğine sırasıyla Müşteri, Sipariş ve Hesap değerlerini atayınız.

Adım 3: İlk sekme sayfası için Görsel 5.6'daki tasarımı yapınız.

🖳 TabCont	rol Uygulama 3	—		×				
Müşteri Sipariş Hesap								
Ad Soyad			txtAdS	oyad				
Telefon		txtTele		fon				
Adres								
			txtAdre	s				

Görsel 5.6: TabControl sekme sayfası tasarımı

Adım 4: İkinci sekme sayfası için Görsel 5.7'deki tasarımı yapınız.

🖳 TabControl	Uygulama	3 —	\times
Müşteri Sipariş	Hesap		
Çorba	0	nCorba	
Salata	0	nSalata	
Ana Yemek	0	nYemek	
Tatlı	0	nTatli	

Görsel 5.7: TabControl sekme sayfası tasarımı





Adım 5: Üçüncü sekme sayfası için txtBilgi name değerine sahip bir TextBox ekleyerek Dock özelliğini Fill olarak değiştiriniz.

Adım 6: TabControl nesnesinin SelectedIndexChanged olayına üçüncü sekme sayfası açıldığında diğer sekme sayfalarındaki bilgileri kullanarak hesaplayan kodlamaları yapınız.

```
private void tabControl1 SelectedIndexChanged(object sender, EventArgs e)
{
  if(tabControl1.SelectedIndex==2)
  {
    txtBilgi.Text = "";
    txtBilgi.Text += txtAdSoyad.Text + "\r\n";
    txtBilgi.Text += txtTelefon.Text + "\r\n";
     txtBilgi.Text += txtAdres.Text + "\r\n";
    decimal hesap = 0;
    if (nCorba.Value > 0)
     {
       hesap += nCorba.Value*12;//Adet fiyati 12 lira.
       txtBilgi.Text += string.Format("Corba \{0:C\}",nCorba.Value * 12) + "\r\n";
     }
    txtBilgi.Text += "-----";
    txtBilgi.Text += string.Format("Toplam {0:C}",hesap);
  }
```

Sıra Sizde

Müşterilerin diğer siparişleri için hesaplama kodlamalarını yapınız.

5.2. MENÜLER

Menüler, bilgisayar uygulamalarında benzer veya ilgili komutları gruplandırmak için kullanılır. Menüler, uygulamanın daha kolay kullanılmasına olanak sağlar. Bu bölümde form içine menü ekleme, menü elemanlarına isim verme, menü elemanlarına tıklandığında ilgili kodları çalıştırma gibi işlemler açıklanacaktır. Bir Windows form uygulamasına menü eklemek için Görsel 5.8'de gösterilen Toolbox penceresinde Menus & Toolbars sekmesinde bulunan menü kontrolleri kullanılır.



Görsel 5.8: Menüler ve araç çubukları







Uvgulama-1

5.2.1. MenuStrip Kontrolü

MenuStrip kontrolü, menü araçlarının içinde en çok kullanılan kontroldür. Bu kontrol; birçok uygulamada karşılaşılan, genelde uygulamanın üst kısmında yer alan menüler ve alt menülerden oluşur. MenuStrip kontrolü, form tasarım görünümünde iken Toolbox penceresinden çift tıklanarak ya da form üzerine sürüklenerek forma eklenebilir.

MenuStrip kontrolü form içine eklendiğinde menülerin tasarlanacağı alan formun üst kısmında belirir. Bu alanda menüler görsel olarak tasarlanabileceği gibi Properties içinde Items özelliğine tıklanarak açılan "Items Collection Editor" penceresinden de tasarlanabilir.

Menüleri tasarım görünümünde oluşturmak için Görsel 5.9'da gösterilen "Type Here" yazan kısımlara bir kez tıklandığında menü içeriklerinin oluşturulmasını sağlayan kutular belirir. Bu kutulardan menü elemanı, açılır kutu, ayraç veya yazı eklenebilmektedir.

Görsel 5.9: MenuStrip tasarım alanı ve nesnesi

Bu uygulamada form içine MenuStrip kontrolü ekleyerek menü elemanlarını oluşturma işlemleri yapılacaktır.

Adım 1: Form üzerine Toolbox içinden MenuStrip kontrolü ekleyiniz. MenuStrip kontrolünün name değerini "AnaMenu" olarak değiştiriniz.

Adım 2: MenuStrip kontrolü Items özelliğine tıklayarak açılan "Items Collection Editor" penceresinden Görsel 5.10'da olduğu gibi üç tane MenuItem ekleyiniz. Eklenen menü elemanlarının Text özelliklerini sırasıyla Dosya, Düzen, Çıkış ve Name özelliklerini de menuDosya, menuDuzen, menuCikis olarak değiştiriniz.

Items Collection Editor					?	\times
Select item and add to list below:		Too	lStripMenultem too	lStripMenultem3		
Members:			Text	toolStripMenuItem	3	^
E AnaMenu	+		TextAlign	MiddleCenter		
E toolStripMenultem1	-		TextDirection	Horizontal		
E toolStripMenultem2	+		TextImageRelation	ImageBeforeText		
toolStripMenultem3	\sim	~	Behavior			
	\sim		AutoSize	True		
			AutoToolTip	False		
			CheckOnClick	False		
			DoubleClickEnabled	False		
			Enabled	True		
			ToolTipText			
			Visible	True		
		~	Data			
		>	(ApplicationSettings)			
			DropDown	(none)		
			DropDownItems	(Collection)		~
				ОК	Cancel	

Görsel 5.10: Items Collection Editor penceresi





Not: "Items Collection Editor" kullanılarak hazırlanan menü tasarımı, menü alanı içindeki "Type Here" kutuları seçilerek de yapılabilir.

Menü elemanlarına açılır menü ekleyerek benzer görevleri yerine getiren alt menüler oluşturulabilir. Örneğin; Dosya menüsünde dosya açma, kaydetme, yazıcıya gönderme gibi görevleri yerine getirecek işlemler için açılır menü kullanılır.

Menü elemanlarının altında açılır menü oluşturma, görsel olarak menü elemanlarının altında beliren kutularla yapılabileceği gibi menü elemanı seçildikten sonra Properties penceresi içinden DropDownItems özelliği kullanılarak da yapılabilir.

Uygulama-2

Açılır Menüler

Bu uygulamada menü elemanlarına tıklandığında benzer görevleri yerine getirecek olan açılır menü oluşturulacaktır. Oluşturulan açılır menülerin Text ve Name değerleri Görsel 5.11'de verilmiştir.

Dosya	(menuDosya)	Düzen	(menuDuzen)	Çıkış (menuCikis)
Yeni	(menuYeni)	Kes	(menuKes)	
Aç	(menuAc)	Kopyala	(menuKopya)	
Kayde	t (menuKaydet)	Yapıştır	(menuYapistir)	
Yazdır	(menuYazdir)			

Görsel 5.11: Açılır menüler

Adım 1: Dosya menü elemanı için açılır menüleri görsel tasarım alanındaki kutuları kullanarak oluşturunuz.

Adım 2: Düzen menü elemanı için açılır menüleri Properties içinde DropDownItems özelliğini kullanarak oluşturunuz.

• Menü Elemanlara Kısayol Tuşlarının Atanması

Kısayol tuşları, menü elemanlarına tıklanmadan klavye üzerindeki tuş birleşimleri ile menü elemanlarına erişim sağlar. Yaygın olarak kullanılan tuş birleşimlerinden bazıları Tablo 5.6'da verilmiştir.

Ctrl+N	Yeni Dosya	Ctrl+C	Kopyala
Ctrl+O	Dosya Açma	Ctrl+X	Kes
Ctrl+S	Kaydetme	Ctrl+V	Yapıştır
Ctrl+P	Yazdır	Alt+F4	Programı Kapat

Tablo 5.6: Yaygın Kullanılan Tuş Birleşimleri

Menü elemanlarına Properties penceresinden ShortcutKeys özelliğiyle Görsel 5.12'de olduğu gibi kısayol ataması gerçekleştirilebilir.

	ShortcutKeyDispl	ayString				
	ShortcutKeys		None		~	~
	ShowShortcutK	Modifiers:				
+	Size					
	Tag	Ctrl	Shift	Alt		
	Tevt	Key:				
S	hortcutKeys					
TI	he shortcut key a			~	Reset	

Görsel 5.12: Menülere kısayol ataması







• Pasif Menü Elemanları

Bir menü elemanı soluk renkte pasif hâle Properties penceresinden Enable özelliği false yapılarak getirilir. Örneğin bir metin editörü uygulamasında yapıştırma işlemi, kopyalama veya kesme işlemleri yapılana kadar pasif hâldedir. Kopyalama veya kesme işlemi yapılırsa Yapıştır menü elemanı aktif hâle gelir.

Uygulama-3	 	 	

Editör Uygulaması

Uygulama 2'deki menü tasarımı kullanılarak form içine eklenen RichTextBox kontrolü ile metinleri kesme, kopyalama ve yapıştırma işlemlerini gerçekleştiren bir editör uygulaması yapılacaktır. **Adım 1:** Form içine Toolbox penceresinde RichTextBox kontrolü ekleyiniz. Görsel 5.13'teki gibi

RichTextBox kontrolünün formun tamamını kaplaması için Properties penceresinde Dock özelliğini Fill olarak değiştiriniz.

🖳 Menu	🖳 MenuStrip Uygulama 2				_	\times	
Dosya	Düzen	Çıkış	i				
merhaba	Ke	s	Ctrl+X				
	Ko	pyala	Ctrl+C				
	Ya	pıştır	Ctrl+V				

Görsel 5.13: Editör uygulaması menü görünümü

Adım 2: Eklenen RichTextBox kontrolünün name değerini txtEditor olarak değiştiriniz. RichTextBox içinden seçilen metnin kesme, kopyalama ve yapıştırma işlemleri için menü elemanları Click olaylarına kodlamaları yapınız.



Yapıştır menü elemanını pasif hâle getiriniz. Kesme veya kopyalama işlemi gerçekleştirildiyse Yapıştır menü elemanını tekrar aktif hâle getirecek kodlamaları yapınız.





http://kitap.eba.gov.tr/ KodSor.php?KOD=21082



5.2.2. ContextMenuStrip Kontrolü

Birçok uygulamada işlemlerin daha hızlı gerçekleştirilebilmesi için sağ tıklama ile açılır menüler kullanılır. Bu menüleri oluşturmak için ContextMenuStrip kontrolü kullanılır. ContextMenuStrip tasarımı ve menü elemanlarının oluşturulması, MenuStrip kontrolüne benzer. Toolbox içinden Menus & Toolbars sekmesindeki ContextMenuStrip kontrolü çift tıklanarak veya sürüklenerek form içine eklenir. Menüler oluşturulduktan sonra bu menüler formla veya form içindeki herhangi bir kontrolle ilişkilendirilir. Uygulama çalıştırıldığında Mirilen kontrol üzerindeyken farenin sağ tuşuna basıldığında menü görünür hâle gelmektedir. **Uygulama-1**

Editör uygulamasında oluşturulan form üzerine ContextMenuStrip kontrolü ekleyerek seçilen metin üzerinde sağ tıklandığında açılır menüleri gösterme işlemi yapılacaktır. Açılır menüden kesme, kopyalama ve yapıştırma işlemleri gerçekleştirilecektir.

Adım 1: Form içine Toolbox penceresinden Menus & Toolbars sekmesindeki ContextMenuStrip kontrolünü ekleyiniz. ContextMenuStrip kontrolünün name değerini cmDuzen olarak değiştiriniz. Açılır menü tasarımını Görsel 5.14'te olduğu gibi yapınız.

🖳 ContextMenuStrip Uygulama 1	
Dosya Düzen Çıkış	
ContextMenuStrip	
Kes	
Kopyala	
Yapıştır	
Type Here	
1	

Görsel 5.14: ContextMenuStrip kontrolü menü tasarımı

Adım 2: ContextMenuStrip içindeki menü elemanları Click olayına Uygulama 3'teki düzen menüsü için yapılan olayları seçiniz (Burada seçim işleminin yapılması kod tekrarını önlemek içindir.).

Adım 3: ContextMenuStrip menülerini RichTextBox nesnesi ile ilişkilendirmek için RichTextBox nesnesini seçtikten sonra Properties penceresinden ContextMenuStrip özelliğini tıklayarak açılır menüyü seçiniz.

5.3. İLETİŞİM KUTULARI (DIALOG BOXES)

Dialog Boxes, kullanıcı ile uygulama arasında iletişimi sağlayan pencerelerdir. İletişim kutuları genellikle kullanıcıya bir komutun nasıl işleneceğini ifade eder veya kullanıcıdan bir sorunun cevabı istenildiğinde kullanılır.

İletişim kutuları yeniden boyutlandırılamaz. İletişim kutuları; kullanıcıya bilgi, hata veya uyarı mesajı göndermede, dosya açma veya kaydetme esnasında dosya konumunu belirlemede, belgeyi yazıcıya gönderme esnasında yazdırma ayarlarında, editör uygulamalarında yazının fontunu veya rengini değiştirmede vb. açılan pencerelerdir.









En çok kullanılan iletişim kutuları şunlardır:

- MessageBox
- OpenDialog
- SaveDialog
- FontDialog
- ColorDialog
- PrintDialog

5.3.1. Mesaj İletişim Kutusu (MessageBox)

MessageBox kullanıcıya bir mesaj vermek amacıyla kullanılır. **Show()** metodu ile parantez içine yazılan mesajları gösterir. Ayrıca overload yöntemi ile farklı kullanım şekilleri de bulunmaktadır. Aşağıda MessageBox'ın temel kullanımı verilmiştir.

MessageBox.Show("Merhaba Dünya");

İletişim kutusuna başlık eklemek için mesaj bilgisi verildikten sonra virgül bırakılarak başlık bilgisi yazılır.

MessageBox.Show("Merhaba Dünya","Bilgi Mesajı");

İletişim kutusunda standart olarak Tamam butonu bulunur. Tamam butonu dışında buton eklemek istenildiğinde MessageBox butonları kullanılır. MessageBox butonları şunlardır:

- MessageBoxButtons.OK Tamam
- MessageBoxButtons.AbortRetryIgnore Durdur, Yeniden Dene, Yoksay
- MessageBoxButtons.OKCancel Tamam, İptal
- MessageBoxButtons.RetryCancel Yeniden Dene, İptal
- MessageBoxButtons.YesNo Evet, Hayır
- MessageBoxButtons.YesNoCancel Evet, Hayır, İptal

MessageBox.Show("Merhaba Dünya","Bilgi Mesajı", MessageBoxButtons.YesNoCancel);

İletişim kutusu içinde ikonlar kullanılabilir. Bu ikonlar, verilmek istenen mesajın içeriğiyle ilişkili olmalıdır. MessageBox ikonları şunlardır:

- MessageBoxIcon.Error Hata ikonu
- MessageBoxIcon.Warning Uyarı ikonu
- MessageBoxIcon. Information Bilgilendirme ikonu
- MessageBoxIcon.Question Soru ikonu

MessageBox.Show("Merhaba Dünya","Bilgi Mesajı",MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);

Uygulama içinden kullanıcıya iletişim kutusu gösterildikten sonra kullanıcının hangi butona tıkladığının kontrolü için **DialogResult** kullanılır.

DialogResult cevap=MessageBox.Show("Bu dosyayı silmek istediğinize emin misiniz?", "Dosya Sil", MessageBoxButtons.YesNo,MessageBoxIcon.Warning); if(cevap==DialogResult.Yes) { MessageBox.Show("Dosya silindi."); else if(cevap==DialogResult.No) { MessageBox.Show("İşlem iptal edildi."); }






Sıra Sizde

Bir önceki bölümde yaptığınız editör uygulamasında Çıkış menü elemanına tıklandığında MessageBox iletişim kutusu gösterilerek "Çıkmak istediğinize emin misiniz?" mesajı verilecektir. Evet butonuna tıklandığında uygulamayı kapatma işlemini gerçekleştiren kodlamayı yapınız.

5.3.2. Dosya Kaydet İletişim Kutusu (SaveFileDialog)

SaveFileDialog, kullanıcının bilgisayarındaki klasörlere göz atmasını ve dosyayı kaydedebileceği konumu belirlemesini sağlayan iletişim kutusudur. SaveFileDialog dosya kaydetme işlemi yapmaz, sadece kaydedilecek dosyanın konumunun ve isminin belirlenmesini sağlar. Dosya iletişim kutusu, tasarım veya çalışma anında oluşturularak kullanılabilir. Tasarım ekranında iken Toolbox penceresi Dialogs sekmesinden SaveFileDialog form içine sürüklenerek veya çift tıklanarak eklenebilmektedir. Çalışma anında ise SaveFile-Dialog sınıfından bir nesne oluşturularak iletişim kutusunun özellikleri kontrol edilebilmektedir.

SaveFileDialog sfd = new SaveFileDialog();
sfd.SHOWDIALOG();

Filter özelliği, SaveFileDialog sınıfından oluşturulan nesne veya Toolbox penceresinde form içine eklenen diyalog ile iletişim kutusu kaydedilecek dosyanın uzantısını veya uzantılarını belirlemek için kullanılır.

sfd.Filter ="Text Dosyası |*.txt"; // Kaydedilecek dosyanın uzantısı txt olarak belirlenir. sfd.Filter = "Text Dosyası |*.txt| Word Dosyası |*.docx"; // Kaydedilecek dosyanın uzantısı txt veya docx olarak belirlenir.

SaveFileDialog penceresinde kaydedilecek dosyanın ismi, uzantısı ve konumu belirlendikten sonra Kaydet butonuna tıklandığını kontrol etmek için DialogResult kullanılır.

```
SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Text Dosyası |*.txt| Tüm Dosyalar |*.*";

DialogResult cevap = sfd.ShowDialog();

if(cevap==DialogResult.OK)

{

MessageBox.Show("Dosya kaydetme işlemi başlıyor.")
```

Uygulama-1 SaveFileDialog

Burada Editor uygulamasındaki RichTextBox kontrolünün içinde yer alan metinleri bir metin dosyası olarak kaydetme işlemi gerçekleştirilecektir.

Adım 1: Editör uygulamasında Dosya menüsündeki Kaydet menü elemanı Click olayına SaveFileDialog sınıfından oluşturulan nesne kullanılarak iletişim kutusu gösterilecektir. İletişim kutusundaki Kaydet butonuna tıklandığında RichTextBox içindeki bilgileri belirlenen konuma **SaveFile()** metodu ile kaydetme işlemini gerçekleştiren kodlamaları yapınız.

```
private void menuKaydet_Click(object sender, EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Filter = "Text Dosyası |*.txt| Tüm Dosyalar |*.*";
    DialogResult cevap = sfd.ShowDialog();
    if(cevap==DialogResult.OK)
    {
        txtEditor.SaveFile(sfd.FileName,RichTextBoxStreamType.PlainText);
    }
}
```





5.3.3. Dosya Aç İletişim Kutusu (OpenFileDialog)

OpenFileDialog, kullanıcıların kendi bilgisayarına veya ağdaki herhangi bir bilgisayarın klasörlerine göz atmasına ve açmak için bir dosya seçmesine olanak sağlayan iletişim kutusudur. OpenFileDialog sınıfından üretilen nesneyi veya Toolbox penceresi Dialogs sekmesinden eklenen kontrolü kullanarak seçilen dosyanın adı ve yolu elde edilir.

```
OpenFileDialog ofd = new OpenFileDialog();
ofd.ShowDialog();
```

OpenFileDialog, Filter özelliği ile iletişim kutusu içinde gösterilecek dosyaların uzantılarına göre filtreleme yaparak sadece belirlenen uzantıda olan dosyaların gösterilmesini sağlar.

ofd.Filter = "Text Dosyaları|*.txt"; // Uzantısı txt olanları gösterir. ofd.Filter = "Text Dosyaları|*.txt| Word Dosyaları|*.docx"; // Uzantısı txt veya docx olanları gösterir. ofd.Filter = "Text Dosyaları |*.txt| Word Dosyaları |*.docx| Tüm Dosyalar |*.*"; // Uzantısı txt, docx veya tüm dosyaları gösterir.

OpenFileDialog penceresinden dosya seçildikten sonra Dosya Aç butonuna tıklandığını kontrol etmek için DialogResult kullanılır.

```
DialogResult CEVAP= ofd.SHOWDIALOG();
if(cevap==DialogResult.OK)
{
    MessageBox.Show(ofd.FileName);
}
```



OpenFileDialog

Burada Editor uygulaması kullanılarak bir metin dosyasının içeriğini RichTextBox kontrolü içinde gösterme işlemi yapılacaktır.

Adım 1: Editör uygulamasında Dosya menüsündeki Dosya Aç menü elemanı Click olayına OpenFileDialog nesnesi kullanılarak txt uzantılı dosyaların seçilmesi sağlanacaktır. İletişim kutusundaki Dosya Aç butonuna tıklandığında seçilen dosyanın RichTextBox kontrolünün **LoadFile()** metodu ile dosya içeriğini gösteren kodlamaları yapınız.

```
private void menuAc_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "Text Dosyaları |*.txt";
    DialogResult cevap= ofd.ShowDialog();
    if(cevap==DialogResult.OK)
    {
        txtEditor.LoadFile(ofd.FileName,RichTextBoxStreamType.PlainText);
    }
}
```









5.3.4. Yazdırma İletişim Kutusu (PrintDialog)

PrintDialog; yazıcı seçme, kâğıt boyutlarını ayarlama veya yazdırmayı başlatma gibi işlemlerin yapılmasını sağlayan iletişim kutusudur. PrintDialog sınıfından bir nesne oluşturularak bu nesne ile iletişim kutusunun özellikleri kontrol edilmektedir.

```
PrintDialog pd = new PrintDialog();
```

pd.ShowDialog();



PrintDialog

Burada Editor uygulaması kullanılarak Dosya menüsündeki Yazdır menü elemanına tıklandığında iletişim kutusu açılacaktır. İletişim kutusundaki Tamam butonuna tıklandığında RichTextBox kontrolü içindeki metinleri yazdırma işlemi gerçekleştirilecektir.

Adım 1: PrintDialog sınıfından oluşturulan nesne iletişim kutusunu göstererek yazıcı seçimi ve ayarların yapılması sağlanmaktadır. Yazdırma işlemi için **PrintDocument** nesnesi kullanılacaktır. Tasarım görünümünde iken ToolBox penceresinden PrintDocument kontrolünü form içine ekleyiniz. PrintDocument kontrolünün name değerini belge olarak değiştiriniz.

Adım 2: Dosya menüsündeki Yazdır menü elemanı Click olayında PrintDialog nesnesi kullanılarak RichTextBox içindeki yazıların yazıcıya gönderilmesi işlemini yapan kodlamaları yazınız.

```
private void menuYazdir_Click(object sender, EventArgs e)
{
    PrintDialog pd = new PrintDialog();
    DialogResult cevap = pd.ShowDialog();
    if (cevap == DialogResult.OK)
    {
        belge.Print();
    }
}
```

Adım 3: Yazdırma işlemini başlatmak için form içine eklenen PrintDocument kontrolü **PrintPage** olayına aşağıdaki kodlamaları yazınız.

```
private void belge_PrintPage(object sender, PrintPageEventArgs e)
{
    e.Graphics.DrawString(txtEditor.Text, txtEditor.Font,
    Brushes.Black,new Point(100,100));
}
```

5.3.5. Yazı Tipi İletişim Kutusu (FontDialog)

FontDialog; yazı tipinin, büyüklüğünün ve renginin seçilmesine olanak sağlayan iletişim kutusudur. FontDialog sınıfından bir nesne oluşturularak bu nesne ile iletişim kutusunun özellikleri kontrol edilmektedir.



FontDialog

Editör uygulamasında FontDialog sınıfında üretilen bir nesne kullanılarak RichTextBox içinde seçilen yazının tipini ve büyüklüğünü değiştirme işlemi yapılacaktır.



FORM UYGULAMALARI ÖĞRENME BİRİMİ

Adım 1: Editör uygulaması menülerine Biçim adında menü ekleyiniz. Biçim menüsü içine de Yazı Tipi adında açılır menü elemanını ekleyiniz.

Adım 2: Yazı Tipi açılır menüsünün name değerini menuYaziTipi olarak veriniz. Yazı Tipi alt menüsü Click olayında FontDialog iletişim kutusunu göstererek belirlenen yazı tipini ve boyutunu RichTextBox kontrolüne uygulayan kodlamayı yapınız.

```
private void menuYaziTipi_Click(object sender, EventArgs e)
{
   FontDialog fd = new FontDialog();
   if (fd.ShowDialog() == DialogResult.OK)
   {
     txtEditor.Font = fd.Font;
   }
}
```

Sıra Sizde >>

1. Uygulamada ShowDialog metodu neden karşılaştırma ifadesi içinde kullanıldı? Açıklayınız.

2. Uygulamada RichTextBox içindeki metnin sadece seçim yapılan kısımlarının biçimini değiştiren kodlamayı yapınız.

5.3.6. Renk İletişim Kutusu (ColorDialog)

ColorDialog, renk paleti içinden bir renk seçilmesini sağlayan iletişim kutusudur.

Uygulama-1

ColorDialog

Editör uygulamasında ColorDialog sınıfında üretilen bir nesne kullanılarak iletişim kutusu içinden seçilen bir rengin RichTextBox kontrolüne uygulanması işlemi yapılacaktır.

Adım 1: Editör uygulaması Biçim menüsü içine Renk adında açılır menü ekleyiniz. Renk menü elemanının name değerini menuRenk olarak veriniz.

Adım 2: Renk menü elemanı Click olayında RichTextBox içinden seçilen metnin renginin ColorDialog iletişim kutusu ile değiştirilmesini sağlayan kodlamayı yapınız.

```
private void menuRenk_Click(object sender, EventArgs e)
{
    ColorDialog cd = new ColorDialog();
    if (cd.ShowDialog() == DialogResult.OK)
    {
        txtEditor.SelectionColor = cd.Color;
    }
}
```

Sıra Sizde >

ColorDialog iletişim kutusunun yaptığı görevi FontDialog iletişim kutusu da yapabilmektedir. FontDialog içinden renk seçme işlemi için gerekli düzenlemeleri yapınız.

5.4. VERİ DOĞRULAMA (VALIDATION)

Veri girişi ile işlem yapılan uygulamalarda önemli noktalardan biri de verinin doğru girilmesidir. Doğru veri girilmemesi veya eksik bilgi girişi yapılması durumunda uygulamada hatalar, aksaklıklar ortaya çıkacak veya uygulama çalışamaz hâle gelecektir.

Yazılımci, bir uygulamayı geliştirirken bu uygulamayı kullanacak kişilerin yapabileceği hataları öngörerek önlem almalıdır. Örneğin, öğrenci not girişleri yapılan uygulamada sınav notu 100 olan bir öğrenciye yanlışlıkla 1000 notu girilirse öğrencinin ortalaması çok yüksek çıkacaktır veya rakam yerine bir harf girilirse uygulama hata verecektir. Uygulama geliştirirken kullanıcıları bilgilendirerek ve hatalara karşı önlem alarak oluşabilecek problemlerin önüne geçilir.



5.4.1. İpucu (ToolTip)

ToolTip, form içindeki nesnelere ait bilgilerin görüntülendiği açıklama pencereleridir. Form içindeki bir nesne üzerine fare ile gelindiğinde açıklama bilgisinin görüntülenmesini ToolTip sağlar.

Uygulama-1			

ToolTip

Bu uygulama, **ToolTip** sınıfından üretilen nesne ile form içindeki kontrollerin üzerine gelindiğinde açıklama pencerelerinin gösterilmesidir.

Adım 1: Görsel 5.15'teki form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

🖳 Uygulama 1: ToolTip		_		×
Ad :	1	b t	d.Ad	
Soyad :		b	dSoyad	
Doğum Tarihi :		b	dDTarih	
Adres :		b	d Adres	
	Kaydet	b	tnKaydet	

Görsel 5.15: ToolTip uygulaması form tasarımı-1

Adım 2: Formun Load olayında ToolTip sınıfından bir nesne oluşturarak form içindeki kontrollerde açıklama pencerelerini gösteren kodlamayı yapınız.



5.4.2. Veri Girişi Doğrulama (Input Validation)

Veri doğrulama, uygulamanın her katmanında yapılabilmektedir. Yazılımcı tarafından belirlenen bazı ölçütlere göre kullanıcıların girdiği bilgilerin doğruluğu kontrol edilmelidir. Kullanıcının girdiği bilgiler ilk olarak form seviyesinden kontrol edilir. Form uygulamalarında veri girişleri en çok TextBox kontrolleri ile gerçekleşmektedir. Yazılımcı TextBox'a girilen verilerin doğruluğunu kullanıcının klavyeden girdiği her karakter esnasında sağlayabileceği gibi **Validating** olayı ile de kontrol edebilir. Girilen veriler uygun biçimde olmadığında uyarı mesajları verilir veya **ErrorProvider** nesnesi kullanılarak bilgilendirme işlemi gerçekleştirilir.







🔶 Uygulama-1

Veri Girişi Doğrulama

Burada form içinden veri girişi yapılan alanlara belirlenen ölçütlere göre doğru veri girişi yapılmadığı takdirde uyarı mesajı veren uygulama gerçekleştirilecektir.

Adım 1: Görsel 5.16'daki form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

🖳 Veri Doğrulama Uyg	ulama 1		_		\times
Kullanıcı Adı Şifre		Giriş	t	ttAd ttSifre tnGiris	

Görsel 5.16: Veri doğrulama uygulaması form tasarımı-1

Adım 2: Giriş butonu Click olayında kullanıcı adı ve şifre verileri girilmez ise kullanıcıyı mesaj ile uyaran kodlamaları yapınız.

private void btnGiris_Click(object sender, EventArgs e)
{
if (string.IsNullOrEmpty(txtAd.Text))
MessageBox.Snow("Kullanici adini giriniz.", "Uyari");
} if(string.IsNullOrEmpty(txtSifre.Text))
{
MessageBox.Show("Şifreyi giriniz.", "Uyarı");
}
}

Uygulama-2

Bu uygulamada uyarı mesajları MessageBox yerine ErrorProvider nesnesi kullanılarak verilecektir. Doğrulama kontrolü ise TextBox nesnelerinin Validating olayı içinde gerçekleştirilecektir.

Adım 1: Görsel 5.17'deki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖳 Veri Doğr	ulama Uygulama 2 🛛 🗆 🗙
Numara	txtNumara
Ad Soyad	txtAdSoyad
Ders Notu	txtDersNotu
	Kaydet btnKaydet

Görsel 5.17: Veri doğrulama uygulaması form tasarımı-2









Adım 2: Uygulamada ErrorProvider sınıfından üretilen nesneyi global olarak oluşturunuz.

ErrorProvider ep = new ErrorProvider();

Adım 3: Form içindeki txtNumara isimli TextBox kontrolüne sayısal bir değer girilmelidir. TextBox içine girilen değerin sayısal olup olmadığını Validating olayında kontrol eden kodlamaları yapınız.

```
private void txtNumara_Validating(object sender, CancelEventArgs e)
{
    if(int.TryParse(txtNumara.Text,out int sonuc))
    {
        ep.SetError(txtNumara, "");
    }
    else
    {
        e.Cancel = true;
        ep.SetError(txtNumara, "Numara girişi hatalı");
    }
}
```

Adım 4: txtAdSoyad isimli TextBox kontrolünün Validating olayında TextBox içine değer girilmemiş ise ErrorProvider nesnesi ile kullanıcıyı uyaran doğrulama kodlarını yazınız.

```
private void txtAdSoyad_Validating(object sender, CancelEventArgs e)
{
    if (txtAdSoyad.Text == "")
        {
            e.Cancel = true;
            ep.SetError(txtAdSoyad, "Adı ve soyadı giriniz.");
        }
        else
        {
            ep.SetError(txtAdSoyad, "");
        }
}
```

Adım 5: txtDersNotu isimli TextBox kontolünün Validating olayında TextBox içine girilen değer sayısal değilse veya sayısal olup 0 ile 100 arasında değilse ErrorProvider nesnesi ile kullanıcıyı uyaran doğrulama kodlarını yazınız.

```
private void txtDersNotu_Validating(object sender, CancelEventArgs e)
{
  int dersNotu;
  if(int.TryParse(txtDersNotu.Text,out dersNotu))
 {
     if(dersNotu<0 || dersNotu>100)
     {
        e.Cancel = true;
        ep.SetError(txtDersNotu, "0-100 arasında değer giriniz.");
     }
     else
     {
        ep.SetError(txtDersNotu, "");
     }
   }
   else
   ł
     e.Cancel = true;
     ep.SetError(txtDersNotu, "Sayısal değer giriniz.");
   }
```



Sıra Sizde

Uygulamada e.Cancel = true ifadesi neden kullanılmıştır? Düşüncelerinizi sınıf arkadaşlarınızla paylaşınız.

5.4.3. Veri Girişi Maskeleme (MaskedTextBox)

MaskedTextBox, belirlenen formatta veri girişinin yapılmasını sağlayan bir kontroldür. MaskedText-Box kontrolleri, form içinde kullanıcının gireceği verilerin doğrulanmasını sağlar. Örneğin, telefon numarası girişi yapılan bir alana uygun formatta veri girilmesi için MaskedTextBox kontrolü kullanılır.

Forma eklemek için tasarım ekranında iken Toolbox penceresinden MaskedTextBox kontrolü form içine sürüklenerek bırakılabilir. MaskedTextBox kontrolü eklendikten sonra giriş maskesi için Görsel 5.18'de gösterilen **Set Mask** linkine veya Properties penceresinden **Mask** özelliğine tıklanarak açılan diyalog kutusundan maske formatı belirlenebilir.



Görsel 5.18: Giriş maskesi linki

Görsel 5.19'da açılan diyalog kutusundan maske formatlarının listesi görülebilir ve istenilen bir format seçilerek giriş maskesi olarak kullanılabilir.

Input Mask				? ×
Select a predefin	ed mask descriptio	on from the list below or sel	ect Custom to define a custom	mask.
Mask Description	on	Data Format	Validating Type	
Numeric (5-digi	ts)	12345	Int32	
Phone number		(574) 555-0123	(none)	
Phone number r	no area code	555-0123	(none)	
Short date	Short date		DateTime	
Short date and t	ime (US)	12/11/2003 11:20	DateTime	
Social security n	umber	000-00-1234	(none)	
Time (European	Time (European/Military)		DateTime	
Time (US)		11:20	DateTime	
Zip Code		98052-6399	(none)	
<custom></custom>			(none)	
Mask:			Use Validat	ingType
Preview:				
			ОК	Cancel

Görsel 5.19: Giriş maskesi diyalog penceresi









Bazı durumlarda Input Mask diyalog kutusundaki formatlar dışında özel formatlar belirlemek için Mask alanına maskeleme karakterleri girilerek özel maskeler oluşturulabilir. Bu karakterler şunlardır:

- **0** Rakam (Girilmesi zorunludur.)
- 9 Rakam veya boşluk (Girilmesi isteğe bağlıdır.)
- # Rakam veya boşluk (Girilmesi isteğe bağlıdır. Maskede bu konum boşsa özellik de bir boşluk olarak işlenir. Artı ve eksi işaretlerine izin verilir.)
- L Harf (Girilmesi zorunludur. A-Z veya a-z arasındaki harflere izin verir.)
- ? Harf (Girilmesi isteğe bağlıdır. A-Z veya a-z arasındaki harflere izin verir.)
- & Karakter (Girilmesi zorunludur.)
- C Karakter (Girilmesi isteğe bağlıdır.)
- A Hem harf hem de rakam (Girilmesi zorunludur.)
- a Hem harf hem de rakam (Girilmesi isteğe bağlıdır.)
- Ondalık ayracı
- _ Binlik ayracı
- Saat ayracı
- 🖌 Tarih ayracı
- s Para birimi işareti
- Tüm karakterleri küçük harfe çevirir.
- Tüm karakterleri büyük harfe çevirir.

Uygulama-1

MaskedTextBox

Bu uygulamada MaskedTextBox kontrollerine Form Load olayı içinde farklı maskeler verilecektir. **Adım 1:** Görsel 5.20'deki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖳 MaskedTextB	ox Uygulama 1		
T.C. No		maskTC	
Telefon		maskTelefon	
Doğum Tarihi		maskDTarih	
Kart Numarası		mask Kart	
IP Adres		masklp	

Görsel 5.20: Giriş maskeleme uygulaması-1

Adım 2: Form Load olayında ilgili MaskedTexBox kontrollerine aşağıdaki kodlamaları yaparak maske formatlarını veriniz.





5.5. VERİ BAĞLAMA (DATA BINDING)

Veri bağlama, veri kaynağının form içindeki bir kontrole bağlanmasıdır. Veri bağlama işleminde form kontrolü ile kaynak arasında veri gönderme veya alma yapılabilir. Veri bağlama en çok veri tabanı içinde depolanan verilerle işlem yapmakta kullanılsa da kontrollerle, dizilerle veya koleksiyonlarla da veri bağlama işlemi yapılabilmektedir.

5.5.1. Basit Veri Bağlama (Simple Data Binding)

Basit veri bağlama, bir form kontrolüne tek veri bilgisinin bağlanmasıdır. Basit veri bağlama genellikle veri kümesi içindeki bir veriyi form içindeki kontrole aktarmak için kullanılır. Basit veri bağlama işlemlerinde Binding sınıfında üretilen nesne kullanılır. Bu nesne temelde üç parametre almaktadır. Bu parametreler sunlardır:

Binding binding=new Binding(string propertyName, object dataSource, string dataMember)

Binding sınıfındaki parametrelerin işlevleri aşağıda verilmiştir.

- propertyName: Bağlanacak olan nesnenin özelliğini belirtir.
- dataSource: Veri kaynağını belirtir.
- dataMember: Veri kaynağının hangi özelliğinin bağlanacağını belirtir.

Uygulama-1

Basit Veri Bağlama

Bu uygulamada form içindeki iki TextBox kontrolünden biri veri kaynağı olarak kullanılacak ve diğer TextBox kontrolüne veri bağlama işlemi gerçekleştirilecektir.

Adım 1: Görsel 5.21'deki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖳 Uygulama 1: Basit Veri Bağlama			_	×
Kaynak	[]	txtKaynak		
Hedef		txtHedef		

Görsel 5.21: Basit veri bağlama uygulaması-1

Adım 2: Formun Load olayına veri bağlama işlemleri için kullanılan kodlamaları yapınız.

private void Form1_Load(object sender, EventArgs e)
{
Binding bagla = new Binding("Text", txtKaynak, "Text");
txtHedef.DataBindings.Add(bagla);
}







	UYGULAMA	LARI					
ÖĞRENME BİRİMİ						• /	
Uygulama-2 Basit Veri Bag Bu uygulamada değer aktarılarak form Adım 1: Görse	ğlama a bir sınıf oluştu içinde kontrolleı el 5.22'deki form	rulacaktır. Sınıfı re veri bağlama tasarımını vap	tan üretilen r işlemi gerçe nız ve kontro	nesne kleştir bllere ı	ile sınıf ilecektir. name de	özelikler :äerlerir	rine
						genen	ii veriinz.
	🔡 Basit Veri Ba	ağlama Uygulamı	a 2	-		×	
	Numara		txtNumara				
	Ad Soyad		txtAdSoyad		Veri Bağla	а	
	Alan		txtAlan		btnBagla		

Görsel 5.22: Basit veri bağlama uygulaması-2

Adım 2: Veri kaynağı olarak kullanmak için aşağıda özellikleri verilen sınıfı oluşturunuz.

```
class Ogrenciler
{
    public int Numara { get; set; }
    public string AdSoyad { get; set; }
    public string Alan { get; set; }
}
```

Adım 3: Veri Bağla butonu Click olayında sınıf özelliklerine değer aktarıldıktan sonra TextBox kontrollerine veri bağlama işlemini gerçekleştiren kodlamaları yapınız.

```
private void btnBagla_Click(object sender, EventArgs e)
{
    Ogrenciler ogrenci = new Ogrenciler();
    ogrenci.Numara = 1111;
    ogrenci.AdSoyad = "Mehmet";
    ogrenci.Alan = "Bilişim Teknolojileri";
    txtNumara.DataBindings.Add("Text", ogrenci, "Numara");
    txtAdSoyad.DataBindings.Add("Text", ogrenci, "AdSoyad");
    txtAlan.DataBindings.Add("Text", ogrenci, "Alan");
}
```







5.5.2. Kompleks Veri Bağlama (Complex Data Binding)

Basit veri bağlama, bir kontrol nesnesine tek bir veri değerinin bağlanması işlemidir. Kompleks veri bağlama işlemi ise bir kontrol nesnesine birden çok veri değerinin bağlanması işlemidir. ListBox, ComboBox ve DataGridView kontrolleri, kompleks veri bağlanmasında en çok kullanılan kontrollerdir.

Kompleks bağlamada veri bağlanacak olan kontrolün **DataSource** özelliği ile veri kaynağı belirtilerek bağlama işlemi gerçekleştirilir.



Kompleks Veri Bağlama

Bu uygulamada dizideki verileri ListBox kontrolü içinde gösterme işlemi yapılacaktır. Diziler konusunda döngülerle gerçekleştirilen işlem, bu uygulamada veri bağlama yöntemi ile gerçekleştirilecektir.

Adım 1: Görsel 5.23'teki gibi form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖳 Kompleks Veri Bağlama Uygulama 1	_		\times
Veri Bağla btnVeriBagla	listeVe	riler	

Görsel 5.23: Kompleks veri bağlama uygulaması form tasarımı-1

Adım 2: Veri Bağla butonu Click olayında bir dizi oluşturarak dizinin değerlerini ListBox kontrolü içinde göstermek için veri bağlama işleminin kodlarını yazınız.



Uygulama-2

Kompleks Veri Bağlama

Bu uygulamada ArrayList koleksiyonu içine sınıf nesnesi eklenerek ListBox içinde gösterilecektir. **Adım 1:** Görsel 5.24'teki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🛃 Komple	ks Veri Bağlama Uygulama	_		×	
			lis	steVeri	
Kitap Adı	l	txtKitapAdi			
Kitap Yazan		txt Kitap Yazari			
Kitap Türü		txtKitapTuru			
	Kitap Ekle				
	btn Kitap Ekle			Listele	
			bt	tnListele	

Görsel 5.24: Kompleks veri bağlama uygulaması form tasarımı-2

10, SINIF





FORM UYGULAMALARI



{

}

Adım 2: Kitaplar adında bir sınıf oluşturunuz ve bu sınıfın özelliklerini belirleyen kodlamaları yapınız.

class Kitaplar

public string KitapAdi { get; set; }
public string KitapYazari { get; set; }
public string KitapTuru { get; set; }

Adım 3: Veri kaynağı olarak kullanılacak ArrayList koleksiyonunu global olarak oluşturunuz.

ArrayList kaynakVeri = new ArrayList();

Adım 4: Kitap Ekle butonu Click olayında Kitaplar sınıfından üretilen nesne kullanılarak sınıfın özelliklerine değer aktarımını ve nesneyi ArrayList koleksiyonuna ekleme işlemini yapınız.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    Kitaplar kitap = new Kitaplar();
    kitap.KitapAdi =txtKitapAdi.Text;
    kitap.KitapYazari = txtKitapYazari.Text;
    kitap.KitapTuru = txtKitapTuru.Text;
    kaynakVeri.Add(kitap);
}
```

Adım 5: Listele butonu Click olayında ListBox kontrolüne ArrayList nesnesini bağlayınız. ListBox kontrolüne bağlanan veriler içinden KitapAdi özelliğinin gösterilmesini sağlayan kodlamaları yapınız.

```
private void btnListele_Click(object sender, EventArgs e)
{
    listeVeri.DataSource=null;
    listeVeri.DataSource = kaynakVeri;
    listeVeri.DisplayMember = "KitapAdi";
}
```

Sıra Sizde

1. ListBox kontrolü içinde veri kaynağının diğer özelliklerinin gösterilmesini sağlayan kodlamaları yapınız.

2. Oluşturacağınız küçük gruplarla Listele butonu Click olayında DataSource özelliğine neden null değerinin aktarıldığını araştırınız. Elde ettiğiniz sonuçları diğer gruplarla paylaşınız.









Uygulama-3

Kompleks Veri Bağlama

Bu uygulamada ComboBox kontrolü ve List koleksiyonu kullanılarak veri bağlama işlemleri gerçekleştirilecektir. List koleksiyonuna plaka numarası ve şehir ismi özelliğine sahip sınıf nesnesi eklenerek ComboBox kontrolü içinde gösterme işlemi yapılacaktır.

Adım 1: Görsel 5.25'teki form tasarımını yapınız ve kontrollere name değerlerini veriniz.

🖶 Kompleks Veri Bağlama Uygulama 3			_		×
Plaka	txtPlaka		Şehirler	~	
Şehir	Şehir Ekle b	xtSehirAdi [cb Sehirler		

Görsel 5.25: Kompleks veri bağlama uygulaması form tasarımı-3

Adım 2: List koleksiyonuna eklenecek nesneler için sınıf oluşturunuz ve bu sınıfın özelliklerini belirleyen kodlamaları yapınız.

```
class Sehirler
{
    public string Plaka { get; set; }
    public string SehirAdi { get; set; }
}
```

Adım 3: Kullanılacak List koleksiyonunu global olarak oluşturunuz.

```
List<Sehirler> listSehirler = new List<Sehirler>();
```

Adım 4: Şehir Ekle butonu Click olayında Sehirler sınıfından üretilen nesne kullanılarak sınıfın özelliklerine değer aktarımı yapıldıktan sonra List koleksiyonuna ekleme işleminin kodlarını yazınız.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    Sehirler sehir = new Sehirler();
    sehir.Plaka = txtPlaka.Text;
    sehir.SehirAdi = txtSehirAdi.Text;
    listSehirler.Add(sehir);
    Bagla();
}
```





Adım 5: Bagla() ismindeki metot ile List koleksiyonunun ComboBox kontrolüne veri bağlama işlemini gerçekleştiriniz.



Uygulama-4

Kompleks Veri Bağlama

Bu uygulamada DataGridView kontrolü ve List koleksiyonu kullanılarak veri bağlama işlemleri gerçekleştirilecektir. List koleksiyonuna öğrenci numarası, adı ve sınav notu özelliklerine sahip sınıf nesnesi eklenerek DataGridView kontrolü içinde gösterme işlemi yapılacaktır.

Adım 1: Görsel 5.26'daki form tasarımını yapınız ve form içindeki kontrollere name değerlerini veriniz.

🔢 Kompleks Veri Bağlama Uygulama 4		-	×
Öğrenci Bilgileri Numara			
Ders Notu Ekle btnEkle			
	gridListe		

Görsel 5.26: Kompleks veri bağlama uygulaması form tasarımı-4

Adım 2: Form kod görünümüne geçerek Ogreciler isminde bir sınıfın özelliklerini oluşturan kodlamayı yapınız.

class	Ogrenciler
{	
put	plic int Numara { get; set; }
put	plic string AdSoyad { get; set; }
put	plic int DersNotu { get; set; }
3	
put	<pre>blic int DersNotu { get; set; }</pre>









Adım 3: Ogrenciler sınıfından nesneleri saklayacak liste isminde bir List koleksiyonunu global olarak oluşturunuz.

List<Ogrenciler> liste = new List<Ogrenciler>();

Adım 4: Ekle butonu Click olayına Ogrenciler sınıfından üretilen nesne kullanılarak sınıfın özelliklerine değer aktarıldıktan sonra List koleksiyonuna ekleme işleminin kodlarını yazınız.

```
private void btnEkle_Click(object sender, EventArgs e)
{
    Ogrenciler ogrenci = new Ogrenciler();
    ogrenci.Numara = int.Parse(txtNumara.Text);
    ogrenci.AdSoyad = txtAdSoyad.Text;
    ogrenci.DersNotu =int.Parse(txtDersNotu.Text);
    liste.Add(ogrenci);
    Bagla();
}
```

Adım 5: Bagla() ismindeki metot ile List koleksiyonunu DataGridView kontrolüne veri bağlama işlemini gerçekleştiriniz.

private void Bagla()
{
 gridListe.DataSource = null;
 gridListe.DataSource = liste;
}

Adım 6: DataGridView kontrolü içindeki hücrelerin Double Click olayında çift tıklanan hücrenin bulunduğu satırdaki bilgileri TextBox kontrollerine aktaran kodlamayı yapınız.

```
private void gridListe_CellDoubleClick(object sender,DataGridViewCellEventArgs e)
{
    txtNumara.Text = gridListe.CurrentRow.Cells[0].Value.ToString();
    txtAdSoyad.Text = gridListe.CurrentRow.Cells[1].Value.ToString();
    txtDersNotu.Text = gridListe.CurrentRow.Cells[2].Value.ToString();
}
```

Sıra Sizde >>

Form içine Sil isminde bir buton ekleyiniz. Sil butonu Click olayına DataGridView kontrolünün seçili satırındaki bilgileri silme işlemini gerçekleştiren kodlamayı yapınız.





ÖLCME VE DEĞERLENDİRME - 5

Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1. Form uygulamaları ile bilgisayar ortamında çalışan kullanıcı etkileşimli geliştirilebilir.

2. Bir Windows form projesinde uygulamanın hangi formdan başlayacağı Program.cs dosyasındaki Main fonksiyonu içindeki metodu ile belirlenir.

3. Windows form sınıfı isim uzayı içinde bulunur.

ControlBox- CenterToScreen - Load - AcceptButton - Show - FormClosed - CancelButton - Hide

4. Yukarıda form sınıfı için verilen özellik, metot ve olayların işlevlerini tablodaki boşluklara yazınız.

Formun ekranın ortasında açılmasını sağlar.
Form kapandığında çalışan olaydır.
Formun üzerindeki büyültme, küçültme ve kapatma butonlarını gösterir veya gizler.
Form açılırken çalışan olaydır.
Form aktifken klavyeden enter tuşuna basıldığında belirlenen bir butonun tıklanma olayı gerçekleşir.
Formu göstermek için kullanılan metottur.

5. Aşağıdakilerden hangisi Toolbox penceresi içindeki Containers sekmesinde bulunmaz?

A) GroupBox

C) TabControl

D) SplitContainer E) WebBrowser

6. Aşağıdakilerden hangisi MenuStrip içine eklenen nesnelerden biri değildir?

B) Panel

A) Label

B) MenuItem E) TextBox C) ComboBox

10. SINIF

D) Separator

7. Diyalog pencereleri ile ilgili aşağıda verilen bilgilerden hangisi vanlıstır?

A) OpenFileDialog, açılacak olan dosyanın konumunu belirler.

B) SaveFileDialog, kaydedilecek dosyanın konumunu belirler.

C) PrintDialog, yazdırma işlemini gerçekleştirir.

D) ColorDialog, renk paletinden renk seçilmesini sağlar.

E) FontDialog, yazı tipinin seçilmesini sağlar.



	FORM UYGULAMALARI	
BİRİMİ	• ••	
		ſ
8.		
Başlık		×
	Mesaj	
	Evet Hayır İp	tal
Görseldeki metodu ici	gibi bir MessageBox penceresi oluş	sturmak için aşağıda verilen kodlardan hangisi Show
A) "Mesai	" "Baslik" MessageBoyButtons VesNo	MessageBoyIcon Error
B) "Mesai	". "Baslik". MessageBoxButtons YesNo(Cancel, MessageBoxIcon, Warning
C) "Baslik	" "Mesai" MessageBoxButtons VesNo	Cancel MessageBoxIcon Warning
	" "Mesai" MessageBoxButtons YesNo	MessageBoyIcon Error
E) "Mosai	" " Bashk " MossageBoxButtons OKCa	uncol MassageBoxIcon Warning
E) Mesaj		incel, Messageboxicon. Warning
III. Tür IV. Gi	n karakterleri büyük harfe çevirir. rilmesi zorunlu harfler için kullanılır.	
A) #,	\$,<,L B) 9, \$, <, A	C) 0, \$, < , a
A) #,	\$, < , L B) 9, \$, < , A D) 0, \$, > , L	C) 0, \$, < , a E) 9, \$, > , A
A) #, 10.	\$, < , L B) 9, \$, < , A D) 0, \$, > , L	C) 0, \$, < , a E) 9, \$, > , A
A) #,10.I. Tek	\$, < , L B) 9, \$, < , A D) 0, \$, > , L veri için basit veri bağlama kullanılır.	C) 0, \$, < , a E) 9, \$, > , A
 A) #, 10. I. Tek II. Bird 	\$, < , L B) 9, \$, < , A D) 0, \$, > , L X veri için basit veri bağlama kullanılır. Ien çok veri için kompleks veri bağlama	 C) 0, \$, < , a E) 9, \$, > , A na kullanılır.
 A) #, 10. I. Tek II. Bird III. Bas 	\$, < , L B) 9, \$, < , A D) 0, \$, > , L veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama it veri bağlama için Binding sınıfından	C) 0, \$, < , a E) 9, \$, > , A na kullanılır. üretilen nesne kullanılır.
 A) #, 10. I. Tek II. Bird III. Bas IV. Cor 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L veri için basit veri bağlama kullanılır. ien çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlama 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir.
 A) #, 10. I. Tek II. Bird III. Bas IV. Cor V. Kor 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlan npleks veri bağlamada kontrolün Data 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır.
 A) #, 10. I. Tel- II. Bird III. Bas IV. Cor V. Kor Yukarıdal 	\$, < , L B) 9, \$, < , A D) 0, \$, > , L a veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama it veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlan npleks veri bağlamada kontrolün Data i bilgilerden hangileri doğrudur?	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır.
 A) #, 10. I. Tele II. Bird III. Bas IV. Con V. Kor Yukarıdal A) I-II 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L x veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlam nboBox kontrolü kompleks veri bağlam npleks veri bağlamada kontrolün Data xi bilgilerden hangileri doğrudur? B) I-II-III 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır. C) I-II-V
 A) #, 10. I. Tel- II. Biru III. Bas IV. Cor V. Kor Yukarıdal A) I-II 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L x veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlam npleks veri bağlamada kontrolün Data x bilgilerden hangileri doğrudur? B) I-II-III D) I-II-III-IV 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır. C) I-II-V E) I-II-III-V
 A) #, 10. I. Tel- II. Biru III. Bas IV. Cor V. Kor Yukarıdal A) I-II 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L x veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlam npleks veri bağlamada kontrolün Data x bilgilerden hangileri doğrudur? B) I-II-III D) I-II-III-IV 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır. C) I-II-V E) I-II-III-V
 A) #, 10. I. Tel- II. Biru III. Bas IV. Con V. Kor Yukarıdal A) I-II 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L x veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlam npleks veri bağlamada kontrolün Data x bilgilerden hangileri doğrudur? B) I-II-III D) I-II-III-IV 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır. C) I-II-V E) I-II-III-V
 A) #, 10. I. Tel- II. Bird III. Bas IV. Con V. Kor Yukarıdal A) I-II 	 \$, < , L B) 9, \$, < , A D) 0, \$, > , L a veri için basit veri bağlama kullanılır. den çok veri için kompleks veri bağlama için Binding sınıfından nboBox kontrolü kompleks veri bağlam npleks veri bağlamada kontrolün Data ki bilgilerden hangileri doğrudur? B) I-II-III D) I-II-III-IV 	 C) 0, \$, < , a E) 9, \$, > , A a kullanılır. üretilen nesne kullanılır. na için uygun değildir. Source özelliği kullanılır. C) I-II-V E) I-II-III-V



10. SINIF

•

-0



VERİ TABANI İŞLEMLERİ

NELER ÖĞRENECEKSİNİZ?

Bu öğrenme biriminde;

- Veri tabanının ne olduğunu ve ne işe yaradığını açıklayacak,
- Veri tabanı yönetim yazılımını kuracak,
- Yapılandırılmış sorgu (SQL) dilini öğrenecek,
- Veri tabanı ve tablolar oluşturacak,
- Veri türlerini öğrenecek,
- Tabloları ilişkilendirecek,
- SQL komutlarını kullanacak,
- Veri tabanı ile nesne tabanlı programlama arasında bağlantı kuracak,
- Nesne tabanlı programlama üzerinden SQL komutlarını kullanacak,
- Entity Framework yapısını öğrenecek,
- Projenize Setup hazırlayıp istediğiniz bilgisayara Setup kuracaksınız.

ANAHTAR KELİMELER

Veri tabanı, MySQL, Connection String, SQL komutları, Entity Framework, Setup hazırlama

NESNE TABANLI PROGRAMALAMA





HAZIRLIK ÇALIŞMALARI

1. Kullandığınız sosyal medya uygulamalarının nasıl çalıştığını düşünüyorsunuz? Fikirlerinizi arkadaşlarınızla paylaşınız.

2. Program, veri ve veri tabanı kavramları size neler ifade ediyor?

6. VERİ TABANI İŞLEMLERİ

Bu öğrenme birimi, adım adım gidilecek basit ve eğlenceli bir proje bölümüdür. Birkaç haftalık bir süre sonunda 350 satır civarı kod yazılacaktır. Bu süreçte sınıf gruplara ayrılarak çalışmalar yapılacaktır. Hem veri tabanı hem de proje dosyalarını taşımak için bir flash disk bulundurulmalıdır.



6.1. VERİ TABANI YAZILIMININ KURULUMU

En basit anlamda veri tabanı, belirli bir amaca yönelik düzenlenen kayıt ve dosyaların tümüdür. Bilgisayar sisteminde arkadaş ya da müşterilerin ad ve adresleri toplanabilir. Yazılan tüm mektuplar bilgisayar sisteminde bulundurulabilir ve alıcıya göre mektuplara düzen verilebilir. Ödeme ve tahsilat yapılacak hesaplar ya da çek defterinin borçları ve bakiyesi gibi finansal verilerin toplandığı bir grup dosya da olabilir. Konularına göre düzenlenen sözcük işlemci belgeler, en geniş anlamda bir tür veri tabanıdır. Kullanıma göre düzenlenen elektronik tablo dosyaları da ayrı bir tür veri tabanıdır. Başlat menüsündeki tüm program kısayolları da bir veri tabanıdır. Sık Kullanılanlar klasöründe düzenlenen internet kısayolları da bir veri tabanıdır.

Çok düzenli kişiler birkaç yüz elektronik tablo ya da kısayolu klasör ve alt klasörleri kullanarak yönetebilir. Bunu yaptıklarında veri tabanı yöneticisi olurlar. Çözülmeye çalışılan sorunlar büyüdükçe ne yapılabilir? Veriler, çeşitli belgelerde ya da elektronik tablolarda saklanmışken tüm müşterilerle ve onların siparişleriyle ilgili bilgiler nasıl kolayca toplanabilir? Yeni bilgiler girildiğinde dosyalar arasındaki bağlantılar nasıl korunabilir? Verilerin doğru girildiğinden nasıl emin olunabilir? Bilgileri birçok kişiyle paylaşma gereksinimi varsa aynı anda iki kişinin aynı verileri güncelleştirmesi istenilmediğinde ne yapılabilir? Bu tür güçlüklerle karşılaşıldığında bir veri tabanı yönetim sistemine (DBMS-Database Management System) gereksinim vardır.





6.1.1. Veri Tabanı Yönetim Sistemi (Database Management System)

Veri tabanları genellikle veri tabanı yönetim sistemi (DBMS) olarak bilinen kapsamlı bir veri tabanı yazılım programını gerektirir. DBMS, veri tabanı ve son kullanıcılar ya da programlar arasında bir arayüz işlevi görür. Kullanıcıların bilgilerin nasıl organize ve optimize edildiğini yönetmesine, bilgileri almasına ve güncel lemesine olanak sağlar. DBMS aynı zamanda veri tabanlarına ilişkin gözetim ve kontrol faaliyetlerini kolaylaştırarak performans izleme, ince ayar yapma, yedekleme ve kurtarma gibi çeşitli yönetim operasyonlarının gerçekleştirilmesini sağlar.

6.1.2. SQL (Structured Query Language)

Yapılandırılmış sorgu dili (SQL-Structured Query Language); veriyi sorgulamak, manipüle etmek, tanımlamak ve veriye erişim kontrolü sağlamak üzere tüm ilişkisel veri tabanlarında kullanılan bir programlama dilidir. SQL ilk olarak 1970'li yıllarda geliştirilmiş, ardından SQL ANSI standardı uygulanmış ve yazılım şirketleri pek çok uzantının temelini atmıştır. SQL günümüzde oldukça yaygın bir şekilde kullanılsa da yeni programlama dilleri de geliştirilmektedir.

6.1.3. Büyük Veri (Big Data) ve Veri Madenciliği

Veri günümüzde paradan daha değerli bir hâle gelmiştir. Bunun sebebi, büyük firmaların stratejilerini bu verilere göre belirlemesidir. Bir arama yapıldığında o aramayla ilgili reklamların gelmesi bu duruma bir örnektir. Dünyanın en büyük firmalarının interneti kullanan milyarlarca kişi için bu işlemi yaptığı düşünülürse verilerin ne kadar önemli olduğu daha iyi anlaşılacaktır.

6.1.3.1. Big Data

Günümüzde adı sıkça duyulan Big Data (Büyük Veri), her ne kadar teknolojinin ilerlemesi ve kullanım alanlarının artması ile ortaya çıkmış bir kelime olarak görülse de yıllardır içinde bulunulan fakat gelişiminden pek haberdar olunmayan bir olgudur. Bu olguya farkında olmadan sürekli destek verilmekte ve "Büyük Veri" olarak isimlendirilen bu ortama sürekli veri akışı sağlanmaktadır.

Büyük Veri, yükselen teknolojiler ve tüketicilerin artan veri kullanım oranı paralelinde daha fazla çeşitlilik içeren veri kümesidir.

Teknolojinin ilerlemesi ve internetin gelişmesi ile birlikte günümüzde bilginin gücü de ön plana çıktı ve bununla beraber internet dünyasındaki birçok olgu "Bilgi Çöplüğü" olarak anılmaya başladı. Bu çöplükten anlamlı verilerin de çıkabileceğini düşünen yazılım şirketleri, AR-GE çalışmalarını bu bağlamda yürüterek Big Data olarak adlandırılan olguyu ortaya çıkarttı.

Big Data diye isimlendirilen bu olgu, diskte çok fazla yer kaplayan veri çağrışımı yapsa da aslında tam olarak böyle değildir. Big Data; sosyal medya paylaşımları, fotoğraf arşivleri, sürekli kayıt alınan "log" dosyaları gibi farklı kaynaklardan elde edilen tüm verilerin anlamlı ve işlenebilir hâle dönüştürülmüş biçimidir. Big Data kullanım alanları ile ilgili örnekler şöyle sıralanabilir:

• Firmaların müşterileri, malzeme tedarikçileri, şirket içindeki her türlü işlem ve ürünleri ile ilgili trilyonlarca bayt'lık veri toplanmakta ve anlamlı raporlar üretilmektedir.

• Sosyal medya paylaşımları sayesinde her gün milyarlarca kilobayt veri elde edilmektedir.

• Normal bir tüketicinin günlük yaşamında internette yaptığı haberleşme, arama, satın alma ve paylaşma türü işlemlerin yarattığı verilerin tümü saklanmaktadır.

• Büyük sosyal medya sitelerinde günlük işlenen veri boyutu 7 terabyte'a yaklaşmakta ve günden güne artmaktadır.

• Sağlık kuruluşları, hastalarına yönelik bireysel ve kişiselleştirilmiş sağlık hizmetlerini sağlayabilmek için bireysel durumdaki verileri kendi sayısal ortamlarında depolamaktadır.

• Bankalar, müşterileriyle ilgili sakladıkları bilgiler aracılığı ile kullanıcılarını tanıyan ve internet şubesinde o gün için hangi hizmeti aldığını bilen, ana sayfayı ve menüleri etkileşimli hâle getiren (kişiselleştirme uygulamaları), müşterilerine hatırlatmalar yapan, kişiselleştirilmiş arayüz deneyimi ve zengin içerikle sürekli hizmet sağlayan birer şube hâline geldi.

• İlaç depolarında yüz binlerce ilacın adını ve tam olarak nerede bulunduğunu bilen, herhangi bir ilaca ulaşmak istenildiğinde ilacın yerini doğru bir şekilde tespit eden sistemlerin arkasında büyük veriler yatmaktadır.

• Arama motorları sayesinde milyonlarca sayfa arasında en doğru ve en hızlı veriye ulaşabilmek için yine aynı şekilde arka planda büyük veriler ve bunların anlamlı hâle getirilmesi bulunmaktadır.





Biq Data ile yapılabilecek her şey hayal gücü ile sınırlıdır. Sadece ne yapmak ve neyi bilmek istendiğinin iyi belirlenmesi gerekmektedir.

6.1.3.2. Veri Madenciliği

Veri madenciliği, çok büyük veriden faydalı bilgi ve kalıp çıkarma süreci olarak tanımlanabilir. Veri madenciliği; toplama, cıkarma, analiz ve veri istatistiklerini icerir.

Veri madenciliği; daha iyi pazarlama stratejileri geliştirmek, performansı artırmak veya işletmeyi yönetme maliyetlerini azaltmak için satış numaraları ile fiyatlar ve müşteriler dâhil olmak üzere ham verilerin incelenmesinde kullanılır. Veri madenciliği, tüketiciler arasında yeni davranış kalıplarını keşfetmeye de hizmet eder.

Veri madenciliğinin faydaları şöyle sıralanabilir:

Veri madenciliğindeki türetilmiş model, müşteri davranışını daha iyi anlamada yardımcı olur. Bu da daha iyi ve üretken gelecek kararlarına yol açar.

 Veri madenciliği, is icin faydalı olan ancak henüz erişilemeyen pazar fırsatlarını yakalamak icin kullanılır. Veri madenciliği ayrıca pazar alanını belirlemek, pazarlama hedeflerine ulaşmak ve iyi bir yatırım getirisi elde etmek icin kullanılır.

 Veri madenciliği, potansiyel yatırım alanlarını keşfederek ve tanımlayarak işletme maliyetini düşürmeye yardımcı olur.

6.1.4. MySQL Veri Tabanı

MySQL şu an dünyada en çok kullanılan veri tabanıdır. MySQL veri tabanının en önemli özelliklerinden biri de ücretsiz sürümlerinin olmasıdır.

6.1.4.1. MySQL Veri Tabanının Kurulumu

Öncelikle mysql.com adresine girilir.



Görsel 6.2'deki ekran gelecektir. Üst tarafta yer alan DOWNLOADS'a tıklanır.





The we	orld's most popular open source database		Contact MySQL Login Regis
MySQL. Mys	QL.COM DOWNLOADS DOCUMENTATIO	N DEVELOPER ZONE	f ⊻ in å
MUSOL Data	abasa Sarvisa		
Grant Control of the second seco	ing manageo valualese service to deputy courd-naire vice ecurity Fixes	e approximations stating on the works of an advance of the statistical constraints of the statistical of the	
Try Now >			
Contact Sales USA: +1-806-221-0034 Canada: +1-866-221-0634 Germany: +49 89 143 01280 France: +33 1 57 60 83 57	MySQL Enterprise Edition MySQL Enterprise Edition includes the most con Learn More » Custome Download » Trial Download »	prehensive set of advanced features, management tools and technical support for MySQL	
taby: 30 02 249 50 120 UI: 44 207 553 8447 Japan: 0120-00556 China: 10800-811-0823 India: 0088001005870 More Countries > Contact Us Online >	MySQL Cluster CGE MySQL Cluster is a real-time open source transa • MySQL Cluster • Pils, everything in MySQL Enterprise Edition Learn More » Clustomer Download • (Select Patches & Update Trail Download •	ctional database designed for fast, always-on access to data under high throughput conditions. sr Tab, Product Search)	

Görsel 6.3: mysql.com downloads sayfası

Görsel 6.3'teki sayfada en altta yer alan "MySQL Community (GPL) Downloads" kısmına tık-

dev.mysql.com/downloads/		
MySQL Community Downloads		
MySQL Yum Repository MySQL APT Repository MySQL SUSE Repository MySQL Community Server MySQL Cluster MySQL Router MySQL Router MySQL Shell MySQL Workbench MySQL Installer for Windows	C API (libmysqlclient) Connector/C++ Connector/NeT Connector/Nede.js Connector/ODBC Connector/Python MySQL Native Driver for PHP MySQL Benchmark Tool Time zone description tables	
MySQL for Visual Studio © 2020, Oracle Corporation and/or its affiliates Legal Policies Your Privacy Rights Terms of Use Trademark Policy Contributor Agreen	Download Archives	

Görsel 6.4: MySQL Community Downloads sayfası

Görsel 6.4'teki sayfada uygun olan MySQL sürümü seçilir.

lanır.



Mİ •		• •		:/
a dev.mysql.	.com/downloads/installer/			7
	MySQL Community Download MySQL Installer	ds		
	General Availability (GA) Releases Archives	V		
	MySQL Installer 8.0.21			
	Select Operating System: Microsoft Windows	~	Looking for previous GA versions?	
	Windows (x86, 32-bit), MSI Installer	8.0.21	24.5M Download	
	(mysq-installer-web-community-8.0.21.0.ms) Windows (x86, 32-bit), MSI Installer (mysq-installer-community-8.0.21.0.ms))	8.0.21	MD5: cf2b46ba35a443f41fb8e94a6e51d93 Signature 427.6M Download MD5: b52294aa854356c266e9a9aec737ba88 Signature	
	• We suggest that you use the MD5 checksums and G	nuPG signatures to verify the i	ntegrity of the packages you download.	

Görsel 6.5: MySQL indirme sayfası

Görsel 6.5'teki sayfada uygun versiyon seçilir. Bu kurulum "**Desktop**" örneği üzerinden devam edecektir. İndirme işlemi bittikten sonra kurulum adımlarına geçilir.

MySQL. Installer Adding Community	Choosing a Setup Type Please select the Setup Type that suits yo	our use case.
Choosing a Setup Type Installation Installation Complete	 Developer Default Installs all products needed for MySQL development purposes. Server only Installs only the MySQL Server product. Client only Installs only the MySQL Client products, without a server. Full Installs all included MySQL products and features. Custom Manually select the products that system. 	Setup Type Description Installs the MySQL Server and the tools required for MySQL application development. This is useful if you intend to develop applications for an existing server. This Setup Type includes: * MySQL Server * MySQL Shell The new MySQL client application to manage MySQL Servers and InnoDB cluster instances. * MySQL Router High availability router daemon for InnoDB cluster setups to be installed on application

Görsel 6.6: MySQL kurulum türü seçme ekranı

Görsel 6.6'daki ekranda ne tür bir kurulum istenildiğinin seçilmesi gerekmektedir. Developer Default (Geliştirici Standart) kurulum için uygundur. İstenilen seçenek işaretlenerek "**Next**" düğmesine basılır.





VERİ TABANI İŞLEMLERİ

Görsel 6.7: Kurulacak bileşenler

Görsel 6.7'deki ekranda kurulacak bileşenler listelenmiştir. "**Executer**" düğmesine basılır ve kuruluma başlanır (Görsel 6.8).

SQL Installer			-	
MySQL. Installer Adding Community	Installation The following products will be installed.			
	Product	Status	Progress	Notes ^
Choosing a Setup Type	🖉 📐 MySQL Server 8.0.21	Complete		
Installation	SQL Workbench 8.0.21	Complete		
Product Configuration	🐼 🛐 MySQL Notifier 1.1.8	Complete		
:	🐼 🛐 MySQL For Excel 1.3.8	Complete		
Installation Complete	🔗 🛐 MySQL for Visual Studio 1.2.9	Installing	92%	
	MySQL Shell 8.0.21	Ready to Install		
	MySQL Router 8.0.21	Ready to Install		
	Connector/ODBC 8.0.21	Ready to Install		
	Connector/C++ 8.0.21	Ready to Install		
	Connector/J 8.0.21	Ready to Install		
	Connector/NET 8.0.21	Ready to Install		
	Connector/Python 8.0.21	Ready to Install		
	MvSOL Documentation 8.0.21	Ready to Install		>
	Show Details >			
		< Back	Execute	Cancel

Görsel 6.8: Kurulumun başlaması



NySQL Installer	_	- 0	\times
MySQL. Installer Adding Community	Product Configuration We'll now walk through a configuration wiz	ard for each of the following products.	
Choosing a Setup Type	You can cancel at any point if you wish to le products.	ave this wizard without configuring all the	
Installation	Product	Status	
Product Configuration	MySQL Server 8.0.21 MySQL Router 8.0.21	Ready to configure Ready to configure	
Installation Complete	Samples and Examples 8.0.21	Ready to configure	
	<		>
		Next > Canc	el

Görsel 6.9: Product Configuration ekranı

Görsel 6.9'daki ekranda kurulan bileşenlerin ayarları yapılabilir. Önemli olan kısım "**MySQL Server 8.0.21**" yazan bileşendir. Bu bileşen, MySQL Server ayarlarının yapılacağını gösterir.

Installer erver 8.0.21	High Availability
king	 Innobs Cluster The InnobB cluster technology provides an out-of-the-box high availability (HA) solution for MySQL using Group Replication.
Method	
nd Roles	
ce	MySQL Shell
	Client App \longleftrightarrow MySQL Router \longleftrightarrow
ns	InnoDB Cluster
ration	Note: InnoDB cluster requires a minimum of three MySQL server instances to provide a fully automated HA solution. Members of a cluster should be located such that network communication latency between servers is low.
	Next > Cance

Görsel 6.10: MySQL Server ayarları

Görsel 6.10'daki ekranda standart bir kurulum için herhangi bir ayar yapılmasına gerek yoktur. "**Next**" düğmesine basılır.





Görsel 6.11: Type and Networking ayarları



NySQL Installer	- 🗆 X
MySQL. Installer MySQL Server 8.0.21	Outpentication Method Outpenties and Encryption for Authentication (RECOMMENDED) Marcol & compared to a new arthestication based on improved to account (Marcol & Cold 2005)
High Availability	methods. It is recommended that all new MySQL Server installations use this method going forward.
Type and Networking	Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (sching charge authentication)
Authentication Method	
Accounts and Roles	this new method. If clients and applications cannot be updated to support this new
Windows Service	authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.
Apply Configuration	 Use Legacy Authentication Method (Retain MySQL 5.x Compatibility) Using the old MySQL 5.x legacy authentication method should only be considered in the following cases: If applications cannot be updated to use MySQL 8 enabled Connectors and drivers. For cases where re-compilation of an existing application is not feasible. An updated, language specific connector or driver is not yet available. Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.
	< Back Next > Cancel

Görsel 6.12: Authentication Method ekranı

Görsel 6.12'deki ekranda MySQL Server erişim güvenliği için seçim yapılması istenmektedir. Günümüzde güvenlik kavramı çok önemlidir çünkü kimse veri tabanına başkalarının girmesini istemez. Bu ekranda "**Use Strong Password Encryption for Authentication**" seçeneği işaretlenerek üst düzey güvenlik ayarları seçilir.



MySQL Installer	-			_		×
MySQL. Installer MySQL Server 8.0.21	Accounts and Ro Root Account Password Enter the password for the place.)les root account. Pleas	e remember to store	e this password i	in a secure	
High Availability	MySQL Root Password:	•••••]	
Type and Networking	Repeat Password:	•••••	the Wesk			
Authentication Method		Password streng	un weak			
Accounts and Roles						
Windows Service	MySQL User Accounts					
Apply Configuration	Create MySQL user accou consists of a set of privile	unts for your users an ges.	nd applications. Assi	ign a role to the	user that	
	MySQL User Name MEB	Host %	User Role DB Admin		Add Use	er er
					Delete	
			< Back	Nexts	Cance	al

Görsel 6.13: Accounts and Roles ayarları

Görsel 6.13'teki ekranda MySQL Server erişimi için bir şifre belirlenmesi ve Server'a erişecek başka kullanıcılar varsa onların eklemesi gereklidir. Görsel 6.13'te üst kısımda MySQL Server şifresi belirlenmişken Görsel 6.14'teki **MEB** isminde bir kullanıcı **"Add User**" düğmesi ile eklenmiştir.

NySQL Installer	- 🗆 X
MySQL. Installe MySQL Server 8.0.21	Accounts and Roles Root Account Password Enter the password for the root account. Please remember to store this password in a secure MySQL User Account X
High Availability	Please specify the user name, password, and database role.
Type and Networking Authentication Method	User Name: MEB Host: <all (%)="" hosts=""></all>
Accounts and Roles	Role: DB Admin ~
Windows Service	Authentication: MySQL
Apply Configuration	MySQL user credentials Password: Confirm Password: Add User
	Password strength: Weak Edit User
	OK Cancel
	< Back Next > Cancel

Görsel 6.14: MySQL Server'a kullanıcı eklenmesi







Görsel 6.15: Service ayarları

Görsel 6.15'teki "Next" düğmesine basılır.

MySQL Installer		×
MySQL. Installer MySQL Server 8.0.21	Apply Configuration Click [Execute] to apply the changes Configuration Steps Log	
High Availability Type and Networking Authentication Method Accounts and Roles Windows Service Apply Configuration	 Writing configuration file Updating Windows Firewall rules Adjusting Windows service Initializing database (may take a long time) Starting the server Applying security settings Creating user accounts Updating the Start menu link 	
	< Back Execute Cancel	

Görsel 6.16: Apply Configuration ekranı

Görsel 6.16'daki ekran ile MySQL Server ayarlarının uygulanması sağlanır. "**Execute**" düğmesine basılır ve ayarlar uygulanmaya başlanır (Görsel 6.17).





VERİ TABANI İŞLEMLERİ



Görsel 6.17: Server ayarları uygulama ekranı

MySQL Installer	- 0	×
MySQL. Installer MySQL Server 8.0.21	Apply Configuration The configuration operation has completed. Configuration Steps Log	
High Availability Type and Networking Authentication Method Accounts and Roles Windows Service Apply Configuration	 Writing configuration file Updating Windows Firewall rules Adjusting Windows service Initializing database (may take a long time) Starting the server Applying security settings Creating user accounts Updating the Start menu link 	
	The configuration for MySQL Server 8.0.21 was successful. Click Finish to continue. Finish	

Görsel 6.18: Server ayarlarının tamamlanması

Görsel 6.18'de görüldüğü gibi bütün ayarlar doğru bir şekilde uygulanmış ve MySQL Server kullanıma hazır duruma gelmiştir.



MySQL. Installer	Connect To Server
	Select the MySQL server instances from the list to receive sample schemas and data.
Connect To Server	Show MySQL Server instances that may be running in read-only mode
pply Configuration	Server Fort Arch type Status MySQL Server 8.0.21 3306 X64 Stand-alone Server Connection succeeded.
	Provide the credentials that should be used (requires root privileges). Click "Check" to ensure they work. User name: root Credentials provided in Server configuration Password:

Görsel 6.19: Connect To Server ekranı

Görsel 6.19'daki ekranda kurulum aşamasında belirlenen şifre ile MySQL Server'ın çalışıp çalışmadığı test edilir. "**Connection Succeeded**" yazısı görülüyorsa MySQL Server ayarları doğru yapılmıştır ve MySQL Server doğru çalışmaktadır. "Next" tuşuna basılarak Görsel 6.20'deki ekrana ulaşılır.



Görsel 6.20: MySQL Server kurulumunun bitiş ekranı



	VERİ TABANI İŞLEMLERİ
)GRENME BİRİMİ	/•• • •

6.2. MYSQL SERVER ARAYÜZ (WORKBENCH) EKRANI

MySQL Server kurulumu bittikten sonra MySQL kurulumunda yüklenen "**Workbench**" özelliği ile MySQL Server'ı yönetecek arayüze ulaşılır. Başlat menüsünden ya da görev çubuğuna gelen MySQL simgesindeki "**Manage Instance...**" kısmından Workbench arayüzü başlatılır (Görsel 6.21).



Görsel 6.21: Görev çubuğundan MySQL Workbench çalıştırma

MySQL Work	doench				- a ×
	fen Database Toda Scripting Help	Welco MySQLWo create ar design and r	The second secon	Dench to design, well as from other	×
	MySQL Connections ④⑤ Local instance MySQL80 ∦ reet ₩ localhers3306	Browse Documentation >	Concentration Concentration Concentration Concentration Concentration	Discuss on the Forums >	Filter connections

Görsel 6.22: MySQL Workbench arayüz giriş ekranı

Görsel 6.22'deki ekranda kurulum esnasında belirlenen parola girilmelidir. Ekranın alt tarafındaki **Save password in vault**" seçeneği işaretlenerek bir daha şifre sorulmaması sağlanır.

Şifre girildikten sonra MySQL başlangıç ekranı ile karşılaşılır. Bu ekranda Server ile ilgili bilgiler ve değerler yer almaktadır.





MySQL Workbench				- 0
le Edit View Query Databas	se Server Tools Scripting Help			۵
Aavigator CHEMAS	Admentation - Server Status -> Connection Name Local instance MySQL80		SpL4dthors ==	🎭 Jump to ic context h
 Bys world 	Hote: Ld700-INT308. Solat: Hyd2 Berver 3006 Herror: 8.021 (Myd20. Commund. Confector 10 Confector 10 Confector 10 Confector 10 Rurring Sroz: Hinde 16 22-0358 2000.	y Server - CPL] HyGQ, Server 8.0(my.au) 22239) an	Sever Stelas CNMLass Convections Instantion. Running 13% 4 Trainic Key Efformery 4.97 KB/s 0.0%	get help fo ret position automatic h
idministration Schemas	Available Server Features Pedramors Stenar Pedramors Stenar On Protect Page On Protect Page On Protect Page On Protect Page On Server Directories	Windows Authentication: ① Off Peaseword Valdation: ① n/a Audition: ① n/a Presail: ① n/a Presail Taco:: ② n/a	Selects per Second Not00 Buffer Usage O S1.296 Not00 Reads per Second Not00 Virtues per Second O O	
information	Base Directory: C:\Program Files\MySQL\MySQL Server 8.0\ Data Directory: C:\ProgramData\MySQL\MySQL Server 8.0\Data	1		

Görsel 6.23: MySQL Workbench Server Status ekranı

Görsel 6.23'te görülen ekrandaki alanlar şunlardır:

VERİ TABANI İŞLEMLERİ

• **Navigator:** Bu penceredeki "Schemas" alanında veri tabanları görünür. MySQL kurulduktan sonra kendi içinde üç örnek veri tabanı görünmektedir.

- Information: Seçilen bileşene ait (veri tabanı, tablo vb.) bilgilerin görülebileceği alandır.
- Output: Yapılan işlemlerin sonuçlarının görülebileceği alandır. İşlem doğru tamamlanmışsa yeşil

renkte, işlemde hata oluşmuşsa hata mesajıyla beraber kırmızı renkte görünecektir.

- **SQL Additions:** Komutlar ve bileşenlerle ilgili yardım alınabilecek alandır.
- **Sekmeler:** MySQL ile ilgili her işlemin yapıldığı alandır. Orta alanın tamamını kullanır. Erişilen bütün bileşenlerle ilgili işlemler bu alanda açılacak sekmelerin içinde yapılır.





6.3. VERİ TÜRLERİ

C#'ta değişkenlerin veri tipi belirlendiği gibi SQL veri tabanı tasarımı yapılırken de her kolon (alan) için veri tipi tanımlanır. Bu tanımlamalar yapılırken dikkat edilmesi gereken noktalardan biri, mümkün olduğunca en uygun ve hafızada en az yer kaplayacak veri tipini seçmektir (Tablo 6.1).

Veri Tipi	Türü	Açıklama				
char(n)	String	Uzunluğu değişmeyen sabit verileri saklar. n değeri 10 ise daha kısa bir değer giri- lince kalan boşluğu kendi tamamlar ve öylece saklar.				
varchar(n)	String	Değişebilir uzunlukta verileri saklar. En fazla 8.000 karakter alır. n değeri maksimum değerdir. Daha kısa bir değer girilse bile bu değeri olduğu gibi kaydeder.				
text	String	Değişebilir uzunlukta karakterleri saklar. En fazla 2 GB metin içerir.				
nchar	String	abit uzunlukta Unicode karakterleri saklar. char tipinden farkı, çoklu dil ve Unicode esteğinin olmasıdır. En fazla 4.000 karakterdir.				
nvarchar	String	Değişebilir uzunlukta verileri saklar. varchar tipinden farkı, çoklu dil ve Unicode desteğinin olmasıdır. En fazla 4.000 karakterdir.				
bit	Boolean	0, 1 ve Null değerlerini saklar.				
tinyint	Tam sayı	0 ile 255 arasındaki değerleri saklar.				
smallint Tam sayı 32.70		32.768 ile 32.767 arasındaki değerleri saklar.				
int Tam sayı 2.147.483.648 ile 2.147.483.647 arasındaki de		2.147.483.648 ile 2.147.483.647 arasındaki değerleri saklar.				
bigint Tam sayı		9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasındaki değerleri saklar.				
datetime	Tarih	1 Ocak 1753–31 Aralık 9999 arasındadır.				
smalldatetime	Tarih	1 Ocak 1900–6 Haziran 2079 arasındadır.				
date	Tarih	1 Ocak 0001–31 Aralık 9999 arasındadır. Sadece tarih içerir, saati saklamaz.				
decimal	Ondalık	Hafızada ondalık sayı türünden veri tutmak için kullanılır. Örneğin 102,35 sayısını hafızada tutmak için DECIMAL (5,2) şeklinde veri türü tanımlanmalıdır.				
float	Ondalık	Hafızada ondalık sayı türünden veri tutmak için kullanılır.				

Tablo 6.1: Temel Veri Tipleri

Bunlar dışında da veri türleri mevcuttur.

6.4. VERİ TABANI TASARIMI

Kodlama öncesi nasıl algoritma yapılıyorsa veri tabanı oluşturmadan önce de bir veri tabanı tasarımı yapmak, kabaca bir taslak çıkarmak önemlidir. Öncelikle veri tabanının hangi amaçla oluşturulacağı belirlenmelidir. Bu amaca yönelik bir isim belirlenmesi de doğru olacaktır. Yapılacak veri tabanı, birbirine bağlı alanların bulunduğu bir yapıda (**ilişkisel**) olmalıdır.

Burada öğrenci bilgilerini, notlarını ve derslerini kayıt altında tutacak bir veri tabanı tasarlanacaktır. Veri tabanında şu an için üç adet tablo olacaktır. Bu tabloların içine gerekli alanlar tanımlanmalıdır. İstenildiği zaman tablolarda ekleme ve çıkarma gibi işlemler yapılabilir.





6.4.1. Veri Tabanı Oluşturma

Veri tabanları, SQL komutları ve MySQL arayüzü kullanılarak iki farklı şekilde oluşturulabilir. SQL

komutuyla veri tabanı oluşturmak için Görsel 6.24'teki MySQL Workbench arayüzünde bulunan simgesine tıklanır.





SQL komutuyla veri tabanı oluşturulurken "**Create Database İsim**" komutu ile belirtilen isimde veri tabanı (**database**) elde edilir (Görsel 6.25).

orkbench										
al instance	MySO	$2L80 \times$								
View (Query	Datab	ase <u>S</u> erver	Tools	Scripting	<u>H</u> elp				
6	0	5	6							
			SQL File	3* ×						
		63		1 🖌	F A C	9	00	🐻 Limit :	to 1000 rows	• 🏡 🛫 🤇
ects			1.	Cr	eate	e Da	atab	ase	e0kı	l
			2							
	0	teast too								
	Ou	ipui ····								
	đ	Actio	n Output		-					
		#	Time	Action				Message	2	
	0	1	21:14:38	Create	Database	eOkul		1 row(s)	affected	

Görsel 6.25: Veri tabanı oluşturma ve komutun doğru çalıştığını gösteren output çıktısı

ÖĞRENME BİRİMİ			
	Navigator		
	SCHEMAS	63	
	A Filter objects		
	eokul		
	sakila		
	► 🤤 world		

Görsel 6.26: Veri tabanının MySQL'e eklenmesi

Sadece tek bir komutla içinde milyonlarca veri tutabilecek bir veri tabanı oluşturuldu (Görsel 6.26).

6.4.2. Veri Tabanında Anahtarlar (Keyler)

• Primary Key (PK-Birincil Anahtar)

Primary Key (birincil anahtar), bir veri tablosunda yer alan her satır için bir vekil veya tanımlayıcı (**identify**) görevi görür, kısıtlamadır (**constraint**) ve eşsizdir. Satırlara ait değerlerin karışmaması adına bu alana ait bilginin tekrarlanmaması gerekir. Birincil anahtarın temel işlevi, verilerin hangi satıra dizileceğine veya hangi satırda düzenleneceğine karar vermesidir. Çoğunlukla tek bir alan (id, user_id, e_mail, username, national_identification_number vb.) olarak kullanılsa da birden fazla alanın birleşimiyle oluşturulabilir. Sayılar genelde birincil anahtar olarak seçilir ancak bu bir zorunluluk değildir. Birincil anahtar değeri boş geçilemez ve Null değeri alamaz. **İlişkisel veri tabanlarında** (Relational Database Management System) mutlaka birincil anahtar olmalıdır.

• Foreign Key (FK-Yabancı Anahtar)

Foreign Key (yabancı anahtar) ikincil anahtar olarak da ifade edilmektedir. Bir veri tablosuna girilebilecek değerleri başka bir veri tablosundaki alanlarla ilişkilendirmeye yarar. Başka bir tablonun birincil anahtarının bir diğer tablo içinde yer almasıdır. **Çoğunlukla bir ana tablo (parent) ile alt tablonun (child) ilişkilendirilmesinde kullanılır.** Bu sayede olası veri tekrarlarının önüne geçilebilmekte ve ilerleyen zamanda söz konusu olabilecek güncellemelerde ilgili verilerin her yerde güncellenmesi sağlanabilmektedir.

• Unique Key (UQ-Tekil veya Benzersiz Anahtar)

Unique Key (tekil anahtar), yer aldığı tablo içinde bir değeri sadece bir kere alır. İlgili değerin tekrar atanmasına izin verilmez. **Bu anahtarın işlevi birincil anahtar ile benzerlik taşısa da ikisi arasındaki en önemli farklılık, Unique Key'in Null değeri alabilmesidir.**

PK ile UQ arasındaki farklar şöyle sıralanabilir:

• Her birincil anahtar benzersiz iken her benzersiz anahtar birincil değildir.

• Bir tablo içinde birden fazla alan tek birincil anahtar ile tanımlanabilir ancak birincil anahtar yapısı her tabloda sadece bir tane olabilir. Benzersiz anahtarda ise böyle bir sınırlama yoktur.

• Birincil anahtarda Null kullanımına izin verilmez ancak benzersiz anahtarlarda Null kullanımına izin verilir.

• Birincil anahtar ile veri tablosu üzerinde bir Index tanımı oluşturulur ve her kaydın benzersiz bir tanımı yapılır. Benzersiz anahtar kullanımında ise değerlerin benzersiz olup olmadığına bakılır.




VERİ TABANI İŞLEMLERİ

6.4.3. Tablo Oluşturma

Bir veri tabanı içinde onlarca ya da yüzlere tablo yer alabilir. Tablolar, veri tabanları içinde verileri tutan yapılardır. En başta analiz doğru yapılırsa tablolar oluşturulurken yaşanacak sıkıntılar en aza indirilir. Tabloda özel bir durum yoksa mutlaka bir tane primary key bulundurulur. Böylece tablo, diğer tablolarla ilişkilendirilmek istenildiğinde sıkıntı yaşanmaz. Tablo oluştururken komutlar veya arayüz kullanılabilir. Aşağıda bir tablo taslağı görülmektedir.

CREATE TABLE tabloadı (sütunAdı1 veriTipi diğerParametreler , sütunAdı2 veriTipi diğerParamatreler , ...)

Create Table komutu kullanılırken tablonun her alanı (sütunu) için veri tipi ve başka parametreler girilir. Bu parametreler şu şekilde sıralanabilir:

• **Not Null:** Alan için bir değerin mutlaka girilmesi gerektiğini gösterir (Bu parametre daha çok birincil anahtarlar için kullanılır.).

- Primary Key: Alanı birincil anahtar olarak belirler.
- Auto_increment: Primary key alanına kayıt eklendikçe otomatik olarak bir sayı artacaktır.

SID int primary key auto_increment, Görsel 6.27: Örnek bir SQL satırı

Tablonun SQL komutları yazılır (Görsel 6.28).

```
Create Table ogrenciBilgi(
  ogrenciNo int primary key,
  ogrenciAdi varchar(20) not null,
  ogrenciSoyadi varchar (20) not null,
  ogrenciBolumu varchar (30) not null,
  ogrenciSinifi tinyint not null,
  ogrencidogumtarihi date not null
  );
```

Görsel 6.28: ogrenciBilgi tablosu için SQL kodları

Create Table komutuyla bir tablo oluşturulduktan sonra bu tabloda yer alacak alanlar tek tek belirlenir. **Tablo oluşturulurken birinci alan (ogrenciNo) primary key olarak belirlenir.**

Sıra Sizde

Primary key alanının neden auto increment olarak belirlenmediğini arkadaşlarınızla tartışınız.

Öğrenci numarasından sonra öğrencinin adı, soyadı ve bölümü varchar veri türüyle belirlenir. Varchar veri türlerinin yanındaki sayılar, o alan için hafızada ne kadar yer ayrıldığını göstermektedir. Daha sonra öğrencinin sınıfının (9, 10, 11, 12) kaydedileceği sınıf alanı ve öğrencinin doğum tarihinin girileceği doğum tarihi alanı belirlenir. Görsel 6.28'deki kodlar yazıldıktan sonra üst taraftaki yıldırım simgesiyle kodlar çalıştırılabilir ve sonuçlar "Output" bölmesinden takip edilebilir (Görsel 6.29, Görsel 6.30).

10, SINIF





Görsel 6.29: Sol taraftan veri tabanı seçmeden kodu çalıştırınca ortaya çıkan hata

6 21:35:04 Create Table ogrenciBilgi (ogrenciNo int primary key auto_increment, ogrenciAdi varchar(20) not null, ogrenciSoyadi varchar (20) not null, ogrenciBolumu varchar (30) no... 0 row(s) affected

Görsel 6.30: SQL kodunun düzgün çalıştığına dair output çıktısı



Görsel 6.31: ogrencibilgi tablosunun veri tabanına eklenmesi

Görsel 6.32'deki kodlarla dersler tablosu oluşturulur.

```
create table dersler(
derskodu varchar(4) primary key,
dersadi varchar(45) not null
);
```

Görsel 6.32: Dersler tablosunun SQL kodları

Burada dersler için 4 karakterlik bir primary key alanı ve ders adının kaydedileceği bir alan oluşturulur. Tıpkı veri tabanı gibi tablolar da hem SQL kodlarıyla hem de MySQL Workbench arayüzüyle oluşturulabilir. Üçüncü tablo, MySQL arayüzü kullanılarak oluşturulur (Görsel 6.33).

SCHEMAS	43	
Q , Filter objects		
🔻 🗐 eokul		
Tabian	Create Table	
	Create Table Like	,
Stor	Search Table Data	
Fun(Table Data Import Wizard	
 sakila sys 	Refresh All	

Görsel 6.33: Tables'a sağ tıklanarak Create Table seçimi

	VERİ TABANI İŞLEMLERİ
GRENME BIRIMI	

SQL kodları çalışacak olsa da arayüz sayesinde MySQL bütün kodları otomatik olarak yazmaktadır. Tek fark, işlemlerin tıklanarak yapılmasıdır (Görsel 6.34).

MySQL Workbench															
Local instance MySQL80 ×															
File Edit View Query Database	Server To	ols Scripting He	elp												
	jo id 🔥	9													
Navigator		new_table - Table	×												
SCHEMAS 🚸		Table Name	new table											Schama	aakul
Q Filter objects	2	Table Name.												Scriettia.	COKUI
V 📄 eokul		Charset/Collation:	Default Charset					~	Defa	ult Col	lation		~	Engine:	InnoDB
Tables															
Views		Comments:													
Stored Procedures					_	_						1			
Sakila	Column Name		Datatype	PK	NN		В		ZF		G	Default/Expression			
▶ 🗐 sys															
▶ 🤤 world															

Görsel 6.34: Tablo oluşturma ekranı

Arayüzdeki kısaltmaların açılımları şöyle sıralanabilir:

- PK: Eklenen satırın primary key olup olmamasının seçildiği kutucuktur.
- NN: Not Null, eklenen satırın boş geçilemeyeceğini belirtir.
- BIN: Seçilen satırdaki bilgilerin Binary olarak saklanıp saklanmayacağının seçildiği alandır.
- **UN: Unsigned**, seçilen alan için sadece pozitif ekleme yapılabileceğini gösteren alandır. Örneğin, bir tinyint alan unsigned olarak işaretlendiğinde 0 ile 255 arasında bir değer alabilir.
- UQ: Seçilen alanın benzersiz olması istenildiğinde kullanılır.
- ZF: Seçilen alanda boşluk kalması hâlinde alanın "0" ile doldurulması istenildiğinde seçilmesi gere-

ken alandır. Örneğin, int(3) alanına 21 bilgisi geldiğinde bu alan 021 olarak eklenecektir.

- G: Bu alanın diğer alanlardan üretildiğini gösterir.
- AI: Auto Increment, bu alandaki değerin bir bir artacağını gösterir. Primary keyler için kullanılır.



SCHEMAS	49	Table	Name: ogrencinot	e: ogrencinot							
Filter objects eokul Tables		Charset/Co	llation: Default Charset	Default Charset V						Default Colla	
lables dersler ogrencibilgi ogrencinot		Comments	5:								
 Views Stored Procedures Functions sakila sys world 	5	Column Name P ogrencinotid ogrNoFK dersKoduFK yazili1 yazili2 yazili3 uygulama1 uygulama2 sozlu1 sozlu2 ortalama	Datatype INT INT VARCHAR(4) TINYINT TINYINT TINYINT TINYINT TINYINT TINYINT TINYINT TINYINT			B					Defa

Görsel 6.35: ogrencinot tablosunun son hâli

Görsel 6.35'teki alanlar incelendiğinde şu bilgilere ulaşılır:

- ogrencinotid alanı, bu tablodaki her bir kayıt için benzersiz bir id üretilmesini sağlayacaktır. Buradaki id'ler özel bir anlam ifade etmediği için auto increment alanı da seçilmiştir. Böylece her kayıt eklendiğinde buradaki id alanı otomatik olarak bir artacaktır. ogrencibilgi tablosunda primary key olan öğrenci numarası alanı, kullanıcı gireceği için auto increment yapılmamıştır.
- ogrNoFK ilişkisel veri tabanı için ogrencibilgi tablosuyla bağlantı kurulacak alanı temsil etmektedir.

Burası bir Foreing Key alanıdır.

dersKoduFK alanı da dersler tablosu ile bağlantı kurulacak alanı temsil etmektedir. Burası da bir

Foreing Key alanıdır.

- Ortalama alanına ise şu an kullanılmamasından dolayı "0" değeri default olarak verilmiştir.
- Diğer alanlar ise not girişi yapılacak alanlardır.

Tablo oluşturulduktan sonra aşağıda yer alan "**Apply**" düğmesine basılır ve gelen SQL komutları incelenir.





Table Na	ne: ogrencinot					Schema:	world					
Charset/Collat	on: Default Charset			 ✓ Default Collation 		Engine:	InnoDB					
Comments:												
ımn Name idogrencinot	Datatype INT	PK N	N UQ B	Apply SQL Script to Database						×		
ogrNoFK dersKoduFK yazili1	TINYINT VARCHAR(4) TINYINT			Review SQL Script Apply SQL Script	Review the SQL Script to be	Applied on	the Databa	se				
yazili2 yazili3 uygulama1	TINYINT TINYINT TINYINT				Online DDL Algorithm: Default	~	Lock Type:	Default	~			
uygulama2 sozlu1 sozlu2	TINYINT TINYINT TINYINT				1	rld`.`ogrer NOT NULL	cinot` (AUTO_INCRE	EMENT,				
ortalama	TINYINT		; 8 8		3 'ogrNoFK' TINYI 4 'dersKoduFK' VA	T NOT NUL	L, IOT NULL,					
					6 'yazili2' TINYINI 7 'yazili3' TINYINT	NOT NULL, NOT NULL,						
					8 'uygulama1' TIN 9 'uygulama2' TIN	INT NOT N	ULL, ULL,					
					10 'sozlu1' TINYINT 11 'sozlu2' TINYINT	NOT NULL, NOT NULL,						
					12 ortalama TINYI 13 PRIMARY KEY (`in	ogrencinot));					
Column Name:					14					а Туре:		
arset/Collation: Default Ch	irset)efault:		0.000
Comments:					¢					> torage:	Virtual Primary Key Rinary	Stored Not Null Unsigned
										_	Auto Tecrement	Generated

Görsel 6.36: ogrencinot tablosunun SQL komutları

Görsel 6.36'daki "Apply" düğmesine basıldıktan sonra aşağıdaki mesajın alınması gerekir (Görsel 6.37).

SQL script was successfully applied to the database.

Görsel 6.37: Kodların düzgün çalıştığını belirten mesaj

Bu mesaj gelmediyse "**Output**" penceresinde hatanın sebebi görülebilir. Bu hata internetten araştırılıp çözüm üretilebilir.

6.5. TABLOLARI İLİŞKİLENDİRME

İçinde veri olmayan üç adet tablo oluşturuldu. Bu tabloların gerekli alanları birbirine bağlanarak aralarında anlamlı bir ilişki oluşturulmalıdır.

6.5.1. İlişkisel Veri Tabanları

Tüm modern veri tabanı yönetim sistemleri, ilişkisel veri tabanı yönetim modelini kullanarak bilgi saklar ve işler. İlişkisel terimi, veri tabanındaki her kaydın bir tek konu hakkında ve yalnızca o konuyla ilişkili bilgileri içerdiği gerçeğinden ortaya çıkar.

İlişkisel veri tabanı yönetim sisteminde (**RDBMS-Relational Database Management System)** tüm veriler tablolar içinde yönetilir. Ayrıca birden çok tablo ya da sorgudan alınan ilişkili değerler birleştirilebilir. Örneğin; hangi yazarların, hangi kitapları yazdığını öğrenmek için yazar ve kitap bilgileri veya hangi satış elemanının prim hak ettiğini bulmak için çalışan ve sözleşme bilgileri birleştirilebilir.

İlişkisel veri tabanı modelinin amacı, yüksek verimliliktir. Örneğin; personel bilgileri bir veri tabanında tutulduğunda personel adı, adresi ve telefonu personelin çalıştığı farklı bölümler için çoğaltılacaktır. Bu gereksiz fazlalık, önemli sayıda kayıtla uğraşan veri tabanları söz konusu olduğunda hem artan disk boşluğu ihtiyacı hem de artan veri ulaşım zamanı sorunlarına yol açar. Veri, ilişkisel modelde iki tabloya ayrılacaktır. Birinci tablo sigorta numarası, kimlik numarası, adres, telefon, yaş gibi kişisel bilgileri içerir. İkinci tablo ise bu bilgileri çoğaltmak yerine personelin çalıştığı bölüm bilgilerini içerir. Bu durum, normalizasyon ile doğrudan ilişkilidir.

• MySQL'de İlişkisel Veri Tabanı Oluşturmak (Reverse Engineer)

MySQL'de veri tabanları arasında ilişki kurmak için "Database \longrightarrow Reverse Engineer" menüsünden diyagram alanına tabloların aktarımı yapılmalıdır (Görsel 6.38). Görsel 6.39'da eokul veri tabanı seçilmelidir. Görsel 6.40'taki ekranda ise istenen tablolar seçilmelidir.





File	Edit	View	Query	Database	Server	Tools	Scripting	Help		
SQL	SQL		8.0	Connect to Database Ctrl+U						
Navi	gator			Manage Connections						
SCHEMAS				Rever	Reverse Engineer Ctrl-					
Q F	Q Filter objects			Schema Transfer Wizard						
	v 🖹 eokul				Migration Wizard					
	▼ 🖶 Tables			Edit Type Mappings for Generic Migration						
		ogren ogren	ncibilgi ncinot	Searc	Search Table Data					

•

Görsel 6.38: Reverse Engineer seçimi



Görsel 6.39: eokul veri tabanı seçimi



Görsel 6.40: Veri tabanından istenen tabloların seçimi

.

•



Görsel 6.41: Diyagram alanına eklenen tabloların son hâli

Görsel 6.41'deki tablolar, diyagram alanına düzgün bir şekilde aktarılmıştır. Tabloların sol tarafında alan ilişki türleri verilmiştir.



Görsel 6.42: Tablolar arasındaki ilişki türleri

Veri tabanları arasındaki bağlantılar; bire çok (one to many), bire bir (one to one), çoka çok (many to many) olmak üzere üç şekilde sağlanmaktadır (Görsel 6.42).

• Bire Bir Bağlantı Türü (1:1): Her iki tablodaki bağlantılı alan, her iki tabloda da benzersiz olursa

bire bir bağlantı türü kullanılmıştır. Örneğin, öğrenci bilgi tablosundaki öğrenci numarası ile mezuniyet tablosundaki öğrenci numarası arasında 1:1 bağlantı vardır çünkü okuldan o numara ile mezun olabilecek 1 öğrenci vardır ve numara alanları bağlandığında ekrana 1 tane kayıt gelecektir.

• Bire Çok Bağlantı Türü (1:n): Bir tabloda primary key olan alanın başka bir tabloda birden çok

tekrar etmesi ile ortaya çıkan bağlantı türüdür. Örneğin, öğrenci bilgi tablosundaki öğrenci numarası alanı ile öğrenci not tablosundaki öğrenci numarası alanı ilişkilendirildiğinde bir öğrencinin birden çok ders notu olacağı için öğrenci numarası, öğrenci not tablosunda birden çok tekrarlanacaktır.

• Çoka Çok Bağlantı Türü (n:m): İki tablo arasında yapılan bağlantı birden çok kez tekrarlanmışsa

bu ilişki türü kullanılmıştır. Örneğin, bir film veri tabanı ve bu veri tabanında film ile kategori tabloları olsun. Filmin kategorileri ilişkilendirilmek istenildiğinde bir filmin birden çok kategorisi olabileceği için kategori tablosunda hem film id'si birkaç kez görülebilir hem de o kategoride başka filmler bulunabileceği için kategori



VERİ TABANI İŞLEMLERİ



id'si de aynı tabloda birden çok tekrar edebilir (Görsel 6.43).

	film_id	category_id
	58	6
	58	9
	59	3
	60	4
	61	7
	62	6
	63	8
	64	7
	65	11
	66	1
•	66	9
	68	3
	69	14
	70	2

Görsel 6.43: Çoka çok bağlantı türü

Tabloların kesik ve düz çizgi olarak görülmelerinin sebebi, aralarındaki bilginin **tutarlı olup olmamasıyla** ilgilidir. Tablonun bir alanındaki verinin diğer tabloda yer alıp almayacağına karar verilerek bağlantı türü seçilmelidir. Tablonun bir alanındaki verinin diğer tabloda kesinlikle olması gerekiyorsa düz çizgi, verinin diğer tabloda olup olmaması önemli değilse de kesik çizgi seçilmelidir. Örneğin, filmlerle ilgili veri tabanı film tablosunda mutlaka bir yönetmen alanı ve yönetmen bilgilerini tutan bir tane de yönetmen tablosu olacaktır. Tabloya film eklendiğinde yönetmen id eklenmesi istenirse düz çizgili bağlantı, yönetmen bilgisinin sonradan eklenmesinde bir sakınca görülmezse de kesik çizgi bağlantısı seçilir.

6.5.2. Tablolar Arası Bağlantı Yapılması

Tablolar hazırlanırken ogrencinot tablosunda gerekli foreing key alanları oluşturulmuştu. O yüzden bağlantı kısmında en altta yer alan ve istenilen alanlar arasındaki bağlantıyı kullanıcının yapmasına olanak

sağlayan bire çok simgesiyle bağlantılar gerçekleştirilir. Bunun için simgeye tıklanarak önce ogrencinot tablosundaki ogrNoFK alanına sonra da ogrencibilgi alanındaki ogrenciNo alanına gelinir. Ardından yine simgeye tıklanarak önce ogrencinot tablosundaki dersKoduFK alanına sonra da dersler tablosundaki derskodu alanına gelinir.



Görsel 6.44: Tablolar arası bağlantı şeması



			-		
VERI	TAB	ANI	IŞL	EMI	.ERI



Görsel 6.44'te ogrencibilgi ve dersler tablolarına gelen uçlar tekli, ogrencinot tablosuna gelen uçlar ise çoklu olmuştur. ogrencinot tablosundaki ogrNoFK ve dersKoduFK alanları tekrar edecek şekilde (**Fore-ing Key**) yapılandırılmış ve ilişkilendirilmiştir.

Bağlaı		· · · · · · · · · · · · · · · · · · ·		⇒lusturulmustur. Bu
Bağlaı bağlantı türü ı lama hesaplar	ogrencibilgi ogrencibilgi ogrenciAdi VARCHAR(20) ogrenciSoyadi VARCHAR(20) ogrenciSoyadi VARCHAR(30) ogrenciSinfi TINYINT ogrencidogumtarihi DATE Indexes	grencinot og	(4)	i girilemez ve orta- lır (Görsel 6.45).
Cardinality One-to-One (1:1) One-to-Many (1:r)	Referenced Table ogrencibilgi ogrenciNo: INT (PK)			
Invert Relationship	Mandatory Edit			

Görsel 6.45: İlişki türünü değiştirme işlemi

Bunun için oluşturulan bağlantıya çift tiklandıktan sonra aşağı tarafta "**Relationship**" penceresinde "**Foreign Key**" sekmesine geçilir. Daha sonra "**Identifying Relationship**" kutucuğu seçilir. Böylece ogrencibilgi tablosunun ogrencino alanında olmayan bir kayıt, ogrencinot tablosuna eklenemeyecektir. Bu da veri tabanı için istenilen bir durumdur (Görsel 6.46).

ogrencibilgi ogrencikligi ogrencikligi ogrencikligi ogrencikligi ogrencikligi	ogrencinoti o





DİKKAT: Bazı durumlarda ekrana veri türleri (INT gibi) gelmemektedir (Görsel 6.41). Gelmemişse tablo başlıklarına çift tıkladıktan sonra aşağıda açılan pencereden tekrar veri türü tanımlanması gerekmektedir.

Yapılan değişiklikleri veri tabanına uygulamak için "**Database**" menüsünden "**Synchronize Model**" seçeneği işaretlenir (Görsel 6.47).

<u>D</u> atabase	Tools	Scripting	<u>H</u> elp							
Conne	Connect to Database Ctrl+L									
Manag	ge Conne	ctions								
Rever	se Engin	eer	Ctrl+R							
Forwa	ard Engine	eer	Ctrl+G							
Scher	ema Transfer Wizard									
Migra	gration Wizard									
Edit T	Edit Type Mappings for Generic Migration									
Synch	nronize M	odel	Ctrl+Shift+Z							
Synch	Synchronize with Any Source Ctrl+Shift+Y									
Comp	are Sche	mas								

Görsel 6.47: Synchronize Model seçimi

Görsel 6.48'deki ekranda senkronize yapılacak veri tabanı seçilir. Görsel 6.49'daki ekranda yapılacak değişiklikler gösterilir. Görsel 6.50'deki ekranda yapılacak değişiklikler SQL kodlarıyla gösterilir. Görsel 6.51'deki ekran ile işlemlerin düzgün sonlandığı görülür.

Synchronize Model with Database				×
Connection Options	Select the Schemas to	be Synchronized		
Sync Options		Su responsere · Instan and Subscription		
Connect to DBMS		ata ta ha Synchronizadi		
Select Schemas		ata to be Synchronized.		
Retrieve Objects	Model Schema	RDBMS Schema		
Select Changes to Apply	eokul	eokul	schema not found in target	
Review DB Changes				
Synchronize Progress				
	Override target scheme to b	e synchronized with		V Override Target
				overlide ruiget
	If you are creating them for	the first time use the Forward Engir	neer function.	
			Back	Next Cancel

Görsel 6.48: Synchronize Model adımları



VERİ TABANI İŞLEMLERİ

Synchronize Model with Database					
Connection Options	Model and Database Differences				
Sync Options					
Connect to DBMS	Double click arrows in the list to choose whether to ignore changes, update the model with database changes or vice-versa. You can also				
Select Schemas	apply an action to multiple selected rows.				
Retrieve Objects	Model Update Source				
Select Changes to Apply	dersler dersler				
Review DB Changes	ogrencing ogrencing ogrencing				
Synchronize Progress					
	Update Model Ignore Update Source Table Mapping Column Mapping				

Görsel 6.49: Synchronize Model adımları



Görsel 6.50: Synchronize Model SQL kodları

Back Next Cancel





Synchronize Model with Database		>
Connection Options	Progress of Model and Database Synchronization	
Sync Options		
Connect to DBMS	The following tasks will now be executed. Please monitor the execution.	
Select Schemas	Press Show Logs to see the execution logs.	
	Apply Changes to Database	
Select Changes to Apply	Read Back Changes Made by Server Annu Changes to Model	
Review DB Changes	 Apply changes to Hodel 	
Synchronize Progress	Synchronization Completed Successfully	
	Show Logs Back Close Cancel	

Görsel 6.51: Senkronizasyonun sorunsuz uygulanması

6.6. VERİ TABANINA BİLGİ GİRİŞİ

Tablolar oluşturulduğuna ve ilişkileri yapıldığına göre tablolara veri girişine başlanabilir. Tablolara SQL kodları veya MySQL arayüzü kullanılarak iki şekilde veri girişi yapılabilir. Görsel 6.52'de görüldüğü gibi dersler, SQL kodlarıyla dersler tablosuna eklenir:

insert into tabloadı (alan1, alan2, ..., alan(n)) VALUES (deger1, deger2, ..., deger(n))

insert into dersler(derskodu	dersadi
derskodu,		Gerakoud	derbuda
dersadi	•	BDN	Beden Eğitimi
) values		BTT	Bilişim Teknoloji Temelleri
("BTT", "Bilişim Teknoloji Temelleri"),		EDE	Edebiyat
("PT", "Programlama Temelleri"),		FİZ	Fizik
("NESN", "Nesne Tabanlı Programlama"),		MAT	Matematik
("MAT", "Matematik"),		NESN	Nesne Tabanlı Programlama
("EDE", "Edebiyat"),		PT	Programlama Temelleri
("FİZ", "Fizik"),		NULL	NULL
("BDN", "Beden Eğitimi")			

Görsel 6.52: Dersler tablosuna ders ekleme

MySQL arayüzü kullanılarak ogrencibilgi tablosuna veri girişi yapmak için tabloya sağ tıklanıp "Select Rows – Limit 100" seçimi yapılır ya da tablonun üzerine gelindiğinde çıkan simgelerden en sağdaki tıklanır (Görsel 6.53).







MySQL Workbench								
Least instance MySO	00 ~							
Local Instance MySQI				1 0 1 4				
e Edit View Query	Database	в :	Server Io	ols Scriptin	g Help			
	6	ď	Ö Ö	2				
avigator		Que	ery 1 S	QL File 3*	ogrencibilgi \times			
HEMAS	•	C	8 🖗	F Q	🕑 🚯 ⊘ (🔊 🔞 Limit to 10)00 rows 👻	🔆 🛷 🔍
Filter objects			1.0		M aakul agnang	ibilai.		
eokul ▼ Tables > dersler > ogrencibilgi > ogrencinot ♥ Views ♥ Stored Procedures ■ Functions ■ sakila ■ sys ♥ world	900 1							
		Re	sult Grid	Eilter	Rows:	Edit: 🖌	Expe	ort/Import:
		111		o gron ci A di	ograpaiSavadi	agrong Rolumu		
			2245	Upydor	DAČ	Caratacilik	10	2002.01.01
			3455	Reste	SAHİNOĞULUAPT	Vivecek İcecek	0	2002-01-01
			3456	Melike	ASIAN	Yivecek İcecek	9	2003-07-22
			3457	Abmet	KOSUCU	Bilisim Teknolojileri	11	2003-07-22
			4324	İdil	ALTIN	Bilisim Teknolojileri	12	2000-09-08
			4544	Avca	DAI	Bilisim Teknolojileri	10	2000-09-00
			4566	Yakun	KITTII	Bilisim Teknolojileri	12	2002-06-04
			4567	Sedat	KÖSE	Vivecek İcecek	10	2002-00-01
			4756	Övkü	KTLINC	Bilisim Teknolojileri	12	2000-04-13
			5456	Sevcan	KUDBET	Gazetecilik	9	2003-12-23
			6554	Nurdeniz	SOVILI	Gazetecilik	12	2000-08-27
			6788	Mehmet	DENÍZ	Viverek İrerek	0	2000-00-27
			7945	Gizem	EIDINCI	Cazetecilik	12	2003-12-17
			7050	tai	LAL	Bilisim Taknalajilari	12	2000-07-23
			0745	Zorda	DENÍZ	Caracterilik	12	2000-11-02
			0/45	Zeroa	TEOMAN	Bilisim Televeleiteri	12	2001-05-17
			NULL	NULL	NULL	NULL	NULL	2000-01-17 NULL

Görsel 6.53: MySQL arayüzü ile veri girişi yapılması

Görsel 6.53'teki gibi öğrenci numarasını kullanıcı gireceği için auto increment seçilmemesi doğru bir karardır. Veri girişi bittikten sonra aşağıdaki "**Apply**" düğmesine basılır. Bilgiler ogrencinot tablosuna girilerek "**Apply**" düğmesine basılır (Görsel 6.54).

ogrencinot \times	dersler	ogrencibilgi								
	60	0 6 6	0	🗑 Li	mit to 100	rows	• 🙀 🗹 🔍 🕅	1 🗊		
1 • SE	FCT * FROM	eokul.ogo	encino	+:						
	ceer rao	construgi	encino	• • •						
							Analy COL Cariates	Detelors		
/							Apply Sqt Script to	Database	8	
		_					Review SQL Scrip		Review the SQL Script to be Applied on the Database	
Result Grid	🕴 📢 Filter I	Rows:		Edit	· 🝊 🖽	6 🖦 🛛	Apply SQL Script			
ogrenciNot	Id ogrNoFK	dersKoduFK	yazii 1	yazili 2	2 yazili3	uygula	m			A Result
NULL	4544	BTT	14	56	66	56			1 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili1', 'yazili: A	Grid
MULL	4544	EDE	55	68	95	65			2 INSERT INTO `eokul`.`ogrencinot` (`ogrNoFK`, `dersKoduFK`, `yazili1`, `yazili1`, `yazili1`,	
MULL	4567	MAT	21	65	32	74			3 INSERT INTO `eokul`.`ogrencinot` (`ogrNoFK`, `dersKoduFK`, `yazili`, `yazili`	
NULL	4567	BDN	85	55	65	47			4 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili1', 'y	Form
NULL	4567	FIZ	88	58	96	31			5 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
NULL	4756	BTT	54	65	32	87			6 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili1', 'y	
HULL	4756	EDE	25	98	65	74			7 INSERT INTO 'eokul'.' ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	Field
1000	4756	FIZ	54	87	98	65			8 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili1', 'y	Types
NULL	5456	BDN	54	65	65	32			9 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	E 6
NULL	5456	MAT	21	32	90	52			10 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
HULL	6554	NAT	32	65	98	75			11 INSERT INTO 'eokul', 'ggrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	Query Stats
HULL	6554	EDE	25	23	66	38			12 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
MULL	6554	BDN	85	85	95	87			13 INSERT INTO 'eokul', 'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
NULL	6788	FİZ	32	32	20	25			14 INSERT INTO 'eokul', 'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	Execution
5500	6788	MAT	65	32	47	52			15 INSERT INTO 'eokul'.'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	Plan
1000	6788	EDE	65	74	85	65			16 INSERT INTO eokul', 'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
MULL	7845	FI7	1/	85	32	90			17 INSERT INTO 'eokul', 'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'yazili', 'yazili'	
NULL	7845	BDN	57	85	69	85	11		18 INSERT INTO eokul . ogrencinot (ogrNork , dersKoduFK', yazili', yazili')	
NULL	7858	NESN	65	65	32	56			19 INSEKI INIO eokul, ogrencinot (ogrivork,) derskoduPk (yazili) yazili	
NULL	7858	BTT	85	85	99	98			ZII INSEKTINTI ANTI ANTI OTROCIOTI OTROCEK GECKONIEK VATILI VATILI S	
NULL	7858	FIZ	52	63	65	85			S.	
2233	8745	FIZ	54	21	65	98				
NULL	8745	BDN	14	58	65	98				
NULL	8/45	MAT	59	05	32	09	74/11			
MULL	9999	NESN	100	95	95	99			teack Apply Cancel	
► MILL	9999	BTT	98	95	100	95	85 98	90		
· NULL	HULL	NULL	NULL	NULL	NULL	NULL	HULL NUL	NULL		×
ogrencinot 1 ×										Apply Revert

Görsel 6.54: Apply düğmesine basıldıktan sonra gelen ekran



ÖĞRENME BİRİMİ

MySQL, kodları **otomatik** olarak oluşturdu. **"Apply**" düğmesine basılarak bir hata yoksa veri girişi tamamlanır. Bu özellik, veri tabanı yönetim sistemlerinin ne kadar önemli olduğunu göstermektedir.

ogrencinot ×	dersler	ogrencibilgi									
	× × Q	0 190 10	0	🐻 Lin	nit to 100	rows .	• 🙀 🛷 🔍	1 1			
1. 5	ELECT * FRO	M eokul.ogr	encino	ti							
				- /							
							Apply SQL Script	to Database			
<							Review SQL Script				>
Result Grid	Fiter	Rows:		Edits	A 18	6 116 E			Applying SQL script to the database		
ogrenciNg	tid ogrNoFK	dersKoduEK	vazii 1	vazii2	vazii3	uvoulan	Apply SQL Scrip			~	
NULL	4544	BTT	14	56	66	56			The following tasks will now be executed. Please monitor the execution.		Result
NULL	4544	EDE	55	68	95	65			Press Show Logs to see the execution logs.		
HULL	4544	PT	54	65	65	21			Evenite SQL Statements		
HULL	4567	MAT	21	65	32	74			G means ste summers		Form
NOCC.	4567	BDN	85	55	65	47			Error: There was an error while applying the SQL script to the database.		Editor
100.00	4567	FIZ	88	58	96	31					
HULL	4/56	BII	54	65	32	8/			Message Log		
NULL	4756	EDC FTZ	54	87	98	65					Field
NULL	5456	BDN	54	65	65	32			(89); INSERTINTO 'eokul', 'ogrencinot' ('ogrNoFK', 'dersKoduFK', 'vazili'		
NULL	5456	EDE	56	32	98	65			uygulama1', 'uygulama2', 'sozlu1', 'sozlu2') VALUES (9999', 'BTT', '98', '95', '100', '95', '85', '98',		
NULL	5456	MAT	21	32	74	52			90); UPDATE 'eokul'.'ogrencingt' SFT 'dersKoduFK' = 'MAT WHERE ('ogrenciblatid' = '1) and ('ogrboFK')		Query
HOLD	6554	NAT	32	65	98	75			= '2345') and ('dersKoduFK' = 'BTT');		Stats
Ritte	6554	EDE	25	23	66	38			UPDATE 'eokul': ogrenanot' SET 'derskoduFK' = 'EDE' WHERE ('ogrenciNotid' = '2) and ('ogrNoFK' = '235') and ('eqrskoduFK' = '9'T):		
NULL	6554	BDN	85	85	95	87			UPDATE 'eokul', 'ogrencinot' SET 'dersKoduPK' = 'FIZ' WHERE ('ogrenciNotId' = '5) and ('ogrNoFK'		
NULL	6788	MAT	52	32	47	52			= '3455') and ('dersKoduFK' = NESN');		Execution
HULL	6788	EDE	65	74	85	65			ERROR 1452: 1452: Cannot add or update a child row: a foreign key constraint fails		
NULL	7845	MAT	17	85	32	65			('eokul', 'ogrencinot', CONSTRAINT' 'K_ogrencinot_dersler1' FOREIGN KEY ('dersKoduFK') RFFERVERS' ('dersk'', 'derskodu'))		
NULL	7845	FİZ	85	85	90	90			SQL Statement:		
NOLE	7845	BDN	57	85	69	85			INSERTINTO eokul: ogrencinot (ogrNoFK; dersKoduFK; yazili; yazili; yazili; 'uvgulama1', uvgulama2', isodul', sodul; Yazili; PES (6554, Nati, 32, 165, 168, 75, 198, 156).		
POLICE	7858	NESN	65	65	32	56			'66') '66')		
NULL	7858	BIT sta	85	62	99	98			r		
NULL	8745	FTZ	54	21	65	98			1		
NULL	8745	BDN	14	58	65	98					
NULL	8745	EDE	54	65	32	65					
HULL	9999	MAT	90	95	95	98			Hide Logs Back Finish Cancel		
HULL	9999	NESN	100	95	95	99	1				
> E0000	9999	BTT	98	95	100	95	85 9	90			
* mild	mill	MARKAN PARTY IN CONTRACTOR	ELLIN C	maddin .	mall ³	Eann	tanan E			× · · · ·	
ogrencinot 1	×									Apply	Revert

Görsel 6.55: Veri ekleme hatası

"Apply" düğmesine basıldıktan sonra bir hata varsa Görsel 6.55'teki ekran ile karşılaşılır. **"ERROR**" kısmında hatanın sebebi görülebilir ve bu hata düzeltilebilir. Burada gösterilen hata, bir foreign key hatasıdır. Bu da tablolar arasında yapılan ilişkilendirmelerle ilgilidir. İlişkiler oluşturulurken **ogrencinot** tablosundaki **ogrencino** alanı **ogrencibilgi** alanındaki **ogrencino** ile, **ogrencinot** tablosundaki **derskodu** alanı da dersler tablosundaki dersler ile ilişkilendirilmiş ve "**Identifying Relationship**" özelliği seçilmişti.

Hatanın sebebi budur. Tabloya bakıldığında 6554 numaralı öğrenci için yapılan girişte ders adı "**MAT**" yerine "**NAT**" yazılmıştır. Dersler tablosunda böyle bir kayıt bulunmadığı ve ilişkilendirmeden dolayı dersler tablosunda olmayan bir giriş yapılmaya çalışıldığı için veri tabanı bu girişe izin vermemiştir. Bu ilişkilendirme yapılmasaydı ve "NAT" ders olarak eklenseydi ortalama hesaplanmasına kadar birçok yerde ciddi hatalara yol açabilirdi. Aynı hata, **ogrencibilgi** tablosunda yer alan **ogrencino**'lar dışında bir ogrencino eklenirse de görülür çünkü ogrencibilgi tablosunda yer almayan bir ogrencino'sunun anlamı yoktur. Hata düzeltilerek veri tabanına tekrar kayıt eklenir.





		-	
VERI	TABANI	IŞLEM	ILERI

Pa													
MySQL Workbench													
Local instance	MySQL80 ×												
File Edit View Q	uery Databas	se Server Tools	Scripting	Help									
	. .	6 IQ 🖓											
Navigator		ogrencinot \times											
SCHEMAS	6)	🗀 🖬 🖗 🖗 🧔 💿 🚯 💿 💿 🐻 Limit to 100 rows 🔹 🤸 🕩 Q. 🖺 🖘											
Q Filter objects											-		
T R askul		1 • FELECT * FROM eokul.ogrencinot;											
Tables Tables Gerencib Gerencib Views Stored Proce	lgi ot dures												
Functions		<											
 sakila sys 		Result Grid	🚯 Filter F	lows:		Edit:	1	Export	/Import:	🔯 w	ap Cell Co	intent: TA	
world		ogrenciNotId	ogrNoFK	dersKoduFK	yazili 1	yazili2	yazili3	uygulama 1	uygulama2	sozlu1	sozlu2	ortalama	
		▶ 1	2345	MAT	87	85	47	95	87	41	58	0	
		2	2345	EDE	25	36	47	95	74	12	55	0	
		3	2345	FİZ	47	85	24	99	87	58	74	0	
		4	3455	MAT	55	87	34	68	74	98	52	0	
		5	3455	FIZ	99	85	74	68	52	47	85	0	
		6	3455	EDE	14	25	66	87	25	36	99	0	
		46	3456	MAT	87	56	52	66	98	78	95	0	
		47	3456	EDE	95	65	32	85	69	98	74	0	
		48	3456	FIZ	54	65	37	98	77	65	32	0	
Administration Schar		49	3457	MAT	14	58	63	33	25	4/	65	0	
Administration Scien	10.5	50	3457	PI	25	05	32	07	32	/4	85	0	
Information		51	4224	INESIN ETZ	07	00	00	0/	90	95	95	0	
		52	4374	BTT	74	54	65	50	74	90	95	0	
Table: ogrencino	t	54	4324	BDN	59	65	32	47	85	99	87	0	
		55	4544	BTT	14	56	66	56	98	98	74	0	
Columns:	int AT	56	4544	EDE	55	68	95	65	12	47	85	0	
<u>ogrenciNotId</u>	DK NI	57	4544	PT	54	65	65	21	65	87	65	0	
ogrNoFK	int PK	58	4567	MAT	21	65	32	74	85	87	85	0	
dersKoduFK	varchar(4	59	4567	BDN	85	55	65	47	35	65	74	0	
vazili1	PK	60	4567	FİZ	88	58	96	31	65	47	75	0	
yazili2	tinyint	61	4756	BTT	54	65	32	87	99	54	75	0	
yazili3	tinyint	62	4756	EDE	25	98	65	74	85	65	32	0	
uygulama1	tinyint	63	4756	FİZ	54	87	98	65	32	78	85	0	
sozlu1	tinyint	64	5456	BDN	54	65	65	32	74	85	24	0	
sozlu2	tinyint	65	5456	EDE	56	32	98	65	74	21	65	0	
ortalama	tinyint	66	5456	MAT	21	32	74	52	85	98	96	0	
		6/	6554	MAI	32	05	98	75	96	56	66	0	
		68	6554	EDE	25	23	05	38	65	90	50	0	
		70	6799	EL/N	22	32	20	25	20	60	52	0	
		ogransingt 1 st	5766	. 12	32	32	20	23	70	00	50	v	
		ogrencinot 1 X											

SQL script was successfully applied to the database.

Görsel 6.56: Hatasız veri ekleme ekranı ve mesajı

6554 numaralı öğrencinin ders kodu "**MAT**" yapılınca diğer alanlarda hata olmadığı için tabloya sorunsuzca veri eklenmiştir. Tablo incelendiğinde "**ogrenciNotId**" alanının otomatik olarak arttığı görülür. ogrenciNotId'isinin 6'dan 46'ya atlamasının birkaç sebebi bulunabilir. Örneğin, bu aradaki öğrenciler silinmiş olabilir. 46'nın devamında ogrenciNotId'isi bir bir artmaya devam etmiştir. Ortalama alanı ise default değeri olan "0" ile doldurulmuştur.

Sıra Sizde >

Siz de 15 öğrenci için 3 dersten birer not girişi yapınız. ogrencinot tablonuzda 45 kayıt olsun. İçinde veri bulunan üç tablolu bir veri tabanına sahipsiniz. En çok kullanılan SQL komutlarını tablolarınıza uygulayınız.

6.7. SQL KOMUTLARI KULLANIMI

Önce simgesinden yeni bir SQL sayfası açılır. Tablo isimleri belirtilirken önceden oluşturulan tablo isimleri kullanılır. Her bir SQL sorgusu aksi belirtilmedikçe boş bir SQL sayfasında çalıştırılır. Aynı say-

fada birden çok SQL komutu hata almaya neden olabilir. Soldaki simge bütün komutları çalıştırmak için, sağdaki simge ise sadece imlecin bulunduğu kodu çalıştırmak içindir. Tıpkı kod editöründe olduğu gibi otomatik kod tamamlama özelliği vardır. SQL kodları yazılırken String'ler için tek tırnak işareti kullanılır.

6.7.1. Select Deyimi

SELECT {*, SÜTUN, ...} FROM tabloAdı

SQL' de sorgulama işlemleri Select deyimi kullanılarak yerine getirilir. Tablonun tüm sütunlarını seçmek için * (yıldız) kullanılır. Tabloda istenilen sütunların seçilmesi için sütunların isimleri belirtilmelidir. ogrenci tablosundaki tüm sütunları seçmek için Select deyiminin kullanımı Görsel 6.57'de verilmiştir. Tablodaki isimler rastgele seçilmiştir.

Qu	ery 1 \times						
6	8 1 🗲	f 🖗	0 🔂 📀 (🔉 🔞 Limit to	100 rows	• 🎭 🗹 🔍 🛽	
	1 • s	elect	* from	ogren	cibile	≠i	
			11.01	ogi cin		5-	
Re	esult Grid	🛛 🚷 Filter	Rows:	Edit:	B	Export/Import: 📳 🐻	w
	ogrenciNo	ogrenciAdi	ogrenciSoyadi	ogrenciBolumu	ogrenciSinifi	ogrencidogumtarihi	
•	2345	Haydar	DAĞ	Gazetecilik	10	2002-01-01	
	3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	2003-11-22	
	3456	Melike	ASLAN	Yiyecek İçecek	9	2003-07-22	
	3457	Ahmet	KOŞUCU	Bilişim	11	2001-09-22	
	4324	İdil	ALTIN	Bilişim	12	2000-09-08	
	4544	Ayça	DAL	Bilişim	10	2002-12-12	
	4567	Sedat	KÖSE	Yiyecek İçecek	10	2002-12-12	
	4756	Öykü	KILINÇ	Bilişim	12	2000-04-13	
	5456	Sevcan	KUDRET	Gazetecilik	9	2003-12-23	
	6554	Nurdeniz	SOYLU	Gazetecilik	12	2000-08-27	
	6788	Mehmet	DENÍZ	Yiyecek İçecek	9	2003-12-17	
	7845	Gizem	FIRINCI	Gazetecilik	12	2000-07-23	
	7858	İdil	LAL	Bilişim	12	2000-11-02	
	8745	7erda	DENÍZ	Gazetecilik	11	2001 05 17	
	07.10	20100	DENTE	Ourcecuik	**	2001-05-17	

Görsel 6.57: Select * From sorgusunun çalıştırılması

Görsel 6.58'de ogrenci tablosundan sadece öğrenci no, ad ve sınıf bilgilerini seçmek için **Select** deyiminin kullanımı verilmiştir.

1 • select ogrenciNo, ogrenciAdi, ogrenciSinifi from ogrencibilgi

c			
R	sult Grid	Filter	Rows:
	ogrenciNo	ogrenciAdi	ogrenciSinifi
Þ	2345	Haydar	10
	3455	Beste	9
	3456	Melike	9
	3457	Ahmet	11
	4324	İdil	12
	4544	Ауçа	10
	4567	Sedat	10
	4756	Öykü	12
	5456	Sevcan	9
	6554	Nurdeniz	12
	6788	Mehmet	9
	7845	Gizem	12
	7858	İdil	12
	8745	Zerda	11
	9999	Murat	12

Görsel 6.58: Select sorgusu ile belirli alanları çağırma

Sütunlarda takma isim kullanılması ve başlıkların alan adları dışında bir isimle görüntülenmesi için AS anahtar kelimesi kullanılır (Görsel 6.59).





1• select ogrenciNo as "Öğrenci Numarası", ogrenciAdi as "Öğrenci Adı" from ogrencibilgi

<		
R	esult Grid 📗	Filter Rows:
	Öğrenci Numarası	Öğrenci Adı
•	2345	Havdar
1	3455	Beste
	3456	Melike
	3457	Ahmet
	4324	İdil
	4544	Ayça
	4567	Sedat
	4756	Öykü
	5456	Sevcan
	6554	Nurdeniz
	6788	Mehmet
	7845	Gizem
	7858	İdil
	8745	Zerda
	9999	Murat

Görsel 6.59: Select sorgusu ile çağrılan alanları adlandırma

Select deyimi, tekrarlı satırların yalnızca bir tanesinin listelenmesi isteniyorsa **Distinct** anahtar kelimesi ile birlikte kullanılır. Aynı bölüm isimlerinden sadece birer tanesinin listelenmesi istenirse Görsel 6.60'taki gibi bir ekranla karşılaşılır.

	1•	select	distinct	ogrenciBolu	ımu from	ogrencibilgi
<						
R	esult Grid	🔢 🚷 Filt	ter Rows:	Export: 📑 Wrap	Cell Content: 1A	
	ogrenci	Bolumu				
۰.	Gazeteci	ilik				
	Vivecek	İcecek				
	inyecck.	IGECER				

Görsel 6.60: Distinct kullanımı

6.7.2. Where Şart İfadesi

Belirli bir koşulu sağlayan kayıtların süzülmesi için Where şart ifadesi kullanılır. Bütün SQL komutları içinde belki de en önemlisi ve çok kullanılanı bu ifadedir. Where ile yapılacak işlemlerin sadece belirtilen değere göre olması sağlanabilir. **Where ile bir filtreleme yapılmadığı takdirde güncelleme, silme vb. işlemler bütün tabloya uygulanacaktır.**

Görsel 6.61'de bölümü "Bilişim" olan öğrencileri listelemek için Where kullanılmıştır.

	1• s	elect	* from	ogrenci	bilgi 🛚	w <mark>here</mark> ogre	enciBolumu=	'Bilişim'
<								
R	esult Grid	🚺 🚷 Filter	Rows:	Edit:	⊿ 🖦 🖦	Export/Import:	Wrap Cell Content:	1A
	ogrenciNo	ogrenciAdi	ogrenciSoyadi	ogrenciBolumu	ogrenciSinifi	ogrencidogumtarihi		
•	3457	Ahmet	KOŞUCU	Bilişim	11	2001-09-22		
	4324	İdil	ALTIN	Bilişim	12	2000-09-08		
	4544	Ayça	DAL	Bilişim	10	2002-12-12		
	4756	Öykü	KILINÇ	Bilişim	12	2000-04-13		
	7858	İdil	LAL	Bilişim	12	2000-11-02		
	9999	Murat	TEOMAN	Bilişim	12	2000-01-17		
	NULL	NULL	NULL	NULL	NULL	NULL		

Görsel 6.61: Where kullanımı





Sütun isimleri, özellikle iki tablo ile çalışıldığında tablo ismiyle birlikte kullanılır (Görsel 6.62).

select ogrencibilgi.ogrenciAdi, ogrencibilgi.ogrenciSoyadi, ogrencinot.sozlu1, ogrencinot.sozlu2 from ogrencinot, ogrencibilgi
 where ogrencibilgi.ogrenciNo =ogrencinot.ogrNoFK

esult Grid 🛛 🚺	Filter Rows:		
ogrenciAdi	ogrenciSoyadi	sozlu1	sozlu2
Haydar	DAĞ	41	58
Haydar	DAĞ	12	55
Haydar	DAĞ	58	74
Beste	ŞAHİNOĞULLARI	98	52
Beste	ŞAHİNOĞULLARI	47	85
Beste	ŞAHİNOĞULLARI	36	99
Melike	ASLAN	78	95
Melike	ASLAN	98	74
Melike	ASLAN	65	32
Ahmet	KOŞUCU	47	65
Ahmet	KOŞUCU	74	85
Ahmet	KOŞUCU	95	95
İdi	ALTIN	98	95
İdi	ALTIN	85	95
Idil	ALTIN	99	87
Ayça	DAL	98	74
Ayça	DAL	47	85
Ayça	DAL	87	65
Sedat	KÖSE	87	85
Sedat	KÖSE	65	74
Sedat	KÖSE	47	75
Öykü	KILINÇ	54	75
Öykü	KILINÇ	65	32
Öykü	KILINÇ	78	85
Sevcan	KUDRET	85	24
Sevcan	KUDRET	21	65
Sevcan	KUDRET	98	96
Nurdeniz	SOYLU	56	66
Nurdeniz	SOYLU	90	50
All configuration	COVILI	74	52
Nurdeniz	30100		

Görsel 6.62: Where kullanarak ilişkili tablolardan veri çekme

Buradaki Select sorgusu, şu ana kadar yapılan her şeyin bir sonucudur. SQL sorgusuna bakıldığında tablo adları belirtilerek (ogrencibilgi ve ogrencinot) iki tablodan görüntülenmek istenilen alanlar çağrılmış ve bu iki tablodaki alanlar Where kısmında öğrenci numarasının eşit olmasına göre listelenmiştir. ogrencinot tablosunda öğrenci adı ve soyadı yerine öğrenci numarası mevcuttur. Daha sonra diyagramlarla tablolar öğrenci numarası üzerinden bağlanmıştır. Burada öğrenci numarası eşitliğinden öğrenci adının ve soyadının görünmesi **bu ilişkilendirmeden** kaynaklanmaktadır. ogrencinot tablosunda yer alan ogrNoFK değeri, ogrencibilgi tablosuna gidip okunan numaradaki değere karşılık gelen satırdan ad ve soyad bilgisini alıp gelmektedir.

6.7.3. Karşılaştırma Operatörleri

Karşılaştırma operatörleri iki değer arasında karşılaştırma yapmak için kullanılan operatörlerdir. Tablolardan gelen değer, sorgulara Where eklenirken bir kıstasa göre filtrelenmelidir. Karşılaştırma operatörleri istenilen verilere ulaşılmasını sağlar.

= Eşit	> Büyük	< Küçük
>= Büyük eşit	<= Küçük eşit	< > Eşit değil

Tablo 6.2: Karşılaştırma Operatörleri

Örnekler:

Select ogrenciNo, ogrenciAdi, ogrenciSoyadi from ogrencibilgi Where ogrenciNo=9999; Select * from ogrencibilgi Where ogrenciBolumu <> 'Bilişim Teknolojileri'; Select ogrenciNo, ogrenciAdi, ogrenciSoyadi from ogrencibilgi Where ogrenciSinifi>10;

6.7.4. Mantıksal Operatörler

Birden çok karşılaştırma operatörünün kullanılması durumda mantıksal operatörler kullanılır. **AND (ve)**: Her iki şartı da sağlıyorsa true döner. **OP (veva)**: Sartlardan berbançi birini sağlıyorsa true döner.

OR (veya): Şartlardan herhangi birini sağlıyorsa true döner.

Örnekler:

Select * from ogrencibilgi Where ogrencibolumu='Gazetecilik' or ogrenciSinifi>=11; Select * from ogrencinot Where dersKoduFK='MAT' and ortalama>=50.0;







6.7.5. Arama Operatörü

LIKE operatörü, karakter grubu içinde arama yapmak için kullanılır.

Öğrenci adı "Murat" olan kayıtlar:

SELECT * FROM ogrencibilgi WHERE ogrenciAdi LIKE 'Murat';

Öğrenci soyadı "S" ile başlayan kayıtlar:

SELECT * FROM ogrencibilgi WHERE ogrenciSoyadi LIKE 'S%';

Ders adında "temel" geçen kayıtlar:

SELECT * FROM dersler WHERE dersadi LIKE '%temel%';

Adının ikinci harfi "n" olan öğrenciler:

SELECT * FROM ogrencibilgi WHERE kitapadi LIKE `_n%';

6.7.6. Order By Komutu (Sıralama)

Sıralama işlemi için Select ifadesinde **ORDER BY** kullanılır. Bu ifadede ASC kelimesi kullanılırsa sıralama küçükten büyüğe doğru (A-Z), DESC kullanılırsa sıralama büyükten küçüğe doğru (Z-A) yapılır.

Veri alar ikisi de kullanılmazsa sıralama küçükten büyüğe doğru yapılır.

Öğrencileri numarasına göre sıralayan SQL ifadesi: Select * from ogrenci Order By ogrno ASC Select * from ogrenci Order By ogrno Öğrencileri not ortalamasına göre (büyükten küçüğe) sıralama: Select ogrno, ad, soyad from ogrenci Order By ortalama DESC

Sıra Sizde

Aşağıdaki 5 satıra kendi oluşturduğunuz Select ve Where içeren sorguları yazınız.

1.	
2.	
3.	
4.	
5.	

6.7.7. Insert Into Komutu (Kayıt Ekleme)

Tabloya veri ekleme işlemi INSERT INTO komutu ile gerçekleştirilir (Görsel 6.63).

insert into tabloadı (alan1, alan2, ..., alan(n)) VALUES (deger1, deger2, ..., deger(n))

```
insert into dersler(
derskodu,
dersadi
) values
("BTT", "Bilişim Teknoloji Temelleri"),
("PT", "Programlama Temelleri"),
("NESN", "Nesne Tabanlı Programlama"),
("MAT", "Matematik"),
("EDE", "Edebiyat"),
("FİZ", "Fizik"),
("BDN", "Beden Eğitimi")
```

Görsel 6.63: Insert into komutu





VERİ TABANI İŞLEMLERİ

6.7.8. Update Komutu (Kayıt Güncelleme)

Tablodaki verileri güncelleme işlemi UPDATE SQL deyimiyle gerçekleştirilir (Görsel 6.64). **UPDATE** tablo **SET** sütun1=değer1, sütun2=değer2,...**WHERE** şart

1 update ogrencinot set sozlu1 = sozlu1+5

Görsel 6.64: Tüm öğrencilerin sözlüsüne 5 puan ekleyen SQL ifadesi

MySQL, "Where" ifadesi primary key alanını sınırlandırmadığından güvenlik sebebiyle buna izin vermeyecektir çünkü burada yapılacak bir hata, 10.000 tane kayıt olsa bile bütün veri tabanını etkileyecektir. Bu bir güvenlik önlemidir. Bu güvenlik önlemi "Edit → Preferences → SQL Editor → Safe Updates" seçeneği kapatılarak kaldırılabilir.

Görsel 6.65'te ogrencinot tablosunda sozlu2 notu 0 ile 50 arasında olan öğrencilere 10 puan ekleyen SQL ifadesi verilmiştir.

1	•	up	date	ogrencinot	: set	sozlu2	=	sozlu2+10	where	sozlu2	between	0	and	50
Out D	put Action	Output	•											
•	* 11	Time 6:15:42	Action update ogrer	ncinot set sozlu2 = sozlu2+10 where	sozlu2 betweer	n 0 and 50				6	Message Frow(s) affected Rows matc	hed: 6	Changed: 6	Warnings: 0

Görsel 6.65: Update sorgusu kullanımı

Görüldüğü gibi 6 kayıt güncellenmiştir.

Görsel 6.66'da ogrencibilgi tablosunda "Bilişim" alanını "Bilişim Teknolojileri" yapan SQL kodu verilmiştir.

1 update ogrencibilgi set ogrenciBolumu='Bilişim Teknolojileri' where ogrenciBolumu='Bilişim'

Görsel 6.66: Update kullanımı

6.7.9. Delete Komutu (Kayıt Silme)

Tablodan satır silme işlemi DELETE SQL deyimiyle gerçekleştirilir.

DELETE FROM tablo WHERE şart

Öğrenci tablosunda e-posta adresi boş (Null) olan kayıtları silmek için kullanılan ifadeler aşağıda verilmiştir. DELETE FROM ogrenci

WHERE eposta IS NULL

Not: Silinecek satırın başka bir tabloyla yapılmış ilişkilendirmede yer almaması gerekmektedir yoksa hata mesajı alınır. Tablolar arası ilişkilendirme yapılırken bir tabloda olan kaydın diğer tabloya eklenemeyeceği belirtilmiştir (Bağlantı düz çizgi hâline getirilmiştir.). Başka tabloda yer alan bir kayıt silinmek istendiğinde diğer tablodaki ogrencibilgi tablosundan türetilen bilgiler boşa çıkacaktır. Kayıtlar ogrencinot tablosuna ogrencibilgi tablosundaki ogrencino'ya göre eklenmiştir. O yüzden önce ogrencinot tablosundan istenilen numaradaki kayıtlar silinmelidir.

6.7.10. Create

Veri tabanı ya da tablo oluşturmak için kullanılır. CREATE DATABASE eokul CREATE TABLE ogrencibilgi

6.7.11. Alter

Var olan bir nesne üzerinde değişiklikler yapmak için kullanılır. Veri alanı eklemek için aşağıdaki kod kullanılabilir.

ALTER TABLE personel ADD eposta VARCHAR(40) NULL

Tablolarda bir eksiklik görüldüğünde veri türleri de değiştirilebilir. Örneğin, ogrencinot tablosundaki ortalama alanının veri türü tinyint seçilmiştir fakat bu bir tam sayı değişkeni türüdür. Ortalama hesaplandığında büyük ihtimalle ondalık bir sayı olarak gelecektir. Bu ufak sorun Görsel 6.67'deki sorgu yazılarak çözülür.





ALTER TABLE ogrencinot MODIFY ortalama DECIMAL (5,2) DEFAULT 0;

Görsel 6.67: Veri türü değiştirme sorgusu

Modify komutuyla ogrencinot tablosundaki ortalama alanı, ondalık bir veri türü olan Decimal ile değiştirildi. 5 rakamı, toplam uzunluğu ve 2 rakamı ise virgülden sonraki rakamları temsil etmektedir. Aradaki değişim ogrencinot tablosundan gözlemlenebilir.

Sıra Sizde

Neden DECIMAL (5,2) ve DEFAULT kullandığınızı açıklayarak arkadaşlarınızla fikirlerinizi paylaşınız.

6.7.12. Drop

Veri tabanındaki herhangi bir nesneyi (tablo veya veri tabanı) kaldırmak için kullanılır. DROP kullanırken çok dikkatli olunmalıdır.

DROP TABLE dersler

6.8. MYSQL VERİ TABANI ALMA VE YÜKLEME

Uzun süren projelerde veri tabanlarının taşınması gerekebilir. Proje uzun süreceği için MySQL üzerinden veri tabanının nasıl alınacağı ve MySQL'e nasıl geri yükleneceği çok önemlidir.

Önce Görsel 6.68'deki Navigator penceresinin altındaki Administration sekmesine geçilir.



Görsel 6.68: Navigator penceresi

Daha sonra bu pencerede Data Export'a tıklanır.

	Local instance MySQL80 Data Export Object Selection Export Progress Tables to Export Exp Schema Bakila Bakila Bys World	Exp Schema I	Objects ler ancibilgi ancinot
Export Options			Data Export
O Export to Dump Project Folder	C: \Users \murat \Documents \dumps \Dump 20	201230	Object Selection Export Progress
Each table will be exported into a separate file. Th	is allows a selective restore, but may be slower.		Export Completed
Export to Self-Contained File	C: \Users \murat \Documents \dumps \eokulver	itabaniyedegi.sql	Status:
All selected database objects will be exported into	a single, self-contained file.		3 of 3 exported.

Görsel 6.69: Data Export penceresi

Gelen pencerede yedeği alınacak veri tabanı seçilir ve alttaki "**Export to Self-Contained File**" seçeneği işaretlendikten sonra yedeğe bir isim verilir. Daha sonra sağ alt taraftaki "**Start Export**" ile veri tabanının yedeği alınır. Veri tabanının yedeği, belirlenen isim ile "**Belgelerim/dumps**" içine aktarılır (Görsel 6.69). Yedeklenen veri tabanını geri yüklemek için **Administration** içindeki **Data Import / Restore** seçeneği işaretlenir. Gelen pencerede "**Import From Self-Contained File**" kısmından yedek seçilir.

VERİ TABANI	İŞLEMLERİ	
	• •	
Local instance MySQ.80 Data Import		Local instance MySQL80
Import Options O Import from Dump Project Folder C: \Uters \mumatbalance	Default Schema to be Imported To	Import from Disk Import Progress
Select the Dump Project Folder to import. You can do a selective restore.	Default Target Schema:	Import Completed
Import from Self-Contained File C:/Users/murat/Docu Select the SCI (dump file to import. Please note that the whole file will be import.	ments\dumps\eokulveritabaniyedegi.sql	Status: 1 of 1 imported.

Görsel 6.70: Data Import penceresi

Yedeği yüklenecek veri tabanı hazır ise aşağıdaki "**Default Target Schema:**" kısmından seçilir. Hazırda yedeği yüklenecek bir veri tabanı yoksa sağ taraftaki "**New**" düğmesi ile yeni ve boş bir veri tabanı oluşturularak "**Start Import**" ile yedek yüklenebilir (Görsel 6.70).

6.9. SQL VE NTP BAĞLANTISI

Veri tabanı, içindeki verilerle beraber kullanıma hazır hâle geldi fakat bu kayıtları insanlara veri tabanı arayüzü ile vermek mümkün değildir. E-okul düşünüldüğünde kullanıcılar için hazırlanmış bir arayüze sahip olduğu ve kişilere ait geçmişe yönelik bilgilere bu arayüz sayesinde kolayca erişildiği görülür. Oradaki arayüz olmasa ve kayıtlar veri tabanı üzerinden kullanıcılara sunulsa çok anlamlı sonuçlarla karşılaşılmayacaktır. Üstelik veri tabanı konusunda ön bilgisi olmayan biri, istediği kayıtlara ulaşamayacaktır. Buradaki ihtiyaçtan dolayı veri tabanları arayüzlerle entegre edilir. Böylece kullanıcı sadece istediği bilgiyi seçer ve bu işlemde hiç zorlanmaz.

Bu tür bir projeye başlamadan önce yapılacakların **algoritması** çıkarılmalı ya da yapılacaklar bir plana dökülmelidir. Bu projede ilişkili üç tablo vardır. Veri tabanındaki tablolar üzerinde ekleme, görüntüleme, düzenleme ve silme işlemleri (**CRUD**) yapılmak istenmektedir. Bu noktada form tasarımı tamamen kişiye özgü olabilir. Projede üç ayrı tablo çağrılıp her bir tablo için görüntüleme, ekleme, düzenleme ve silme işlemleri gerçekleştirecektir (ogrencibilgi tablosunda öğrenci bilgileri, ogrencinot tablosunda öğrencilerin notları ve dersler tablosunda dersler ile ilgili işlemler). İşlemlere başlanmadan önce arka plan için birkaç tane resim ve simge paketi indirilirse form tasarlanırken bunlar kullanılabilir.

Kod editörü ile veri tabanı bağlantısı yapmak için bir arayüz oluşturulur.

6.9.1. Form Tasarımları

Hazırlanacak form verilerin gösterilmesini, eklenmesini, silinmesini, güncellenmesini sağlayacak şekilde tasarlanmalıdır. Yeni bir proje ve boş bir form açılır (Görsel 6.71, Görsel 6.72, Görsel 6.73, Görsel 6.74).

Open recent		Get sta	arted	
Search recent (Alt+S)	<i>۹</i> -	¥	Clone a repository Get code from an online repository like GitHub or Azure DevOps	
 This month Older 		Ċ	Open a project or solution Open a local Visual Studio project or sln file	
			Open a local folder Navigate and edit code within any folder	
		* 3	Create a new project Choose a project template with code scaffolding to get started	
			Continue without code \rightarrow	
	Görsel 6.71: Kod	editörü açılış penc	eresi	





VERİ TABANI İŞLEMLERİ

Create a new pro	ject	Search for templates (Alt+S)	Clear all
Recent project templates		C# • Windows • Desktop	•
Windows Forms App (.NET Framework) Console App (.NET Framework)	C#	NUnit Test Project (.NET Core) A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS. C# Linux macOS Windows Desktop Test Web	
🖞 WPF App (.NET Framework)	C#	Windows Forms App (.NET Framework) A project for creating an application with a Windows Forms (WinForms) user interface C# Windows Desktop	
		WPF App (.NET Framework) Windows Presentation Foundation client application C# XAML Windows Desktop	
		WPF App (.NET Core) Windows Presentation Foundation client application C# XAML Windows Desktop	
		Windows Presentation Foundation custom control library	

Görsel 6.72: Yeni proje oluşturma

		-		×
Configure your new project				
Windows Forms App (.NET Framework) C# Windows Desktop				
Project name				
EOkulProjesi				
Location				
C:\Users\Murat\source\repos •				
Solution name 1				
EOkulProjesi				
Place solution and project in the same directory				
Framework				
.NET Framework 4.8				
	Back	Cre	eate	

Görsel 6.73: Yeni proje ayarları penceresi



•

• •



Görsel 6.74: Kod editörü yeni proje form ekranı

6.9.2. Form Özellikleri

Formla ilgili birkaç ayar yapılır. Bu ayarlar şunlardır:

- Formun Name özelliği eokulForm yapılır.
- Formun **Text** özelliği E-Okul Programı yapılır.
- Formun Icon kısmına internetten bulunan bir ikon dosyası (*.ico) eklenir.
- Formun StartPosition özelliği CenterScreen yapılır.
- Formun BackColor özelliğine bir renk veya BackgroundImage özelliğine bir resim eklenir.
- Formun FormBorderStyle özelliği FixedToolWindow yapılır.

Form tasarımı gözden geçirilir (Görsel 6.75).



Görsel 6.75: Form özelliklerini değiştirme

Projeye başlanmadan önce bir yol haritası çizilir (Bu yol haritasının kesin olmadığı ve istenildiğinde değiştirebileceği unutulmamalıdır.). Öncelikle yapılacak işlemler belirlenmelidir. Projede bir ana ekranda bütün bilgiler gösterilecektir. Üç tablonun birleşimi sonucu bütün bilgiler ana forma gelecek ve bu ana formun bir kısmında veri ekleme alanı olacaktır. Her bir tablo için bir alan oluşturulup bu alanda ekleme, silme, güncelleme, arama işlemleri yapılacaktır. Ana sayfa ile beraber üç tablo da düşünülürse toplamda dört tane alan olacaktır. Bu yapı istenirse dört ayrı formla istenirse de **TabControl** bileşeniyle yapılabilir.

6.9.3. TabControl Bileşeni

TabControl bileşeninin sekmeli yapısı sayesinde formu değiştirmeden sekmeler arasında geçiş yaparak işlemler gerçekleştirilir. Forma TabControl eklemek için sol taraftaki **ToolBox** penceresinin arama kutusuna TabControl yazılır. Çıkan bileşen tıpkı bir buton gibi forma sürüklenir (Görsel 6.76).





Okul Programi	×
tahPaga1 tahPaga2	

Görsel 6.76: Forma TabControl ekleme

TabControl bileşeninin özellikleri Propertis kısmından belirlenir.

Not: Burada forma tıklandıktan sonra mutlaka propertis kısmından hangi bileşende olunduğu kontrol edilmelidir çünkü TabControl, içinde TabPage1, TabPage2 gibi sekmeleri de barındırmakta dır ve bazen bunlar seçim ekranında karışabilmektedir.

- TabControl'un Name özelliği tabctrlEokul yapılır.
- TabControl'un **Font** özelliğinden Size değeri **20** yapılır ve istenirse Türkçe karakter destekleyen bir **font** seçilir.
 - TabControl'un Dock özelliği ortadaki kutu seçilerek Fill yapılır (Görsel 6.77).



Görsel 6.77: TabControl Fill özelliği

• [StoPages (Collection) kısmındaki "..." kutucuğuna basılarak iki tane daha tab eklenir ve ayarlar yapılır. Toplamda dört adet tab olacaktır (Görsel 6.78).

TabPage Collection Editor			?		
Members:		tabPage1 propertie			
0 tabPage1	+	🔡 🛃 🔎			
1 tabPage2		✓ Accessibility			
2 tabPage3	+	AccessibleDes	cript		
3 tabPage4		AccessibleNa	me		
		AccessibleRol	e Default		
		✓ Appearance	ppearance		
		BackColor	Transparent		
		Backgroundlr	nage 🔄 (none)		
		Backgroundin	nagel Tile		
		BorderStyle	None		
		Cursor	Default		
		> Font	Microsoft Sans Serif, 8.		
		ForeColor	ControlText		
		RightToLeft	No		
Add Remove		Text	tabPage1		

Görsel 6.78: TabControl ayarları penceresi

	VERİ TABANI İŞLEMLERİ
OGRENME BIRIMI	

Görsel 6.78'deki ekranda tab'ların ayarları yapılabilir.

• Her bir tab sayfasına sırasıyla "tabpageAna", "tabpageOgrenci", "tabpageNot" ve "tabpageDers" olarak name verilir.

• Bulunan resimler her bir tabpage sayfasına **BackgroundImage** olarak eklenir. Bunun için aşağıda gelen ekrandan "**Import**" seçeneği işaretlendikten sonra istenilen resmin seçilmesi yeterlidir (Görsel 6.79).

Select Resource	?	×
Resource context		
Import Clear		
O Project resource file:		
Properties\Resources.resx ····································		
(nona)		
Import		
ОК	Cance	I

Görsel 6.79: TabControl arka plana resim ekleme

• Bu ekleme işlemi yapıldıktan sonra tab'lar için sırasıyla "**Ana Sayfa"**, "**Öğrenci İşlemleri**", "**Not İşlemleri**" ve "**Ders İşlemleri**" yazılır. Burada yazılan isimler ekranda görünecektir (Görsel 6.80).

TabPage Collection Editor			? ×
Members:		tabpageAna properti	es:
0 tabpageAna	+	811 94 P	
1 tabpageOgrenci		✓ Accessibility	^
2 tabpageNot	+	AccessibleDescr	ipt
3 tabpageDers		AccessibleName	
		AccessibleRole	Default
		 Appearance 	
		BackColor	Transparent
		> BackgroundIma	ge System.Drawing.
		Backgroundima	gel Stretch
		Currer	Default
		> Font	Microsoft Sans Serif 8
		ForeColor	ControlText
		RightToLeft	No
Add Remove		Text	Ana Sayfa



6.9.4. ImageList Bileşeni

Formun görsel kısmını biraz daha güçlendirmek için forma ToolBox penceresinden bir tane "**ImageList**" bileşeni eklenir. ImageList, içine resimler atılabilen bir bileşendir. Daha sonra bu resimler istenilen yerlerde kullanılabilir. Burada da ImageList bileşeni forma eklendikten sonra başlık kısmına yerleştirilecek simgeler, ImageList bileşeninin **Images** (Collection) i özelliğine eklenebilir.





Görsel 6.81: ImageList bileşeni ayarları

Simgeler eklendikten sonra "ImageList"in "**ImageSize**" özelliği kullanılarak istenilen en ve boy oranı verilebilir. Bu projede **64x64** boyutu kullanılmıştır (Görsel 6.81). TabControl bileşenine tıklanıp Image-List özelliği, eklenen "**imageList1**" olarak seçilebilir. Bu seçim yapıldıktan sonra TabPage'lerde bu simgeler kullanılmaya başlanabilir. TabPage'ler tıklanır ve "**ImageIndex**" veya "**ImageKey**" özelliklerinin birinden istenilen simgeler tek tek seçilir (Görsel 6.82).

ImageIndex	📅 2 🗸 🗸
	0
	a : 1
	<u>iii</u> 2
1	3
	(none)
and the second sec	

Görsel 6.82: ImageList resim seçme

TabControl bileşeni Görsel 6.83'teki gibi görünecektir.



Görsel 6.83: TabControl sekmelerine ikon eklenmiş hâli

Her biri birbirinden farklı 4 adet sekmesi bulunan bir form elde edildi. Bu forma veri tabanı ile ilgili bileşenler eklenebilir.

10. <u>SINIF</u>



6.10. ADO.NET



Görsel 6.84: ADO.NET şeması

ADO.NET, "ActiveX Data Objects.NET" olarak adlandırılır. Veri tabanı ile uygulamalar arasında köprü görevi görür. ADO.NET ile uygulamada veri tabanına bağlanılabilir, veriler buradan listelenebilir, güncellenebilir, veri eklenebilir veya silinebilir. ADO.NET ile MySQL sorguları ve komutları uygulamada kullanılabilir (Görsel 6.84).

ADO.NET platformu sadece MySQL için değil, diğer veri tabanları için de kullanılabilir. ADO.NET mimarisinde kullanılan temel nesneler; **Connection**, **Command**, **DataReader**, **DataAdapter** ve **Data-Set**'tir.

• **Connection Nesnesi:** Projede veri tabanı kullanımı için olmazsa olmaz nesnedir. Connection nesnesi, veri tabanı ile uygulama arasında bağlantı kurmak için kullanılır.

• **Command Nesnesi:** Veri tabanı ile uygulama arasındaki veri akışını iki yönlü olarak kontrol etmeye yarar. Execute.NonQuery, ExecuteScalar ve ExecuteReader olmak üzere üç alt nesnesi vardır.

• DataReader Nesnesi: Command nesnesi aracılığı ile getirilen verileri okumak için kullanılır.

• **DataAdapter Nesnesi:** Veri tabanından alınan veriler üzerinde değişiklik yapma, veri tabanına tekrar aktarma gibi işlemler için kullanılır. Bu işlemleri yapabilmek için dört adet alt nesne bulundurur. Bunlar; seçmek için SelectCommand, eklemek için InsertCommand, güncellemek için UpdateCommand ve silmek için de DeleteCommand nesneleridir.

• **DataSet Nesnesi:** ADO.NET teknolojisinin en yenilikçi ve güçlü tarafını DataSet oluşturur çünkü DataSet nesnesi, DataAdapter ile alınan verileri bağlantısız olarak depolayabilir ve yönetebilir.

.NET Framework ile veri tabanından veri çekmek için kullanılan System.Data komutu, ADO.NET mimarisinin temelini içerir.

6.11. VERİ TABANI BAĞLANTISI VE BİLEŞENLERİN EKLENMESİ

Projenin ilk sekmesinde bütün bilgilerin ve arama kutusunun yer aldığı bir alan ve diğer üç sekmede ise her bir tabloya ait verileri görüntüleyecek, güncelleyecek, silecek alanlar oluşturuldu. Böylece ana formda üç tablo birleştirilip sadece görüntüleme işlemi yapılarak karışıklık azaltmış olur. Diğer sekmelerde de istenilen değişiklikler yapılabilir.

Projeye bir veri tabanı ekleneceğinde yapılması gereken bazı işlemler vardır. Bunlardan ilki ve en önemlisi, proje ve veri tabanı arasında köprü görevi görecek bir **"Connection String"** oluşturmaktır. Connection String'ler, veri tabanına göre değişiklik gösteren ve kalıplar hâlinde bulunan yapılardır. İnternette veri tabanına göre bu kalıplar bulunabilir ve özelleştirilebilir.

Not: Connection String'lerdeki en ufak hata, veri tabanı ile bağlantıda soruna yol açacaktır.



10, SINIF

		VERİ TABANI İŞLEMLERİ
b.	OGRENME BİRİMİ	

6.11.1. MySQL Bağlantı Kontrolü

MySQL veri tabanı kurulmuştu fakat bu kurulumda MySQL ile kod editörü arasında bağlantıyı sağlayacak bileşenler kurulmamış olabilir. Bağlantıyı sağlayacak bileşenlerin kurulup kurulmadığını anlamanın en kolay yolu, menü çubuğunda yer alan "Tools—>Connect to Database —>Change" penceresinde MySQL veri tabanının görünüp görünmediğini kontrol etmektir.

Change Data Source	? ×
Data source: Microsoft Access Database File Microsoft ODBC Data Source Microsoft SQL Server Microsoft SQL Server Database File MySQL Database Oracle Database Oracle Database <other> Data provider: NET Framework Data Provider for MySC</other>	Description Use this selection to connect to MySQL Server using the .NET Framework Data Provider for MySQL
Always use this selection	OK Cancel

Görsel 6.85: Kod editörü veri tabanı bağlantı ekranı

Görsel 6.85'teki "MySQL Database" görünmüyorsa aşağıdaki iki eklenti indirilip sisteme kurulur.

- https://www.mysql.com/products/connector/ adresinden "ADO.NET Driver for MySQL (Connector/NET)"
- https://dev.mysql.com/downloads/windows/visualstudio/

Bu işlemin ardından kod editörü yeniden başlatıldıktan sonra MySQL veri tabanlarına erişim sağlanır. MySQL Database seçildikten sonra bağlantı kontrol edilir.

Add Connection	?	×
Enter information to connect to the selected dat "Change" to choose a different data source and/	a source or or provider.	click
Data source:		
MySQL Database (MySQL Data Provider)	Change.	
Server name:		
User name:		
Password:		
Save my password		
Database name:		\sim
[Advanced	
Test Connection OK	Cancel	

Görsel 6.86: Kod editörü veri tabanı bağlantısı

• Görsel 6.86'da "Server name" kişisel bilgisayarda çalışıldığı için "localhost" olmalıdır.

• "User name", MySQL Server' kurulurken kullanılan "**root**" olmalıdır. Başka bir isim belirlendiyse o isim girilmelidir.

• "Password" ise MySQL Server kurulurken belirlenen şifre olmalıdır.

		VERİ
Ň	ÖĞRENME	

			_
Add Connection		? ×	
Enter information "Change" to choo	to connect to the selected dat se a different data source and/	a source or click or provider.	
Data source:			
MySQL Database	(MySQL Data Provider)	Change	
Server name:	localhost		
User name:	root		
Password:	•••••		
	Save my password		
Database name:		\sim	
	eokul performance_schema sakila	1	
Test Connectio	sys world		

Görsel 6.87: Kod editörü eklenebilecek tabloları listeleme

Bilgiler girildikten sonra veri tabanlarının adı geliyorsa ya da **"Test Connection**" düğmesine basıldığında "**Test connection succeeded**" yazıyorsa bağlantı doğru yapılmıştır. Bunlar olmuyorsa işlem adımları tekrar kontrol edilir (Görsel 6.87).

6.11.2. MySQL Connection String

Bağlantı cümlesi aşağıdaki kalıp ifade şeklinde yazılır.

TABANI İŞLEMLERİ

"Server= ;Database= ;Uid= ;Pwd= ;"

Kod editörü içine String olarak yukarıdaki kalıp eklenir. Buradaki değerlerin düzenlenmesi gerekmektedir. Bu projede aşağıdaki gibi bir bağlantı cümlesi olmalıdır:

"Server=localhost;Database=eokul;Uid=root;Pwd=milliegitimbakanligi;"

Server değeri, yerelde çalışıldığı için localhost; **Database** değeri, kullanılacak veri tabanı olan eokul; kullanıcı adı **(Uid)** ve şifre değerleri **(Pwd)**, MySQL kurarken belirlenen değerler olmalıdır.

6.11.3. Projeye Giriş

Görsel olarak form düzenlendi, Connection String belirlendi ve MySQL ile bağlantı sağlandı. Projenin en önemli aşaması olan .NET üzerinden veri tabanı bağlantısı yapılabilir. MySQL, references olarak projeye dâhil edilir. Bu referans ile projede bütün MySQL özellikleri kullanılabilir. Bunun için projeye sağ tıklandıktan sonra "Add References" yolu izlenip gelen pencerede arama yapılarak MySQL Data seçilmelidir (Görsel 6.88). MySQL Data ekrana gelmiyorsa "MySQL Bağlantı Kontrolü" kısmında yazılanlar tekrar uygulanır.

Reference Manager - EOkulPro	ojesi						?	×
 Assemblies 						mysql		× •
Framework Extensions Recent Search Results	✓	Name MySql.Data MySql.Data.EntityFramework for EF6 MySql.Web		Version 8.0.21.0 8.0.21.0 8.0.21.0		Name: MySql.Data Created by: Oracle Version:		
 Projects Shared Projects COM Browse 			<i>L</i> ≉			8.021.0 File Version: 8.0.21.0		
				Bro	wse.	ОК	Can	cel

Görsel 6.88: Kod editörü referans ekleme





"Connection String" özelliği çok fazla kullanılacağı için bu bir class olarak eklenir. Böylece her seferinde bütün kodu tekrar yazmak yerine sadece bu class'tan üretilen nesne ile bağlantı yapılır. Bunun için "Solution Explorer" penceresinde projenin adına sağ tıklanır. Daha sonra gelen pencerede "Add" ve ardından "Class" seçenekleri işaretlenir (Görsel 6.89).



Görsel 6.89: Kod editörü class ekleme

Class'a "**sqlbaglantisi**" adı verildikten sonra gelen kod ekranında referans olarak "**using MySql. Data**;" ve "**using MySql.Data.MySqlClient**;" kütüphaneleri eklenir (Görsel 6.90).

sqlbaglantisi.cs -	🖻 🗙 Form1.cs [Design]
🖙 EOkulProjesi	
1	□using System;
2	<pre>using System.Collections.Generic;</pre>
3	using System.Linq;
4	<pre>using System.Text;</pre>
5	<pre>using System.Threading.Tasks;</pre>
6	using MySql.Data;
7	<pre>using MySql.Data.MySqlClient;</pre>
0	



Not: Önceki adımlarda MySQL ile düzgün bağlantı kurulamamışsa burada eklenen satırların altı çizilecektir.

Connection String ile ilgili kodlar class dosyasının içine yazılır.

namespace EOkulProjesi { class sqlbaglantisi { public MySqlConnection baglan() // class icinde baglan adında fonksiyon oluşturuldu. MySqlConnection baglanti = new MySqlConnection("Server=localhost;Database=eokul;Uid=root;Pwd=12345;"); // Bağlantı yolu verildi. baglanti.Open();// Bağlantı açıldı. MySqlConnection.ClearPool(baglanti);// Bundan önceki bağlantılar temizlendi. return (baglanti);// Çağrıldığı yere bağlantı gönderildi. } }

Bu class çağrıldığında veri tabanına bir kanal açmakta ve ona erişim sağlamaktadır.

10. <u>SINIF</u>





Veri gösterimi yapılacağı için bütün sekmelere birer tane "**DataGridView**" nesnesi eklenir. DataGridView eklenirken gelen düzenleme penceresindeki **Enable** özellikler kaldırılır. "Choose Data Source" kısmı da kod ile belirtileceği için bir işlem yapılmasına gerek yoktur (Görsel 6.91).

	DataGridView Tasks	OK
~	Choose Data Source: (none)	
	Edit Columns	
	Add Column	
	Enable Adding	
	Enable Editing	
	Enable Deleting	
	Enable Column Reordering	
	Dock in Parent Container	
	Enable Column Reordering Dock in Parent Container	

Görsel 6.91: DataGridView eklenmesi

DataGridView'in Properties kısmı aşağıdaki şekilde düzenlenir:

- Önce dtgvwAna, dtgvwOgrenci, dtgvwNot, dtgvwDers şeklinde birer Name verilir.
- AllowUserToAddRows true ise false yapılır. Bu özellik en altta boş bir satır oluşmasına yol açar.

• EditMode özelliği "EditProgrammatically" olarak değiştirilir. Bu değişiklik sağlanmazsa her bir hücreye tıklandığında değişiklik yapılabilir.

• SelectionMode özelliği "FullRowSelect" yapılır. Böylece Datagridview'e tıklandığında bütün satır seçilir.

• MultiSelect özelliği false yapılır yoksa aynı anda birden fazla satır seçilir.

• AutoSizeColumnsMode özelliği AllCells yapılır. Böylece sütunlar yazı boyutu kadar otomatik olarak genişler.

• **DefaultCellStyle** özelliğinden de yazı tipi ile ilgili ayarlar yapılır. TabControl için yazı tipi büyüklüğü 20 yapılmıştır. Bu değişiklik TabControl içindeki bütün nesneleri etkilediği için DataGridView de yazı tipi boyutunu 20 almak isteyecektir. Burada uygun bir şekilde ayarlama yapılabilir. Standart yazı tipi boyutu 12'dir. Kolon başlıklarında aynı sorun yaşanırsa **ColumnHeadersDefaultCellStyle** özelliğinden font biçimlendirilebilir.

6.11.5. PROJENİN KODLAMASINA GİRİŞ

"Events" menüsünde Load'a çift tıklanarak ya da formda boş bir yere çift tıklanarak "Load" olayı aktifleştirilir, MySQL kütüphaneleri eklenir ve kullanılacak global değişkenler tanımlanır. Global değişkenler, metotlarla sınırlı olmayan küme parantezlerinin dışında tanımlanan değişkenlerdir. Böylece global değişkenler bir metotta kullanıldıktan sonra global değişkenler içindeki veri başka metotlarda da kullanılabilir.

```
using MySql.Data; //Eklenen kütüphaneler
using MySql.Data.MySqlClient;
namespace EOkulProjesi
{
   public partial class eokulForm : Form
   ł
     public eokulForm()
     {
        InitializeComponent();
     ļ
     //Global değişkenler
sqlbaqlantisi baq = new sqlbaqlantisi();// Bağlantı için oluşturulan class'tan türetilen bir nesne
DataTable table = new DataTable();//Tablolara yüklemek için bir adet DataTable değişkeni tanımlandı.
MySqlCommand kmt = new MySqlCommand();//İşlemler için bir MySqlCommand nesnesi tanımlandı.
     private void eokulForm Load(object sender, EventArgs e)
     {
      }
```

VERİ TABANI İŞLEMLERİ

ÖĞRENME BİRİMİ

Veri tabanı DataGridView nesnesine bağlanarak veri tabanındaki bilgiler proje içinde görülür. TabControl bileşeninde sekmeler **"SelectedIndexChanged"** olayı ile ayırt edilebilir. Sekmeler eklenirken her sekmeye 0'dan itibaren bir indeks verilir. Bu indeks değişimleri kontrol edilerek sekmelere erişim sağlanır.

TabControl bileşeninin SelectedIndexChanged olayına çift tıklanarak girilir. TabControl bileşeninin adı "**tabctrlEokul**" olduğu için C# bu isimde bir metot oluşturur. Hangi sekmeye gelindiğinde ne yapılması istenirse if-else yapısı içinde yazılacak şekilde aşağıdaki yapı oluşturulur.

```
private void tabctrlEokul_SelectedIndexChanged(object sender, EventArgs e)
{
    if(tabctrlEokul.SelectedIndex==0)//Ana sekmesine girilince ne yapılsın kısmı
    else if (tabctrlEokul.SelectedIndex==1)//Öğrenci sekmesine girilince ne yapılsın kısmı
    else if (tabctrlEokul.SelectedIndex==2)//Not sekmesine girilince ne yapılsın kısmı
    else //Dersler sekmesine girilince ne yapılsın kısmı
    {
        //Dersler sekmesine girilince ne yapılsın kısmı
    }
}
```

Sekmelere girildiğinde istenilen durum, veri tabanındaki tablolardan alınan verilerin **DataGridView** üzerinde gösterilmesidir. Bu gösterim işlemleri için birer metot tanımlanır ve bu metotlar **selectedIndexChanged** içinde çağrılır.

6.11.6. Dersler Sekmesi

Dersler sekmesi derslerin gösterilmesi için hazırlanır.

```
private void DersYukle()
     {
        dtgvwDers.RowHeadersVisible = false;//Satır başlıkları görünmez yapılır.
        dtgvwDers.Font = new Font("Verdana", 12);// Datagridview için font ayarlaması yapılır.
        table.Columns.Clear();// Tablonun önceki kullanımından kalan sütunlar temizlenir.
        table.Clear(); // Tablo temizlenir.
        MySqlDataAdapter adtr = new MySqlDataAdapter("select * from dersler", bag.baglan());// Komut ve bağ-
lantı yüklenen DataAdapter nesnesi
        adtr.Fill(table);//DataAdapter nesnesine yüklenen bilgiler tabloya aktarılır.
        dtgvwDers.DataSource = table; //Datagridview bileşeni data kaynağı olarak table seçilir.
        dtgvwDers.Columns["derskodu"].HeaderText = "Ders Kodu"; //MySQL'den gelen kolon adı Datagridview'de
istenildiği gibi yazılır.
        dtgvwDers.Columns["dersadi"].HeaderText = "Ders Adı";
//Bu metot, ders sekmesi tıklandığında çağrılacak şekilde yazılır.
private void tabctrlEokul_SelectedIndexChanged(object sender, EventArgs e)
        if(tabctrlEokul.SelectedIndex==0)
else if (tabctrlEokul.SelectedIndex==1)
        else if (tabctrlEokul.SelectedIndex==2)
        { }
        else //Dersler sekmesine girilince ne yapılsın kısmı
        {
           DersYukle();
        }
```

|--|



Dersler sekmesine girildiğinde **DersYukle()** metodu çağrılacak ve metottaki kodlar çalışacaktır. F5 ile kod çalıştırılarak denenir (Görsel 6.92).

E-Okul Programi			
🏥 Ana Sayfa Öğr	enci İşlemleri 📰	Not İşlemleri 🚐 De	rs İşlemleri
Ders Ders Adı			
BDN Beden Eğitimi			
BTT Bilişim Teknoloji Temelleri			
EDE Edebiyat			
FİZ Fizik			
MAT Matematik			
NESN Nesne Tabanlı Programlama			
PT Programlama Temelleri			
			And the second sec

Görsel 6.92: DersYukle() metodunun görünüşü

Diğer tablolar için de benzer şekilde metotlar yazılır ve sekmelere gelindiğinde metotların çağrılması sağlanır.

Not: Bu aşamada MySQL ile veri tabanına tekrar erişilerek gerekli alanların yazımları mutlaka kontrol edilir. Buradaki alanlar veri tabanı ile aynı olmalıdır. Tek bir harf hatası, veri tabanının düzgün çalışamamasına neden olacaktır.

6.11.7. Notlar Sekmesi

Notlar sekmesi notların gösterilmesi için hazırlanır.

private void NotYukle()
{ dtgvwNot.RowHeadersVisible = false;//Satır başlıkları görünmez yapılır. dtgvwNot.Font = new Font("Verdana", 12);// Datagridview için font ayarlaması yapılır.
table.Columns.Clear();//Tablonun önceki kullanımından kalan sütunlar temizlenir. table.Clear(); // Tablo temizlenir.
MySqlDataAdapter adtr = new MySqlDataAdapter("select * from ogrencinot", bag.baglan());//Komut ve bağlantı yüklenen DataAdapter nesnesi
<pre>adtr.Fill(table);//DataAdapter nesnesine yüklenen bilgiler tabloya aktarılır. dtgvwNot.DataSource = table; //Datagridview bileşeni data kaynağı olarak table seçilir. dtgvwNot.Columns["ogrenciNotId"].HeaderText = "Not ID"; //MySQL'den gelen kolon adı Datagridview'de istenildiği gibi yazılır. dtgvwNot.Columns["ogrNoFK"].HeaderText = "Öğrenci No"; dtgvwNot.Columns["dersKoduFK"].HeaderText = "Ders Kodu"; dtgvwNot.Columns["yazili1"].HeaderText = "1. Yazılı"; dtgvwNot.Columns["yazili2"].HeaderText = "2. Yazılı"; dtgvwNot.Columns["yazili3"].HeaderText = "3. Yazılı"; dtgvwNot.Columns["yayılama1"].HeaderText = "1. Uygulama"; dtgvwNot.Columns["sozlu1"].HeaderText = "1. Sözlü"; dtgvwNot.Columns["sozlu1"].HeaderText = "2. Sözlü"; dtgvwNot.Columns["sozlu2"].HeaderText = "2. Sözlü"; dtgvwNot.Columns["ortalama"].HeaderText = "Ortalama"; }</pre>



VI	ERİ TAB	SANI İŞ	ŞLEMI	.ERİ				
E				• •	•			•
E-Okul Program	na Sayfa	👮 Öğ	jrenci İ	İşlemle	eri 📰	Not İşler	nleri 🚝	Ders
Not ID	Öğrenci No	Ders Kodu	1. Yazılı	2. Yazılı	3. Yazılı	1. Uygulama	2. Uygula ^	
1	2345	MAT	87	85	47	95	87	
2	2345	EDE	25	36	47	95	74	
3	2345	FİZ	47	85	24	99	87	
4	3455	MAT	55	87	34	68	74	
5	3455	FİZ	99	85	74	68	52	
6	3455	EDE	14	25	66	87	25	
46	3456	MAT	87	56	52	66	98	
47	3456	EDE	95	65	32	85	69	
48	3456	FİZ	54	65	37	98	77	
49	3457	MAT	14	58	63	33	25	
50	3457	PT	25	65	32	65	32	
51	3457	NESN	87	85	86	87	90	
52	4324	FİZ	87	85	85	96	99	
53	4324	BTT	74	54	65	65	74	
54	4324	BDN	59	65	32	47	85	
55	4544	BTT	14	56	66	56	98	
56	4544	EDE	55	68	95	65	12	
57	4544	РТ	54	65	65	21	65	
58	4567	MAT	21	65	32	74	85	
150			0.5				N.C.	

EDE	55	68	95	65	12
РТ	54	65	65	21	65
MAT	21	65	32	74	85
					~~ ×

Görsel 6.93: NotYukle() metodunun görünüşü

6.11.8. Öğrenci İşlemleri Sekmesi

Öğrenci işlemleri sekmesi öğrenci bilgilerinin gösterilmesi için hazırlanır.

private void OgrenciYukle()
{ dtgvwOgrenci.RowHeadersVisible = false;//Satır başlıkları görünmez yapılır. dtgvwOgrenci.Font = new Font("Verdana",12);// Datagridview için font ayarlaması yapılır.
table.Columns.Clear();//Tablonun önceki kullanımından kalan sütunlar temizlenir. table.Clear(); // Tablo temizlenir.
MySqlDataAdapter adtr = new MySqlDataAdapter("select * from ogrencibilgi", bag.baglan());//Komut ve bağlantı yüklenen DataAdapter nesnesi
<pre>adtr.Fill(table);//DataAdapter nesnesine yüklenen bilgiler tabloya aktarılır. dtgvwOgrenci.DataSource = table; //Datagridview bileşeni data kaynağı olarak table seçilir. dtgvwOgrenci.Columns["ogrenciNo"].HeaderText = "Numara"; //MySQL'den gelen kolon adı Datagridview'de istenildiği gibi yazılır. dtgvwOgrenci.Columns["ogrenciAdi"].HeaderText = "Adı"; dtgvwOgrenci.Columns["ogrenciSoyadi"].HeaderText = "Soyadı"; dtgvwOgrenci.Columns["ogrenciBolumu"].HeaderText = "Bölümü"; dtgvwOgrenci.Columns["ogrenciSinifi"].HeaderText = "Sinifi"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bölümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bölümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bölümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bölümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bolümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bolümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bolümü"; dtgvwOgrenci.Columns["ogrenciGolumu"].HeaderText = "Bolümü"; dtgvwOgrenci.Columns["ogrenciGolumtarihi"].HeaderText = "Doğum Tarihi";</pre>

E-Okul Programi								
Ar	na Say	fa Öğı	enci İşleml	eri	Not İşl	emleri		Ders İşlemler
Numara	Adı	Sovadı	Bölümü	Sunifi	Doğum Tarihi	1		
2245	Haydar	DAČ	Cazatacilik	10	1/1/2002			
2343	Reste		Gazetetiik Vivosok İsosok	10	11/22/2002			
2455	Moliko	SANINOGULLARI	Vivocok İçecek	9	7/22/2003			
3430	Abmot	ASLAN	Pilicim Taknalajilari	9	0/22/2003			
4224	tai		Bilişini Teknolojileri Bilişim Teknolojileri	11	9/22/2001			
4524	Aven		Bilişini Teknolojileri Bilişim Teknolojileri	12	9/0/2000			
4344	Ayça	VÄE	Missini Teknolojilen Vivocok İsosok	10	12/12/2002			
4307	Övkö	KUSE	Pilicim Taknalajilari	10	12/12/2002			
47.50	Сука	KILINÇ	Cazatasilik	12	4/13/2000			
6554	Nurdoniz	SOVUU	Gazetecilik	9	2/23/2003 8/27/2000			
6700	Mohmot		GdZetetilik Vivocok İsosok	12	0/2//2000			
7045	Cizom	EIDINCI	Cazatacilik	9	7/22/2000			
7043	tdi		Bilicim Teknolojileri	12	11/2/2000			
0745	Zorda		Cazatacilik	11	5/17/2001			
0000	Murat	TEOMAN	Bilicim Teknolojileri	12	1/17/2001			
3333	rudt	LOPAN	Diagina reknolojijen	12	1, 1, 2000		-/	

Görsel 6.94: OgrenciYukle() metodunun görünüşü



İşlemleri



SelectedIndexChanged metodunun son hâli aşağıda verilmiştir.



Görüldüğü gibi öğrenci, not ve ders sekmesine girildiğinde metotlar çağrılır ve verilerin datagridview içine yüklenmesi sağlanır.

6.11.9. Anasayfa Sekmesinin Doldurulması

Ana sekmede üç tablo birleştirilecektir. SQL komutu yazılırken MySQL arayüzünde çalışmak çok daha iyi olacaktır. MySQL Workbench arayüzü açılıp eokul veri tabanındaki **ogrencibilgi**, **ogrencinot** ve **dersler** tablosunu kapsayacak bir **"inner join**" SQL sorgusu yazılır.

🗀 🗔 🗲 搅 👰 🔘 😥 🥥 🛞 📕 Limit to 100 rows 🔹 🌟 🛫 🔍 🖺 🖃											
1 • SELECT * FROM eokul.ogrencinot;											
Re	sult Grid 📘 🐧	Filter Rows:		Edi	t: 🔏 🗄	b 🖦	Export/Impor	t: 🗓 🐻	Wrap Ce	l Content:	<u>‡A</u>
	ogrenciNotId	ogrNoFK	dersKoduFK	yazili 1	yazili2	yazili3	uygulama 1	uygulama2	sozlu1	sozlu2	ortalama
•	1	2345	MAT	87	85	47	95	87	41	58	0.00
	2	2345	EDE	25	36	47	95	74	12	55	0.00
	3	2345	FİZ	47	85	24	99	87	58	74	0.00
	4	3455	MAT	55	87	34	68	74	98	52	0.00
	5	3455	FİZ	99	85	74	68	52	47	85	0.00
	6	3455	EDE	14	25	66	87	25	36	99	0.00
	46	3456	MAT	87	56	52	66	98	78	95	0.00
	47	3456	EDE	95	65	32	85	69	98	74	0.00
	48	3456	FİZ	54	65	37	98	77	65	42	0.00
	49	3457	MAT	14	58	63	33	25	47	65	0.00
	50	3457	PT	25	65	32	65	32	74	85	0.00

Görsel 6.95: MySQL'de ogrencinot tablosunun görünüşü

Normalizasyon kuralları çerçevesinde tablo olabildiğince sade ve sadece sayılardan oluşacak bir hâle getirildi. Görsel 6.95'te ogrNoFK alanı, ogrencibilgi tablosundaki ogrenciNo alanı ile ve dersKoduFK alanı ise dersler tablosundaki derskodu alanı ile ilişkilendirilmişti. Daha önce Where ifadesiyle iki tablo ilişkilendirildiği alanlar üzerinden birleştirildi fakat şu an üç tablodan veri çekilmesi gerekmektedir. Burada devreye **"Inner Join"** komutu girer çünkü ogrNoFK'ya göre öğrencinin adı, soyadı, dersKoduFK ile de dersin adı ve ogrencinot tablosundan notların gösterilmesi gerekmektedir. Inner Join geniş bir komut olduğu için MySQL içinde yazılması daha doğru olacaktır. Böylece hatalar daha net görülerek onlara müdahale edilebilir (Görsel 6.96).


I • select B.ogrenciNo, B.ogrenciAdi, B. ogrenciSoyadi, D.dersAdi, N.yazili1, N.yazili2,

- 2 N.yazili3, N.uygulama1, N.uygulama2, N.sozlu1, N.sozlu2, N.ortalama from ogrencinot as N
- 3 inner join ogrencibilgi as B
- 4 on
- 5 N.ogrNoFK=B.ogrenciNo
- 5 inner join dersler as D
 7 on
- 3 N.dersKoduFK=D.derskodu
- order by B.ogrenciNo DESC

Görsel 6.96: MySQL'de Inner Join sorgusunun yazımı

Yazılan SQL kodu incelendiğinde aşağıdaki bilgilere ulaşılır.

Select B.ogrenciNo, B.ogrenciAdi, B.ogrenciSoyadi, D.dersAdi, N.yazili1, N.yazili2, N.yazili3, N.uygulama1, N.uygulama2, N.sozlu1, N.sozlu2, N.ortalama from ogrencinot as N

Bu kısımda kullanılacak alanlar sorguya eklenir. Buradaki harfler, veri tabanlarına sonradan eklenen kısaltmaları temsil etmektedir. Örneğin, **"ogrencinot as N"** ile ogrencinot yerine N harfinin kullanılacağı belirtilmiştir.

inner join ogrencibilgi as B on N.ogrNoFK=B.ogrenciNo inner join dersler as D on N.dersKoduFK=D.derskodu order by B.ogrenciNo DESC

Bu kısımda eklenen tablolara harf olarak kısaltmalar verilirken bu tabloların hangi alana göre eşitleneceği de belirtilir. Buradaki eşitlikler, veri tabanı oluşturulurken diyagram alanında yapılan ilişkilendirmelerden kaynaklanır. Öğrenci numaraları ve ders kodu alanları birbiriyle eşitlenmiştir.

Tablo, en son satırdaki SQL kodu ile öğrenci numarası alanına göre büyükten küçüğe doğru sıralanmıştır.

	ogrenciNo	ogrenciAdi	ogrenciSoyadi	dersAdi	yazili 1	yazili2	yazili3	uygulama 1	uygulama2	sozlu1	sozlu2
►	2345	Haydar	DAĞ	Matematik	87	85	47	95	87	41	58
	2345	Haydar	DAĞ	Edebiyat	25	36	47	95	74	12	55
	2345	Haydar	DAĞ	Fizik	47	85	24	99	87	58	74
	3455	Beste	ŞAHİNOĞULLARI	Matematik	55	87	34	68	74	98	52
	3455	Beste	ŞAHİNOĞULLARI	Fizik	99	85	74	68	52	47	85
	3455	Beste	ŞAHİNOĞULLARI	Edebiyat	14	25	66	87	25	36	99
	3456	Melike	ASLAN	Matematik	87	56	52	66	98	78	95
	3456	Melike	ASLAN	Edebiyat	95	65	32	85	69	98	74
	3456	Melike	ASLAN	Fizik	54	65	37	98	77	65	42
	3457	Ahmet	KOŞUCU	Matematik	14	58	63	33	25	47	65
	3457	Ahmet	KOŞUCU	Programl	25	65	32	65	32	74	85
	3457	Ahmet	KOŞUCU	Nesne Ta	87	85	86	87	90	95	95
	4324	İdil	ALTIN	Fizik	87	85	85	96	99	98	95
	4324	İdil	ALTIN	Bilişim Te	74	54	65	65	74	85	95
	4324	İdil	ALTIN	Beden Eğ	59	65	32	47	85	99	87
	4544	Ayça	DAL	Bilişim Te	14	56	66	56	98	98	74
	4544	Ayça	DAL	Edebiyat	55	68	95	65	12	47	85
	4544	Ayça	DAL	Programl	54	65	65	21	65	87	65
	4567	Sedat	KÖSE	Matematik	21	65	32	74	85	87	85
	4567	Sedat	KÖSE	Beden Eğ	85	55	65	47	35	65	74
	4567	Sedat	KÖSE	Fizik	88	58	96	31	65	47	75
	4756	Öykü	KILINÇ	Bilişim Te	54	65	32	87	99	54	75
	4756	Öykü	KILINÇ	Edebiyat	25	98	65	74	85	65	42
	4756	Öykü	KILINÇ	Fizik	54	87	98	65	32	78	85
	5456	Sevcan	KUDRET	Beden Eğ	54	65	65	32	74	85	34
	5456	Sevcan	KUDRET	Edebiyat	56	32	98	65	74	21	65
	5456	Sevcan	KUDRET	Matematik	21	32	74	52	85	98	96
	6554	Nurdeniz	SOYLU	Matematik	32	65	98	75	98	56	66
	6554	Nurdeniz	SOYLU	Edebiyat	25	23	66	38	85	90	60
	6554	Nurdeniz	SOYLU	Beden Eğ	85	85	95	87	58	74	52
	6788	Mehmet	DENİZ	Fizik	32	32	20	25	70	60	60
	6788	Mehmet	DENİZ	Matematik	65	32	47	52	58	58	55
	6788	Mehmet	DENİZ	Edebiyat	65	74	85	65	65	78	95
	7845	Gizem	FIRINCI	Matematik	17	85	32	65	75	65	65
	7845	Gizem	FIRINCI	Fizik	85	85	90	90	98	98	88
	7845	Gizem	FIRINCI	Beden Eğ	57	85	69	85	65	32	75
	7858	İdil	LAL	Nesne Ta	65	65	32	56	58	95	85
	7858	İdil	LAL	Bilişim Te	85	85	99	98	65	65	57

Görsel 6.97: Inner Join komutu sonrası tablonun görünüşü







Dygulama

Tabloyu Görsel 6.95 ile Görsel 6.97'yi kullanarak karşılaştırınız.

AnaTabloYukle() adında bir metot oluşturulur ve bu metodun içine MySQL'de yazılan "Inner Join" kodu eklenir (Kopyala yapıştır yapılabilir.).

private void AnaTabloYukle()
{
dtgvwAna.RowHeadersVisible = false;//Satır başlıkları görünmez yapılır.
dtgvwAna.Font = new Font("Verdana", 12);// Datagridview için font ayarlaması yapılır.
table.Columns.Clear();//Tablonun önceki kullanımından kalan sütunlar temizlenir.
table.Clear(); // Tablo temizlenir.
MySqlDataAdapter adtr = new MySqlDataAdapter("select B.ogrenciNo, B.ogrenciAdi,B. ogren- ciSoyadi, D.dersAdi, N.yazili1, N.yazili2, N.yazili3, N.uygulama1, N.uygulama2, N.sozlu1, N.sozlu2, N.or- talama from ogrencinot as N inner join ogrencibilgi as B on N.ogrNoFK = B.ogrenciNo inner join dersler as D on N.dersKoduFK = D.derskodu order by B.ogrenciNo DESC", bag.baglan());//Komut ve bağlantı yüklenen DataAdapter nesnesi
adtr.Fill(table);//DataAdapter nesnesine yüklenen bilgiler tabloya aktarılır.
dtgvwAna.DataSource = table; //Datagridview bileşeni veri kaynağı olarak table seçilir.
dtgvwAna.Columns[``ogrenciNo"].HeaderText = ``Numara"; //MySQL'den gelen kolon adı Datagridview'de istenildiği gibi yazılır.
dtgvwAna.Columns["ogrenciAdi"].HeaderText = "Adı";
dtgvwAna.Columns["ogrenciSoyadi"].HeaderText = "Soyadı";
dtgvwAna.Columns["dersadi"].HeaderText = "Ders Adı";
dtgvwAna.Columns["yazili1"].HeaderText = "1. Yazılı";
dtgvwAna.Columns["yazili2"].HeaderText = "2. Yazılı";
dtgvwAna.Columns["yazili3"].HeaderText = "3. Yazılı";
dtgvwAna.Columns["uygulama1"].HeaderText = "1. Uygulama";
dtgvwAna.Columns["uygulama2"].HeaderText = "2. Uygulama";
dtgvwAna.Columns["sozlu1"].HeaderText = "1. Sözlü";
dtgvwAna.Columns["sozlu2"].HeaderText = "2. Sözlü";
dtgvwAna.Columns["ortalama"].HeaderText = "Ortalama";
}

Kod yazılır ve datagridview'de görünmesi istenilen kolon başlıkları eklenir. Metot, diğerlerinden farklı olarak **FormLoad()** içinde çağrılır. Form açıldığında ilk gelen bu sekme olduğu için form yüklendiği zaman verilerin ekrana gelmesi gerekir. Ayrıca **SelectedIndexChanged** metodu içine de bu metot eklenir.

private void tabctrlEokul_SelectedIndexChanged(object sender, EventArgs e)
if(tabctrlEokul.SelectedIndex==0)
AnaTabloYukle();
e^{f} else if (tabctrlEokul.SelectedIndex==1)
် OgrenciYukle();
e^{I} else if (tabctrlEokul.SelectedIndex==2)
رُ NotYukle();
else
َ DersYukle();
}



Projenin son hâli Görsel 6.98'deki gibi olacaktır. Artık hangi sekmeye geçilirse veriler tablolardan o sekmeye gelecektir.

E-Okul Progr	ami											
	Ana S	Sayfa	Öğrenci İşleı	nleri		Not	İşlem	leri ₄	P I	Ders İ	şlem	leri
Numara	Adı	Soyadı	Ders Adi	1. Yazılı	2. Yazılı	3. Yazılı	1. Uygulama	2. Uygulama	1. Sözlü	2. Sözlü	Ortalama	^
9999	Murat	TEOMAN	Bilişim Teknoloji Temelleri	98	95	100	95	85	98	90	0.00	
9999	Murat	TEOMAN	Matematik	90	95	95	98	90	85	85	0.00	
9999	Murat	TEOMAN	Nesne Tabanlı Programlama	100	95	95	99	95	97	89	0.00	
8745	Zerda	DENÍZ	Fizik	54	21	65	98	91	91	74	0.00	
8745	Zerda	DENÍZ	Beden Eğitimi	14	58	65	98	63	58	74	0.00	
8745	Zerda	DENÍZ	Edebiyat	54	65	32	65	82	17	82	0.00	
7858	İdil	LAL	Fizik	52	63	65	85	54	55	54	0.00	
7858	İdil	LAL	Bilişim Teknoloji Temelleri	85	85	99	98	65	65	57	0.00	
7858	Ídil	LAL	Nesne Tabanlı Programlama	65	65	32	56	58	95	85	0.00	
7845	Gizem	FIRINCI	Fizik	85	85	90	90	98	98	88	0.00	
7845	Gizem	FIRINCI	Beden Eğitimi	57	85	69	85	65	32	75	0.00	
7845	Gizem	FIRINCI	Matematik	17	85	32	65	75	65	65	0.00	
6788	Mehmet	DENÍZ	Edebiyat	65	74	85	65	65	78	95	0.00	
6788	Mehmet	DENÍZ	Fizik	32	32	20	25	70	60	60	0.00	
6788	Mehmet	DENÍZ	Matematik	65	32	47	52	58	58	55	0.00	
6554	Nurdeniz	SOYLU	Edebiyat	25	23	66	38	85	90	60	0.00	
6554	Nurdeniz	SOYLU	Matematik	32	65	98	75	98	56	66	0.00	
6554	Nurdeniz	SOYLU	Beden Eğitimi	85	85	95	87	58	74	52	0.00	~

Görsel 6.98: Projenin son hâli

6.12. KAYIT EKLEME

Tablolara C# üzerinden veri eklemek için projeye yeni bir form eklenir ve bu form, ana sekme üzerine eklenecek bir düğme yardımıyla çağrılır (Görsel 6.99).

	- 0	Solution Explo	rer		
		Solution Explo	rer III Exp IEOki	(o + 5 C (P (o)) / - Horer (Ctri+) Build Rebuild Clean Analyze and Code Cleanup Publish	
		b c≈ Pr b c≈ sc	Ð	Scope to This New Solution Explorer View	
10 10 10 10 10 10 10 10 10 10 10 10 10 1	New Iem Existing Iem New Folder From Cockiecuter RESTAPI Client Machine Learning Reference Service Reference Connected Service	Ctrl+Shift+A Shift+Alt+A	曲 中 人 人 へ 人	Add Manage Nuclea Packages Set as Startup Project Debug Initialize Interactive with Project Source Control Cot Cot Paste Remove Remo	, Ctrl+X Ctrl+V Del F2
10 10 10 10 10 10 10 10 10 10 10 10 10 1	Analyzer Form (Windows Forms) User Control (Windows Forms) Component Class New EditorConfig			Unload Project Load Direct Dependencies of Project Load Entire Dependency Tree of Project Copy Full Path Open Folder in File Explorer Open in Terminal Properties	Alt+Enter

Görsel 6.99: Projeye yeni form ekleme

Form eklenirken forma bir ad verilmelidir. Forma **"YeniKayit"** ismi verilir. Boş bir form gelir ve bu form Görsel 6.100'deki gibi düzenlenir. İstenilen şekilde biçimlendirme yapılabilir.

YeniKayit
Öğrenci No IblOgrenci
Ders Kodu
IbIDers
1. Yazılı
2. Yazılı
3. Yazılı
1. Uygulama
2. Uygulama
1. Sözlü
2. Sözlü
Kayıt Ekle

Görsel 6.100: Yeni kayıt formunun tasarımı





İlk formdan ikinci formu çağırmak için ilk forma bir buton eklenir ve **Click** olayına aşağıdaki kodlar yazılır (Görsel 6.101).

YeniKayit kayitform = new YeniKayit(); //Formun türünden bir nesne oluşturuldu. kayitform.Show(); //Bu nesne (form) görünür yapıldı.



Görsel 6.101: Kayıt Ekle butonu eklenmesi

Burada dikkat edilmesi gereken nokta, öğrenci bilgilerinin **ogrencibilgi** tablosundan ve ders bilgisinin de **dersler** tablosundan alınması gerektiğidir. Bu bilgiler, kullanıcıdan girdirilmeye kalkılırsa bu tablolarda olmayan bilgilerde veri tabanı hata verecektir. Bu yüzden bu bilgiler ilgili tablolardan çekilerek kullanıcıya seçtirilir. Formun tasarımı tamamlandıktan sonra veri tabanı bağlantısı için tekrar eklenir.

using MySql.Data; using MySql.Data.MySqlClient;

//İki adet global değişken de eklenir.

sqlbaglantisi bag = new sqlbaglantisi();//Bağlantı için class oluşturuldu.

MySqlCommand kmt = new MySqlCommand();//İşlemler için komut değişkeni

Artık veri tabanıyla bağlantı kurulup veriler alınabilir. Kodlara başlamadan önce öğrenci numarası ile ders kodları veri tabanından çekilir. Seçilen değere göre öğrenci adı ve soyadı IblOgrenci'ye, ders koduna göre dersin adı IblDers'e aktarılır. Ardından alt tarafta notların girişi tamamlandıktan sonra ortalama da hesaplanarak veri tabanına ekleme işlemi gerçekleştirilir. Önce Load anında comboboxlar doldurulur. private void YeniKayit_Load(object sender, EventArgs e) {

DataTable tableogrenci = new DataTable();//tableogrenci nesnesi oluşturuyoruz

MySqlDataAdapter adtrogrenci = new MySqlDataAdapter("select ogrenciNo from ogrencibilgi", bag.baglan());//dataadapter içerisine sql cümlemizden gelen kayıtları yüklüyoruz

adtrogrenci.Fill(tableogrenci);//veriyi datatable nesnemize yüklüyoruz

adtrogrenci.Dispose();

cmbOgrNo.DataSource = tableogrenci;//data kaynağı olarak datatable veriyoruz

cmbOgrNo.ValueMember = tableogrenci.Columns[0].ToString();

DataTable tableders = new DataTable();//tableogrenci nesnesi oluşturuyoruz

MySqlDataAdapter adtrders = new MySqlDataAdapter("select derskodu from dersler", bag.baglan());//dataadapter içerisine sql cümlemizden gelen kayıtları yüklüyoruz

adtrders.Fill(tableders);//veriyi datatable nesnemize yüklüyoruz

cmbDersKodu.DataSource = tableders;//data kaynağı olarak datatable veriyoruz

cmbDersKodu.ValueMember = tableders.Columns[0].ToString(); }



ÖĞRENME BİRİMİ

Öğrenci no combobox'una çift tıklanarak **cmbOgrNo_SelectedIndexChanged** metodu oluşturulur. Bu metot ile öğrenci numarası değiştiğinde seçilen öğrencinin adı ve soyadı, altındaki lblOgrenci değişkenine aktarılır.

private void cmbOgrNo_SelectedIndexChanged(object sender, EventArgs e)
{
 kmt.Connection = bag.baglan();//SQL komutu yazılacak değişkene veri tabanı ile bağlantı sağlanır.
 kmt.CommandText = ("select * from ogrencibilgi where ogrenciNo = `" + cmbOgrNo.Text.ToString()
 + ``");//Combobox'tan gelen değer SQL komutuna eklenir.
 kmt.ExecuteNonQuery();//SQL komutu çalıştırılır.
 MySqlDataReader oku = kmt.ExecuteReader();//SQL komutu sonucunda okunan bütün değerler datareader nesnesine aktarılır.
 while (oku.Read())//While döngüsü ile datareader içindeki veriler bitene kadar tek tek okunur.
 Şu an için bu veriler öğrenci adı ve soyadıdır.
 {
 IblOgrenci.Text = oku["ogrenciAdi"].ToString() + `` `` + oku["ogrenciSoyadi"].ToString();//
Gelen öğrenci ad ve soyadı birleştirilip Iblogrenci'ye aktarılır.
 }
 oku.Close();// Veriler bittikten sonra bağlantı kapatılır.
 }
}

Yaklaşık aynı kodlar **cmbDersKodu_SelectedIndexChanged** için yazılır. Bu tür durumlarda copy-paste yapılabilir fakat bileşen ve değişken isimlerine çok dikkat edilmelidir.

private void cmbDersKodu_SelectedIndexChanged(object sender, EventArgs e)
{
 kmt.Connection = bag.baglan();//SQL komutu yazılacak değişkene veri tabanı ile bağlantı sağlanır.
 kmt.CommandText = (`select * from dersler where derskodu = `" + cmbDersKodu.Text.ToString() + ``");//Combobox'tan gelen değer SQL komutuna eklenir.
 kmt.ExecuteNonQuery();//SQL komutu çalıştırılır.
 MySqlDataReader oku = kmt.ExecuteReader();//SQL komutu sonucunda okunan bütün değerler datareader nesnesine aktarılır.
 while (oku.Read())//While döngüsü ile datareader içindeki veriler bitene kadar tek tek okunur.
 Şu an için bu veri ders adıdır.
 {
 lblDers.Text = oku[``dersAdi''].ToString();//Gelen ders adı lblDers'e aktarılır.
 }
 oku.Close();//Veriler bittikten sonra bağlantı kapatılır.
 }
}

🚔 YeniKayit — 🗆 🗙
Öğrenci No 4756 - Öykü KILINÇ
Ders Kodu NESN
Nesne Tabanlı Programlama
1. Yazılı
2. Yazılı
3. Yazılı
1. Uygulama
2. Uygulama
1. Sözlü
2. Sözlü
Kayıt Ekle

Görsel 6.102: Formun son durumu





Son olarak ortalama hesaplanarak bilgiler veri tabanına kaydedilir. Önce int türünde bir metot tanımlanır.

Bu metotta bütün textbox içindeki notlar alınarak önce integer'e convert edilmiş sonra da toplanarak sonuç 7'ye bölünmüştür. En sonunda da ortalama değer döndürülmüştür.

Kayıt ekleme işlemine girmeden önce aynı öğrenci için aynı dersin eklenmesinin önüne geçilmelidir. Kontrol için **Boolean** metodu yazılır.

Bağlantı kısmı aynı kalır. Veri okunduktan sonra gelen değerler if ile kontrol edilerek hem öğrenci no hem de ders kodu varsa bu kaydın önceden eklendiği anlaşılır ve **false** döndürülür. Gelen verinin tamamında aynı kayıt yoksa **true** döndürülür. Bu verileri veri tabanına eklemek için gerekli kodlar butonun **click** olayına yazılır. Bu kodlar istenirse metot olarak da yazılabilir.





Veri tabanına kayıt eklerken parametreler kullanılabilir. Parametrelerin kullanımı görsel 6.103'te görülmektedir. Insert into cümlesine değerleri direkt vermek yerine (txtAd.Text gibi) @ad gibi bir parametre ile değeri değişken üzerinden aktarılabilir. Böylece hem kod daha düzenli gözükecektir hem de veri güvenliği açısından daha avantajlı bir kullanım olacaktır. Koda başlarken **KayitVarmi()** metodu kullanılarak öğrenci için bu dersin olup olmadığı kontrol edilir. **True** dönerse kod **try-catch** blokları içinde çalıştırılır, **false** dönerse hata mesajı verdirilir (Görsel 6.103).

	🚔 VeniKayit — 🗆 🗙	🚔 Yan Kayê — 🗆 🗙
	™ Kor Öğrenci No 4324 ∽ İdil ALTIN	™≪ Öğrenci No <mark>9999 →</mark> Murat TEOMAN
	Ders Kodu <mark>FİZ</mark> Fizik	Ders Kodu <mark>NESN →</mark> Nesne Tabanlı Programlama
	1. Yazılı 56 2. Yazılı 3. Yazıl 1. Uygurama 55 2. Uygulama 25 1. Sözlü 47 2. Sözlü 44	1. Yazılı 87 2. Yazılı bere delar en delararadır. 3. Yazılı bere delar en delararadır. 1. Uygularma 45 1. Sözlü 21 2. Sözlü 12
	Kayıt Ekle	Kayıt Ekle
Uygulama	Görsel 6.103: Başarılı ve	e başarısız kayıt ekleme sonuçları

- 1. Veri tabanına ekleme yapıldıktan sonra bütün alanları temizleyen metodu araştırarak yazınız.
- 2. Forma girilen değer 0'dan küçük veya 100'den büyükse tekrar değer girilmesini sağlayınız.
- 3. Forma sadece rakam girilmesini sağlayınız.





Kayıt eklendikten ve kayıt formu kapatıldıktan sonra ana form üzerinde kaydı görmek için bir tane yenile butonu eklenerek **AnaTabloYukle()** metodu çalıştırılır.



Böylece kayıt eklendikten sonra düğmeye basılarak kaydın görünmesi sağlanabilir. Metot kullanımı sayesinde tek bir satır ile bütün kodlar tekrar çağrılarak kullanılabilir.

6.13. ARAMA METODU

Ana formda öğrenci adına göre arama yapan bir metot yazılır. Bunun için ana sekmeye bir textbox eklenip **TextChanged** olayına arama metodu yazılarak çağrılır.

private void AnaSekmeAramaYap()
{
DataTable Anatable = new DataTable();//Bilgilerin aktarılacağı bir Datatable nesnesi oluşturulur.
Anatable = table.Copy();//Sekme çağrılırken bilgilerin yüklendiği table nesnesinin bir kopyası
içine aktarılır.
DataView dv = Anatable.DefaultView;//Bu bilgileri görüntülemek için DataView nesnesi oluştu-
rulup table ile bağlanır.
dv.RowFilter = "ogrenciAdi LIKE '%" + txtAra.Text + "%' OR ogrenciSoyadi LIKE '%" + txtAra.
Text + "%";//Arama cümlesi yazılır.
dtgvwAna.DataSource = dv;//Gelen arama sonuçları datagridview içine aktarılır.
}

Burada SQL kullanmak işi uzatacağı için doğrudan datagridview nesnesinin **RowFilter** özelliği kullanılmıştır. Bu özelliğe veri tabanı üzerinden çekilen kolon adları baz alınarak (ogrenciAdi ve ogrenciSoyadi) sorgusu yazılır. Bu sorgu, datagridview üzerinde ad ve soyad alanına göre arama yapacaktır. Arama işlemine istenirse diğer alanlar da eklenebilir (Görsel 6.104).

A	na Sa	yfa 🚦	Öğrer	nci İşle	emleri		ot İşleml	eri 🚐 D	ers İşl	emleri
Numara	Adı	Soyadı	Ders Adı	1. Yazılı	2. Yazılı	3. Yazılı	1. Uygulama	2. Uygulama	1. Sözlü	2. Sözlü
8745	Zerda	DENİZ	Beden Eğitimi	14	58	65	98	63	58	74
8745	Zerda	DENİZ	Fizik	54	21	65	98	91	91	74
8745	Zerda	DENİZ	Edebiyat	54	65	32	65	82	17	82
6788	Mehmet	DENİZ	Matematik	65	32	47	52	58	58	55
6788	Mehmet	DENİZ	Edebiyat	65	74	85	65	65	78	95
6788	Mehmet	DENİZ	Fizik	32	32	20	25	70	60	60
6554	Nurdeniz	SOYLU	Beden Eğitimi	85	85	95	87	58	74	52
6554	Nurdeniz	SOYLU	Edebiyat	25	23	66	38	85	90	60
6554	Nurdeniz	SOYLU	Matematik	32	65	98	75	98	56	66
٢										>
Kayıt	Ekle		Yenile		Arar	na: der	n			

Görsel 6.104: Arama işlemi

6.14. EKLEME, SİLME VE GÜNCELLEME İŞLEMLERI

Öğrenci, not ve ders işlemleri sekmelerinde ekleme, silme, güncelleme yapılır. Sekmelerin üçünde de aynı işlemler yapılacağı için ilk seferden sonra bunlar copy-paste ile gerçekleştirilebilir.





6.14.1. Öğrenci Sekmesi İşlemleri

Öğrenci sekmesi içinde **OgrenciYukle();** metodu kullanılır. Böylece veri tabanına bağlanılarak bilgiler çekilir. Veri tabanından çekilen kolon başlıklarına göre işlemler gerçekleştirilir. Önce form tasarlanarak formdaki bileşenlere birer **"Name"** verilir.

E-Okul Prograr	m									
	Ana Say	yfa Ö	İğrenci İşleı	nleri		t İşle	mleri		Ders İşle	emler
			and the			_				
Numara	Adı	Soyadı	Bölümü	Sinifi	Doğum Tarihi				0	
2345	Haydar	DAĞ	Gazetecilik	10	1/1/2002		• кау	текіе	O kayit Gunce	lie
3455	Beste	ŞAHİNOĞULLAR	I Yiyecek İçecek	9	11/22/2003					
3456	Melike	ASLAN	Yiyecek İçecek	9	7/22/2003					
3457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	9/22/2001		Numar	a		
4324	İdil	ALTIN	Bilişim Teknolojileri	12	9/8/2000					
4544	Ауçа	DAL	Bilişim Teknolojileri	10	12/12/2002		Adı		-	
4567	Sedat	KÖSE	Yiyecek İçecek	10	12/12/2002				A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF A CONTRACTOR OF	
4756	Öykü	KILINÇ	Bilişim Teknolojileri	12	4/13/2000		Sovadı			
5456	Sevcan	KUDRET	Gazetecilik	9	12/23/2003		, au	1 1.4		*61-H
6554	Nurdeniz	SOYLU	Gazetecilik	12	8/27/2000		Rölüm	-		
6788	Mehmet	DENİZ	Yiyecek İçecek	9	12/17/2003		Joium			
7845	Gizem	FIRINCI	Gazetecilik	12	7/23/2000					
7858	İdil	LAL	Bilişim Teknolojileri	12	11/2/2000		SINIT		~	
8745	Zerda	DENİZ	Gazetecilik	11	5/17/2001		. T		1/10/2020	
9999	Murat	TEOMAN	Bilişim Teknolojileri	12	1/17/2000		J. Tari		1/18/2020	
						100				
							F	klo	Sil	1.1
								e	31	2.8
<					>	> Si	State State			
14 150	1	A Sector Sector	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	Sec. 20		1000				

Görsel 6.105: Öğrenci sekmesinde kayıt ekleme ve güncelleme için form tasarımı

Görsel 6.105'teki gibi sağ tarafa bir **panel** ve içine de kullanılacak bileşenler eklenir. Burada en üstte **radiobutton** ekleme sebebi, seçilen radiobutton'a göre ekleme veya güncelleme işleminin yapılmasıdır. Daha sonra güncelleme ve ekleme yapılacak her bir alan için bir **label** ve bir **textbox** eklenir. Doğum tarihi için bir adet **datetimepicker**, en alta da butonlar eklenir. Sınıf combobox'ının **DropDownStyle** değeri **DropDownList** yapılır ve **Items** özelliğinden 9, 10, 11, 12 eklenir.

Not: Burada textbox'lara birer name verilmesi çok önemlidir çünkü formda bolca kullanılacaktır. İsimlendirme yapılmaması çok büyük karışıklıklara yol açar.

Tasarım yapıldıktan ve isimlendirmeler gerçekleştirildikten sonra kodlar yazılabilir. Burada seçilen radiobutton'a göre bu düğmede ekle veya güncelle yazacaktır. Bunun için soldaki **rdbEkle** radiobutton'unun **Checked** özelliği true yapıldıktan sonra **CheckedChanged** olayına aşağıdaki kodlar yazılır.

private void rdbEkle_CheckedChanged(object sender, EventArgs e)
{
 if (rdbGuncelle.Checked)
 btnGuncelleEkle.Text = "Güncelle";
 else
 btnGuncelleEkle.Text = "Ekle";
}

Böylece form açıldığında butonda **Ekle** yazarken diğer seçenek seçildiğinde düğme üzerinde **Güncelle** yazacaktır. Ekle için btnGuncelleEkle düğmesine gerekli kodlar yazılır. Burada basit bir **INSERT INTO** cümlesi yazılır ve textbox'lardaki değerler veri tabanına gönderilir.

Not: Ekleme işlemi yapılırken kod yazımına çok dikkat edilmelidir. Kesin bir sonuca varılmak istenirse alanlar birer birer eklenip alanlara tek tek değer girilerek gidilmesi eklemeyi kolaylaştıracaktır.



10, SINIF



Burada normal bir SQL cümlesi yazılır fakat veri tabanına değerler el ile değil, kullanıcının forma girdiği değerlere göre eklenir. **Kmt** değişkeni, MySqlCommand'dan türetilmiştir ve içine SQL komutları yazılabilir. Burada da kmt içine komut olarak **INSERT INTO** cümlesi yazıldı. Önce veri tabanındaki veri aktarılacak alanlar yazıldı, ardından VALUES ile her bir alana denk gelen bir parametre tanımlandı ve son olarak bu parametrelere değerler atandı. Formdaki veriler, **kmt.ExecuteNonQuery()** çalıştırıldığında bir hata yoksa veri tabanına eklenecektir.

Veriler eklendikten sonra bağlantı kapatılır. Veri eklendiği için tablo **güncellenir**. Daha sonra kayıt eklendiğine dair mesaj, **MessageBox.Show** ile ekrana yazdırılır.

Kodda bir hata olursa bu hatanın giderilmesi için **try-catch-finally** kullanımı önemlidir. Try kodların olduğu alanı, catch bir hata olduğunda verilecek mesajı ve finally de hata olsa da olmasa da çalışacak kodları içerir. Bu yapı ile kodlardan veya kullanıcıdan kaynaklanan bir hata olsa bile program çökmeyecektir. Onun yerine catch kısmında yazılan mesajı veya kodları çalıştıracaktır. Bu projede finally'e gerek olmadığı için kullanılmamıştır. Proje çalıştırılarak denenir.





Okul Programi										
	na Sa	vfa 👮 Ö	ărenci İslei	nleri		İsler	nleri 🛲		ers İsl	eml
<u> </u>			g. eeş.e.			Lyici	÷		CIO 191	
Numara	Adı	Soyadı	Bölümü	Sinifi	Doğum Tarihi			1		
2345	Havdar	DAĞ	Gazetecilik	10	1/1/2002		Kayıt Ekle	9	O Kayıt Günc	elle
3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	11/22/2003					
3456	Melike	ASLAN	Yiyecek İçecek	9	7/22/2003					
3457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	9/22/2001	N	umara	45	56	
4324	İdil	ALTIN	Bilişim Teknolojileri	12	9/8/2000					
4544	Ayça	DAL	Bilişim Teknoloj	i	×	A	dı	Yal	cun	
4566	Yakup	KUTLU	Bilişim Teknoloj			12 2.5			(up	
4567	Sedat	KÖSE	Yiyecek İçecek	ler veritabanu	na hasari ile eklendil	S	heve	VII	TI II	
4756	Öykü	KILINÇ	Bilişim Teknoloj				yuui	NU		
5456	Sevcan	KUDRET	Gazetecilik			B	slüm	D:1:		
6554	Nurdeniz	SOYLU	Gazetecilik		OK	DU	Juin	ып	şım re	
6788	Mehmet	DENÍZ	Yiyecek İçecek	9	12/17/2003	-		-		
7845	Gizem	FIRINCI	Gazetecilik	12	7/23/2000	SI	nir	12	~	
7858	İdil	LAL	Bilişim Teknolojileri	12	11/2/2000		Tault	-	4/2022	
8745	Zerda	DENÍZ	Gazetecilik	11	5/17/2001	D.	larihi	6/	4/2002	U
9999	Murat	TEOMAN	Bilişim Teknolojileri	12	1/17/2000					
							Ekle		Sil	
						1	State	1034		

Görsel 6.106: Öğrenci sekmesine kayıt eklenmesi

Görsel 6.106'da kayıt eklendi ve tabloya geldi. Hata almak için aynı kayıt tekrar eklenmeye çalışılır.

A	na Sa	yfa 🐺 Ö	ğrenci İşle	mleri	Not	İşler	nleri 🚆	🚆 D	ers İşle	emle
Numara	Adı	Soyadı	Bölümü	Sinifi	Doğum Tarihi					
2345	Haydar	DAĞ	Gazetecilik	10	1/1/2002		Kayıt Ekle	e	O Kayıt Günce	lle
3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	11/22/2003					
3456	Melike	ASLAN	Yiyecek İçecek	9	7/22/2003					
8457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	9/22/2001	N	umara	456	56	
324	İdil	ALTIN	Bilişim Teknolojil		×				-	
1544	Ayça	DAL	Bilişim Teknolojik			A	dı	Yal	(UD	
4566	Yakup	KUTLU	Bilişim Teknolojile	Ekleme Basarısı	z			_		
4567	Sedat	KÖSE	Yiyecek İçecek	ütfen Alanları	Kontrol Ediniz.	S	ovadı	KII	тни	
1756	Öykü	KILINÇ	Bilişim Teknolojile	30ş alan olama	iz.			NO		
5456	Sevcan	KUDRET	Gazetecilik			B	ilim	DII	cime To	
5554	Nurdeniz	SOYLU	Gazetecilik		OK		Juin	DIII	şim re	
5788	Mehmet	DENÍZ	Yiyecek İçecek			C.				
7845	Gizem	FIRINCI	Gazetecilik	12	7/23/2000	5	nii	12	*	
7858	İdil	LAL	Bilişim Teknolojileri	12	11/2/2000	-	Taulhi	61	4/2002	_
3745	Zerda	DENÍZ	Gazetecilik	11	5/17/2001	D	rarini	6/	4/2002	
9999	Murat	TEOMAN	Bilişim Teknolojileri	12	1/17/2000					
							Ekle		Sil	
					>	La Carton				

Görsel 6.107: Öğrenci sekmesine kayıt ekleme hatası

Görsel 6.107'de hata mesajı geldi fakat program sonlanmadı.

Sıra Sizde

E-Okul Program

Görsel 6.107'deki hatanın sebebini açıklayınız.

VERİ TABANI İŞLEMLERİ

6.14.1.2. Öğrenci Sekmesi Kayıt Güncelleme

Ekleme kodları bittikten sonra güncelleme kısmına geçilebilir. Güncelleme kodlarına geçmeden önce güncellenecek kaydı seçmek için bir işlem daha yapılmalıdır. Güncelleme işlemi, var olan bir kayıt üzerinde değişiklik yapmaktır. **Datagridview** nesnesine çift tıklandığında güncellenecek kaydı otomatik olarak forma yükleyen kodlar datagridview'in **CellDoubleClick** olayına yazılır.

private void dtgvwOgrenci_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
 //Burada yapılan işlem textbox'lara veri aktarmaktır.
 txtOgrNumara.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[0].Value.ToString();
 txtOgrAd.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[1].Value.ToString();
 txtOgrSoyad.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[2].Value.ToString();
 txtOgrBolum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[3].Value.ToString();
 txtOgrBolum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[3].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.Text = dtgvwOgrenci.Rows[e.RowIndex].Cells[5].Value.ToString();
 dateDogum.T



|--|



D	الد مالم	In	 I	 	La :	السيد ومرا	

ÖĞRENME BİRİMİ VERİ TABANI İŞLEMLERİ

Bu kodla beraber herhangi bir kayıt çift tıklandığında o satırdaki veriler otomatik olarak textbox'lara yüklenecektir. Burada sağ taraftaki **Cells[0]** kısmı, datagridview'deki kolonları temsil etmektedir ve sol tarafla uyumlu olmak zorundadır. İlk satır baz alınırsa numara textbox'ına datagridview'deki o satırın ilk kolonundaki öğrenci numarası aktarılmaktadır. **Bu uyum çok önemlidir.**

Ekleme kısmı, radiobutton kontrol edilerek yapılmıştır. Ekleme yerine **rdbGuncelle** seçiliyse kodlar güncelleme işlemine göre yazılmalıdır. Bu kodlar önceki if'in devamıdır. Alanlar dolduktan sonra güncelleme kodları yazılabilir.

else if (rdbGuncelle.Checked) ſ try //Komut cümlesi parametreler ile yazılacak ve aşağıdaki parametrelere değer atanacaktır. string sorgu = "UPDATE ogrencibilgi SET ogrencino= @no , ogrenciAdi= @ad , ogrenciSoyadi= @so-yad , ogrenciBolumu= @bolum , ogrenciSinifi= @sinif , ogrenciDogumTarihi= @dogum where ogrenciNo= @no "; kmt = new MySqlCommand(sorgu, bag.baglan()); //Parametrelere değer atanması //Parametrelere deger atanmasi kmt.Parameters.AddWithValue(``@no", txtOgrNumara.Text); kmt.Parameters.AddWithValue(``@ad", txtOgrAd.Text); kmt.Parameters.AddWithValue(``@soyad", txtOgrSoyad.Text); kmt.Parameters.AddWithValue(``@bolum", txtOgrBolum.Text); kmt.Parameters.AddWithValue(``@osinif", cmbOgrSinif.Text); kmt.Parameters.AddWithValue(``@dogum", dateDogum.Value.Date.ToString(``yyyy-MM-dd")); kmt.Connection = bag.baglan();//Bağlantı yolunu verildi. kmt.ExecuteNonQuery();//Komut calistirildi. kmt.Connection.Close();//Bağlantı kapatıldı. OgrenciYukle();//Datagridview'e yeni öğrenci listesi yüklendi. MessageBox.Show("Güncelleme İşlemi Başarılı");//işlem başarılı mesajı veriyoruz catch//Hata yakalama baslangici MessageBox.Show("Güncelleme Başarısız!\nLütfen Alanları Kontrol Ediniz.\nAynı numarada 2 öğrenci olamaz.\nBoş alan olamaz.");//Uyarı mesajımız } } }

Buradaki yapı da ekleme ile paraleldir sadece kmt.CommandText'indeki SQL sorgu değişmiştir. UPDATE sorgusunda veriler doğrudan alanlara atanır. Ayrıca Where kısmına veri tabanından alınan sütun ismi yazılmalıdır. Burada "**ogrenciNo**" seçilen satırdan okunmuştur.

6.14.1.3. Öğrenci Sekmesi Kayıt Silme

Veri silme de diğer işlemlere benzer şekilde yapılmaktadır fakat silme işlemi geri alınamadığı için öncesinde kodlara bir onay mekanizması eklenmelidir. Burada yanlışlıkla veri silinmesinin önüne geçebilmek için bir uyarı penceresi ile onay alınacaktır.

264



DialogResult türünde tanımlanan cikis değişkeni ile bu onay penceresinden gelen Evet (Yes) veya Hayır (No) ifadelerine göre işlem yapılmalıdır.

Numara	Adı	Soyadı	Bölümü	Sinifi	Doğum Tarihi						
2345	Haydar	DAĞ	Gazetecilik	10	1.01.2002			Kayıt Eki	e	O Kayıt Günce	elle
3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	22.11.2003						
3456	Melike	ASLAN	Yiyecek İçecek	9	22.07.2003				-		
3457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	22.09.2001		Nu	mara			
4324	İdil	ALTIN	Bilişim Teknolojileri	12	8.09.2000						
4544	Ayça	DAL	Bilişim Tek Uyan			\times	Ad				
4566	Yakup	KUTLU	Bilişim Tek								
4567	Sedat	KÖSE	Yiyecek İçe 🔒 🔬	ne islemine d	levam etmek istivormusunuz	,	So	adı			
4756	Öykü	KILINÇ	Bilişim Tek					- Carella			
5456	Sevcan	KUDRET	Gazetecilik			_	Böl	üm			
6554	Nurdeniz	SOYLU	Gazetecilik		Yes No		DUI	um			
6788	Mehmet	DENİZ	Yiyecek İç ecen	5	17.12.2005	-	C				
7845	Gizem	FIRINCI	Gazetecilik	12	23.07.2000		SIN	п		~	
7858	İdil	LAL	Bilişim Teknolojileri	12	2.11.2000			T-ullet.	10.	1 2020	_
8745	Zerda	DENİZ	Gazetecilik	11	17.05.2001		D .	larini	18.1	1.2020	
9999	Murat	TEOMAN	Bilişim Teknolojileri	12	17.01.2000	1					
						1		Eklo		Sil	1
								EKIE			

Görsel 6.108: DialogResult penceresi görünüşü

Bu onay ekranında "**Evet**" gelirse silme işlemi yapılacaktır. Veri tabanları arasındaki ilişkilerden dolayı silmek istenilen veri başka bir tabloda kullanılmışsa oradaki verilerde boşluk oluşmaması için silme işlemine izin verilmeyecektir. Bir öğrencinin notlar tablosunda kaydı varsa öğrencinin silinmesi istenildiğinde aradaki ilişkiden dolayı notlar tablosunda bu kişi bulunduğu için hata mesajı alınır. Böyle bir yapı olmasa ve öğrenci silinebilse notlar tablosunda artık öyle bir öğrenci olmadığı için çöküş yaşanır. **Bu yüzden silme işlemlerinde kontroller iyi yapılmalıdır.**

i_/-			•	•				⇒.∕~	
	na Say	yfa <u> </u> ö	ğrenci İşle	mler	i 📰 Not	İş	lemleri 🚝	Ders İş	le
			1			1 Г			
Numara	Adı	Soyadı	Bölümü	Sinifi	Doğum Tarihi		O Warnet Field	O Knut Oliv	
2345	Haydar	DAĞ	Gazetecilik	10	1.01.2002		C Kayit Ekie		cei
3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	22.11.2003				
3456	Melike	ASLAN	Yiyecek İçecek	9	22.07.2003				7
3457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	22.09.2001		Numara		
4324	İdil	ALTIN	Bilişim Teknolojilori	110					
4544	Ayça	DAL	Bilişim Teknoloji		^		Adı		-
4566	Yakup	KUTLU	Bilişim Teknoloji			1		and the second second	-
4567	Sedat	KÖSE	Yiyecek İçecek Sil	me Gerçekleşr inecek öğreno	nedi! inin diğer alanlarda		Sovadı		
4756	Öykü	KILINÇ	Bilişim Teknoloji ka	ydının olmam	asına dikkat ediniz.	1			
5456	Sevcan	KUDRET	Gazetecilik				Bölüm		7
6554	Nurdeniz	SOYLU	Gazetecilik		ОК		Dorum		
6788	Mehmet	DENİZ	Yiyecekİçecek 📖	-			Curif		
7845	Gizem	FIRINCI	Gazetecilik	12	23.07.2000		Sinii	×	
7858	İdil	LAL	Bilişim Teknolojileri	i 12	2.11.2000		D Taulhi	10 11 2020	
8745	Zerda	DENİZ	Gazetecilik	11	17.05.2001		D. Tarini	18.11.2020	
9999	Murat	TEOMAN	Bilişim Teknolojileri	i 12	17.01.2000	1			
						1	Eklo	cil	
							EKIE	511	

Görsel 6.109: Kayıt silme hatası

Görsel 6.109'da 4324 numaralı öğrencinin notlar tablosunda notları olduğu için silinmesine izin verilmemiştir.

6.14.2. Notlar Sekmesi

Notlar sekmesinde ekleme işlemi ayrı formda yapıldığı için sadece güncelleme işlemine ihtiyaç

vard	ır.
------	-----

E-Okul Program	ור						
	Ana Sayf	a Ö	ğrenci	İşlem	leri 📰	Not İşleı	mleri 🚐 Ders İşlemleri
Not ID	Öğrenci No	Ders Kodu	1. Yazılı	2. Yazılı	3. Yazılı	1. Uygulama ^	Numara
1	2345	MAT	87	85	47	95	
2	2345	EDE	25	36	47	95	Ders Kodu
3	2345	FİZ	47	85	24	99	
4	3455	MAT	55	87	34	68	1. Yazılı
5	3455	FİZ	99	85	74	68	
6	3455	EDE	14	25	66	87	2. Yazılı
46	3456	MAT	87	56	52	66	
47	3456	EDE	95	65	32	85	3 Vazılı
48	3456	FİZ	54	65	37	98	5. 14211
50	3457	PT	25	65	32	65	4 11
51	3457	NESN	87	85	86	87	1. Uyg.
52	4324	FİZ	87	85	85	96	
54	4324	BDN	59	65	32	47	2. Uyg.
55	4544	BTT	14	56	66	56	
56	4544	EDE	55	68	95	65	1. Sözlü
57	4544	PT	54	65	65	21	
58	4567	MAT	21	65	32	74	2. Sözlü
59	4567	BDN	85	55	65	47	
60	4567	FİZ	88	58	96	31	Ortalama
< l	1.755	DTT.		1.2.5	100	>	ortainia
	1						Güncelle Sil

Görsel 6.110: Notlar sekmesi tasarımı

6.14.2.1. Notlar Sekmesi Kayıt Güncelleme

Öğrenci sekmesindeki kodların benzerleri burada da kullanılır. Önce **datagridview**'e çift tıklanarak değerler textbox'lara alınır.





Not: Güncelleme yapılırken öğrenci numarası veya ders kodu, öğrenci tablosunda ya da ders tablosunda olmayan değerlerden yazılırsa tablolar ilişkisel olmasından dolayı hata verecektir.

Kodlar neredeyse aynıdır sadece bu sekmede notlar olduğu için biraz daha fazla veri aktarımı mevcuttur. Notlar değiştirildiğinde doğal olarak ortalama da değişeceği için ortalama tekrar hesaplanmıştır. MySQL'de veri aktarıldığında ortalama değeri "**Default**" olarak "0" girilmiştir. Güncelleme işlemi yapılarak "0" olan değerlerin gerçek değerleri hesaplatılabilir. Burada try-catch yapısı da bulunmaktadır.



ri	Ders İşlemleri	leri 🚐	Not İşlem	ri 🗐 N	şlemle	enci İş	Öğr	na Sayfa	
3455	Numara	2. Uyqula^	1. Uvqulama	3. Yazılı	2. Yazılı	1. Yazılı	Ders Kodu	Öğrenci No	Not ID
rt7	Dars Kadu	87 74	95	47	85 36	87 25	MAT EDE	2345	1 2
F1Z	Ders Kouu	87	99	24	85	47	FİZ	2345	3
00	1 Vanili	74	68	34	87	55	MAT	3455	1
99	I. Yazılı	52	68	74	85	99	FIZ	3455	5
	0 V- I	25	87	66	25	14	EDE	3455	5
85	2. Yazılı	98	00	100	50	87	MAI	3450	40 17
		77	×	37	65	54	Fİ7	3456	47 18
74	3. Yazılı	32		32	65	25	PT	3457	50
		90	me İslemi Başarılı	86 Günceller	85	87	NESN	3457	51
68	1. Uva.	99	ne işieni büşeni	85	85	87	FİZ	4324	52
00	1.0,9.	85		32	65	59	BDN	4324	54
50	2 11/0	85	ОК	32	65	21	MAT	4567	55
52	2. Oyg.	12	24	95	68	55	EDE	4544	56
47	4 68-10	05	21	22	65	54	PI	4544	5/
4/	1. Soziu	35	4	65	55	85	BDN	4567	50
		65	31	96	58	88	FİZ	4567	59 60
85	2. Sozlü	99	87	32	65	54	BTT	4756	61
		85	74	65	98	25	EDE	4756	62
72,86	Ortalama	74 ~	32	65	65	54	BDN	5456	64
85 72,86	2. Sözlü Ortalama	99 85 74	87 74 32	32 65 65	65 98 65	54 25 54	BTT EDE BDN		4756 4756 5456

Görsel 6.111: Güncelleme işleminin sonucu

Not: Güncelleme işlemi sonucunda ortalama sütunu kontrol edilir.

6.14.2.2. Notlar Sekmesi Kayıt Silme

VERİ TABANI İŞLEMLERİ

Aşağıdaki kodlara göre kayıt silme işlemi gerçekleştirilir.



			•	• •				E_/+
Ders İşlemle	emleri 🚆 De	Not İşle	leri 📰	İşlem	ğrenci	a Ö	ana Sayfa	
ara	Numara	1. Uygulama (3. Yazılı	2. Yazılı	1. Yazılı	Ders Kodu	Öğrenci No	Not ID
		65	32	85	17	MAT	7845	73
Kodu	Ders Kodu	90	90	85	85	FİZ	7845	74
		85	69	85	57	BDN	7845	75
7111	1. Yazılı	56	32	65	65	NESN	7858	76
		98	99	85	85	BTT	7858	77
2111	2 Yazılı	85	65	6.3	52	FİZ	7858	78
	2. 10211	×		Uyarı	54	FİZ	8745	79
	2 Vazili				14	BDN	8745	30
2111	S. Tazin			A	54	EDE	8745	81
		cistiyonnusunuz :	remine devam etmes	-i sume isi	90	MAT	9999	32
g.	1. Uyg.				100	NESN	9999	33
		No	Yes		98	BTT	9999	34
g.	2. Uyg.			100	99	BDN	2345	37
-		50	100	100	100	BDN	9999	38
zlü	1. Sözlü	25	77	96	85	MAT	4756	39
		74	78	65	45	BDN	2345) 0
zlü	2. Sözlü	45	65	76	65	MAT	3456	92
	21 00214	55	47	87	56	FİZ	4324	93

Görsel 6.112: Notlar sekmesi kayıt silme işlemi

6.14.3. Dersler Sekmesi

Son olarak dersler sekmesine geçilir. Dersler tablosunda iki alan olduğu için işlem çok kısa sürecektir. Sadece iki alan olduğu için güncelleme işlemi yerine ekleme ve silme işlemleri de kullanılabilir (Görsel 6.113).

E-Okul Programi								
An 👔	a Sayfa 幕 Öğı	enci	İşlemle	ri 📰	Not İşle	mleri	📮 De	rs İşlemleri
Ders Kodu	Ders Adı	_						
BDN	Beden Eğitimi							
BTT	Bilişim Teknoloji Temelleri							
EDE	Edebiyat							
FÍZ	Fizik		D					
MAT	Matematik		Ders Ko	bau				
NESN	Nesne Tabanlı Programlama		_					
PI	Programlama lemelleri		Ders Ac	11				
			EM	la	cil			
			Ek		311			

Görsel 6.113: Dersler sekmesi form tasarımı 6.14.3.1. Dersler Sekmesi Kayıt Ekleme

Aşağıdaki kodlara göre ders ekleme işlemi gerçekleştirilir.

private void btnDersEkle_Click(object sender, EventArgs e)
<pre>{ kmt.Connection = bag.baglan();//Bağlantı yolu verildi. kmt.CommandText = "INSERT INTO dersler(derskodu, dersadi) VALUES (\" + txtDersKodu.Text</pre>
+ "', "' + txtDersAdi.Text + "')";//Kayıt için SQL cümlesi
kmt.ExecuteNonQuery();//Komut çalıştırıldı.
kmt.Connection.Close();//Bağlantı kapatılır.
DersYukle();//Derslerin yeni hâli yüklendi.
MessageBox.Show("Bilgiler veri tabanına başarı ile eklendi.");//İşlem başarılı mesajı verilir.
}



6.14.3.2.Dersler Sekmesi Kayıt Silme



6.15. KURULUM (SETUP) HAZIRLAMA

Hazırlanan projenin bir başka bilgisayarda kullanılabilmesi için proje "**Setup**" (**Kurulum**) hâline getirilmelidir. Kod editörü bu imkânı da sağlamaktadır.

Not: Bu tür işlemlerde diğer bilgisayarda da MySQL Server olmalıdır ve projedeki veri tabanı kurulum yapılacak bilgisayarın veri tabanına import edilmelidir. MySQL üzerinden "Export" ve "Import" işlemleri daha önce anlatılmıştır (6.8. MySQL Veri Tabanı Alma ve Yükleme).

Setup hazırlamanın adımlarına geçilir.

İlk olarak kod editörü programına Setup bileşeni kurulmalıdır. Bunun için üstteki menüden "**Exten**sion --> Manage Extensions" bölümüne girilir. Gelen pencerenin arama kısmına "Installer" veya "Setup" yazılır. Çıkan sonuçlardan "Installer Projects" seçilerek Download düğmesine basılır (Görsel 6.114). Kurulum bittikten sonra yüklemenin başlaması için kod editörü programı kapatılır.





Görsel 6.114: Projeye Setup projesi ekleme

Yükleme bittikten sonra kod editörüne Setup özelliği eklenecektir. Proje açılır ve menüden "FileAdd — New Project" ile projeye eklenecek Setup projesi seçilir. Setup görünmezse yukarıdaki arama çubuğuna "Setup" yazılabilir. Gelen pencereden Setup Project seçilerek ikinci adıma geçilir (Görsel 6.115).

Recent project templates		C#	 All platforms 	 All proje 	ct types
Console App (.NET Framework)	C#	No exact matches f	ound		
Sur L. F. A. OFF.		Other results based	on your search		
 Windows Forms App (.NET Framework) 	C#	Setup Pro	ject		New
UWPF App (.NET Framework)	C#	Create a V	Vindows Installer project to which file	es can be added	
		Web Setu Create a V	p Project Vindows Installer web project to whic	ch files can be added	New
		Create a V	ard Vindows Installer project with the aid	of a wizard.	New
			Not finding what you're I Install more tools and	looking for? features	

Görsel 6.115: Projeye Setup projesi ekleme

Gelen pencerede projeye eklenecek Setup projesinin ayarları yapılır. Burada istenen bir isim ve yer belirlenebilir. Bu, ana projeden ayrı bir bölümdür. Örneğin, Setup projesi masaüstündeki bir klasöre çıkartılmak istendiğinde bir masaüstü klasörü oluşturulup o seçilir (Görsel 6.116).



VERİ TABANI İŞLEMLERİ		
• • •		
Configure your new project		
Setup Project		
Project name		
OkulProjesiSetup		
Location		
C:\Users\Murat\Desktop\OkulProjesiSetup\		
	Back	Create

Görsel 6.116: Setup projesi ayarları

Setup projesi, projeye eklendi ve Setup projesinin ayarlarının yapılacağı kısımlar ekrana geldi. Solution Explorer penceresine dikkat edilirse Setup projesinin eklendiği görülür (Görsel 6.117). Projenin ayarlarına geçilebilir.

×) File Edit View Proj ⊙ + ⊙ 🏠 + 🏊 🔛 🗳	ect Build Del '୨ - 연 - D	bug Test ebug •	t Analyze Any CPU	Tools E	Extensions Start + 5	Window	Help	Search (Ctrl+Q)		٩	EOkulProjesi		
Server Explorer Toolbox Data Sources	File System (OkuProjesiSetup I File System on Target Me I Application Folder Uer's Desktop Uer's Programs Men) • x sqlbagla chine u	ntisi.cs	YeniKayi	t.cs	YeniKayit.c	s (Design)	F	orm I.cs (Design)	Form1.cs			Name Da Application Folder Duser's Desktop Duser's Programs Me	Type Folder Folder Folder

Görsel 6.117: Setup proje işlem penceresi

Burada ilk yapılması gereken işlem, projenin dosyalarını Setup projesine eklemektir. Sağdaki bölümde "**Application Folder**" çift tıklanarak giriş sağlandıktan sonra "Add — Project Output" seçeneği seçilir. Gelen ekranda "**Primary output**" seçilir (Görsel 6.129). Bu işlemden sonra projenin çalışması için yüklenen dosyaların tamamı Setup'a eklenecektir.

	•			•	•			•
						Add Project Output	Group	? ×
						Project:	EOkulProjesi	~
						Primary output Localized resource	es	^
		Add Create New Shorts	•	Folder		Debug Symbols Content Files		
بر ۵	5	Create New ShortCu View Cut Paste	Ctrl+X Ctrl+V	Project Output File Assembly		Source Files Documentation F XML Serialization	iles Accemhliec	>
						Configuration: Description:	(Active)	~
						Contains the DLL	or EXE built by the project.	Cancel

Görsel 6.118: Setup projesine ana projenin bileşenlerini ekleme

Setup'a eklenen ana dosyadan (.exe dosyası) iki kısayol oluşturulup bunlar kurulum yapılan bilgisayarın masaüstü ve programlar menüsüne aktarılır. Bunun için eklenen dosyalardan **"Primary out-put from EokulProjesi (Active)"** dosyasına sağ tıklanarak "**Create Shortcut** ..." seçeneği ile iki kısayol oluşturulur. Bu kısayollara proje adının verilmesi unutulmamalıdır çünkü kısayollar masaüstüne ve başlat menüsüne eklenecektir. Bu kısayollar, isimler de verildikten sonra sürükle bırak ile "**User's Desktop**" ve "**User's Programs Menu**" içine taşınır (Görsel 6.119).

 File System on Target Machine Application Folder User's Desktop User's Programs Menu 	Name BouncyCastle.Crypto.dll dte80a.olb EnvDTE.dll Google.Protobuf.dll K4os.Compression.LZ4.dll K4os.Compression.LZ4.Streams.dll	Type Assembly File Shortcut Assembly Assembly Assembly Assembly
---	--	--

Görsel 6.119: Oluşturulan kısayolların taşınması

Setup'a ikon eklenmesi işleminde öncelikle ikon dosyası da Setup içine alınmalıdır. Bunun için yine "**Application Folder**" içinde sağ tıklandıktan sonra "Add — File" seçeneği ile ikon seçilir ve Setup projesine dâhil edilir (Görsel 6.120).

lcon		?	×
File name:			
eokulprojesiikonu.ico		Browse	
Current icon:			
0			
	OK	Cancel	
	UK	Cancer	

Görsel 6.120: Setup projeye ikon eklenmesi





Belirlenen ikonun eklendiği kısayollar, propertis kısmından Icon seçeneği ile seçilir (Görsel 6.121).

	App.config 🚡 🗙 👻 🌣	Properties		→ ₽ ×
Name	Туре	E-Okul Uygulaması File li	nstallation Properties	•
E-Okul Uygulaması	Shortcut			
		(Name)	E-Okul Uygulaması	
		Arguments		
		Description		
		Folder	User's Desktop	
		lcon	(None)	\sim
		ShowCmd	(None)	
		Target	(Browse)	
		Transitive	Torse	
		WorkingFolder	Application Folder	

Görsel 6.121: Eklenen ikonun kısayollara atanması

Setup dosyasının ayarları için Solution Explorer penceresinden Setup projesi seçilir ve properties penceresine geçilir (Görsel 6.122).

Properties	- 4 ×	Properties	* ₽ X
EokulProjesiSetup Deplo	oyment Project Properties 🔹	EokulProjesiSetup Deployme	nt Project Properties 🔹
E≣ ∰↓ ≯			
AddRemovePrograms	(None)	AddRemoveProgramsIcon	(lcon)
Author	Murat	Author	MEB Öğretmeni
BackwardCompatiblell	False	BachwardCompatibleIDGen	Falce
Description		Description	Ru program ile öğrencilerinizin potlarını kay
DetectNewerInstalledV	True	Detection Detection	True
InstallAllUsers	False	Detectivewerinstalledversion	Irue
Keywords		InstallAllUsers	True
Localization	English (United States)	Keywords	eokul programı
Manufacturer	Murat	Localization	Turkish
ManufacturerUrl		Manufacturer	Milli Eğitim Bakanlığı
PostBuildEvent		ManufacturerUrl	
PreBuildEvent		PostBuildEvent	
ProductCode	{67986516-105A-41C5-A316-F5278AI	PreBuildEvent	
ProductName	EokulProjesiSetup	ProductCode	(67986516-105A-41C5-A316-E5278ADEBDC2
RemovePreviousVersio	False	ProductName	FokulProjeri
RunPostBuildEvent	On successful build	Productivanie Remove Dravious Versions	True
SearchPath		Removerreviousversions	
Subject		RunPostBuildEvent	On successful build
SupportPhone		SearchPath	
SupportUrl		Subject	
TargetPlatform	x86	SupportPhone	
Title	EokulProjesiSetup	SupportUrl	https://meb.gov.tr/
UpgradeCode	{09431D63-C170-4851-AA69-7D66CE	TargetPlatform	x86
Version	1.0.0	Title	Eokul Projesi Kurulumu
		UpgradeCode	{09431D63-C170-4851-AA69-7D66CDEBFC2
		Version	1.0.0

Görsel 6.122: Setup projesi bilgiler penceresi öncesi ve sonrası

Setup projesinin ayarları şu şekilde sıralanabilir:

- AddRemoveProgramsIcon: Program ekle / kaldır kısmında görünen simge
- Author: Programın kodlayıcısı
- DetectNewerInstalledVersion: Programın daha yeni sürümü yüklü mü kontrol et.
- InstallAllUsers: Program bütün kullanıcılar için kurulsun mu?
- Keywords: Programla ilgili anahtar kelimeler
- Localization: Bölgesel ayarlar (Localization, program Türkiye'de kullanacağı için Türkiye seçilir.)
- Manufacturer: Projeyi yapan firma adı
- ManufacturerUrl: Projeyi yapan firmanın web adresi
- ProductName: Projenin adı
- RemovePreviousVersions: Eski sürümü varsa kaldır.
- Subject: Programın kısa özeti
- SupportPhone: Program desteği için telefon
- SupportUrl: Program desteği için web adresi
- TargetPlatform: Programın desteklediği işlemci mimarisi
- Title: Programın başlığı
- Version: Programın versiyonu

NESNE TABANLI PROGRAMALAMA





Program başka bilgisayara kurulurken MySQL Server ayarları çok önemlidir. Bu ayarlar bütün bilgisayarlarda aynı değildir. O yüzden program kurulduktan sonra bu bağlantıyı değiştirmek için bazı düzenlemeler yapılmalıdır.

Önce Solution Explorer penceresine gidilip App.config dosyası açılır ve Connection String cümlesi Görsel 6.123'teki gibi **<connectionStrings></connectionStrings>** yazılarak içine eklenir.

1	<pre><?xml version="1.0" encoding="utf-8"?></pre>
2	<pre>□<configuration></configuration></pre>
3	<pre><connectionstrings></connectionstrings></pre>
4	<pre>add name="Eokul_Connection_String"</pre>
5	<pre>connectionString="Server=localhost;Database=eokul;Uid=root;Pwd=milliegitim;"/></pre>
6	

Görsel 6.123: Setup projesine Connection String eklenmesi

Daha sonra projeye referans olarak Görsel 6.124'teki gibi System.Configuration eklenir.

Reference Manager - EOkulP	rojesi				?	×
 Assemblies 				System.Conf		× •
Framework Extensions Search Results	V	Name System.Configuration System.Configuration.Install	Version 4.0.0.0 4.0.0.0			
 Projects Shared Projects COM Browse 						
			Brov	rse OK	Can	icel

Görsel 6.124: System.Configuration referansı ekleme

Connection String'in olduğu sqlbaglantisi.cs dosyasına gidilip "**using System.Configuration**" kütüphanesi eklenir, ardından **App.config** içine yazılan **Connection String**'in **Name**'i eklenir.

public MySqlConnection baglan() // class içinde baglan adında fonksiyon oluşturuldu.
{ MySqlConnection baglanti = new MySqlConnection(ConfigurationManager.ConnectionStrings["Eokul_Con- nection_String"].ConnectionString);// Bağlantı yolu
baglanti.Open();//Bağlantı açıldı.
MySqlConnection.ClearPool(baglanti);//Önceki bağlantılar temizlendi.
return (baglanti);//Çağrıldığı yere bağlantı gönderildi.
}

Connection String, kurulan bilgisayarın MySQL özelliklerine göre (ad ve şifre gibi) sonradan değiştirebilir ve bu programın doğru çalışması için çok önemlidir.

Artık Setup projesi derlenebilir. Görsel 6.125'teki gibi Setup projesine gidilerek sağ tıklanır ve Build seçeneği işaretlenir.



Görsel 6.125: Setup projesinin derlenmesi

- -

10. SINIF

Build, en başta seçilen yerde iki tane Setup dosyası oluşturacaktır (Görsel 6.126).

🛃 EokulProjesiSetup	29.11.2020 15:31	Windows Installer	9.327 KB
🔄 setup	29.11.2020 15:31	Application	554 KB

Görsel 6.126: Derleme sonucu oluşan kurulum dosyaları

Kuruluma geçilir. İstenen bir Setup dosyası çalıştırılır ve işlem adımları takip edilir. Gerekli bir bileşen kurulumda eksikse program Görsel 6.127'deki gibi uyarı verir.

Bu kurulum için .NET i	Framework sürüm v.NE	T Framework 4.7.2 gerekiy	or. Lütfen .NET
Framework'ü yükleyin	ve bu kurulumu yenide	n çalıştırınNET Framework	web üzerinder
alınabilir. Bunu şimdi ya	apmak ister misiniz?		

Görsel 6.127: Uygulamanın kurulumunda yapılan kontroller sonucu eksik bileşen yüklenmesi

Gerekli kurulum yapılıp Görsel 6.128'deki gibi adımlara devam edilir.

😥 EokulProjesi — 🗆 🗙	🙀 EokulProjesi — 🗆 🗙	🛃 EokulProjesi — 🗆 🗙
EokulProjesi Kurulum Sihirbazı'na Hoş Geldiniz	Yükleme Klasörünü Seçin	Yükleme Tamamlandı
Yükleydi, EskuPhgesi üününü biğasyamıza yüklemek için gerekli adınlarda azer yol gösterecek.	Yükleyis Eokul/Rejesi üürünü appüdali Kasöre yükleyecek. Bu kasöre yüklemek için "Ber"yi takatın. Fakâ bir klasöre yüklemek için appüge grim veya "Gözat") alalar. Basör: [C:Vhogram Files (465)Milli Eğitim Bakandği/EskulProjesi/ Gözat Dak Nam	EokuPhogen baganja yuklendi. Çıkmak için "Kapat"ı bildan.
UYARI Bu bigaayar program telf halds yasalan ve ukatarasa arlaşmalar tarafından. kourumatalad: Bu program veşa bir biblini'nin instra çoğdatmanın veşa dağılmanarın old halduk takibata konu dacatar.	EdiulPhopei ürününü kendiniz veya bu biğisəyəri kullarıan hərhangi biri için yükləyin: @ Harkes Yahaca ben Cleri Ees >	Läfen Winderes Updatel kullananis. NET Francesok ign ketik güncellegismeler olup olmadığırı denetleşin. < Giet Kepat istal

Görsel 6.128: Uygulamanın kurulum adımları

ÖĞRENME BİRİMİ





Kurulum başarıyla gerçekleştikten sonra Görsel 6.129'daki gibi simgeler oluşur.



Görsel 6.129: Uygulamanın kurulumu sonucu masaüstü ve program files klasöründe oluşan simgeler ve dosyalar Bunoktada **EOkulProjesi.exe.config** dosyasının içinde yer alan Connection String'in kurulum yapılan



Görsel 6.130: Kurulum sonrası Connection String değiştirme



kısayolu çift tıklanarak proje çalıştırılabilir.

<u> </u>	na Sa	yfa 🦹 Ö	İğrenci İşlemler	i	Not İşl	emleri	Der:	s İşlemler
Numara	Adı	Soyadı	Ders Adı	1. Yazılı	2. Yazılı	3. Yazılı	1. Uygulama	2. Uygulama 🏾 ^
9999	Murat	TEOMAN	Beden Eğitimi	100	100	100	50	50
9999	Murat	TEOMAN	Bilişim Teknoloji Temelleri	98	95	100	95	85
9999	Murat	TEOMAN	Matematik	90	95	95	98	90
9999	Murat	TEOMAN	Nesne Tabanlı Programlama	100	95	95	99	95
8745	Zerda	DENİZ	Beden Eğitimi	14	58	65	98	63
8745	Zerda	DENİZ	Fizik	54	21	65	98	91
8745	Zerda	DENİZ	Edebiyat	54	65	32	65	82
7858	İdil	LAL	Bilişim Teknoloji Temelleri	85	85	99	98	65
7858	İdil	LAL	Fizik	52	63	65	85	54
7858	İdil	LAL	Nesne Tabanlı Programlama	65	65	32	56	58
7845	Gizem	FIRINCI	Beden Eğitimi	57	85	69	85	65
7845	Gizem	FIRINCI	Fizik	85	85	90	90	98
7845	Gizem	FIRINCI	Matematik	17	85	32	65	75
6788	Mehmet	DENİZ	Edebiyat	65	74	85	65	65
6788	Mehmet	DENİZ	Fizik	32	32	20	25	70
6788	Mehmet	DENİZ	Matematik	65	32	47	52	58
6554	Nurdeniz	SOYLU	Beden Eğitimi	85	85	95	87	58
6554	Nurdeniz	SOYLU	Edebiyat	25	23	66	38	85
6554	Nurdeniz	SOYLU	Matematik	32	65	98	75	98
5456	Sevcan	KUDRET	Beden Eğitimi	54	65	65	32	74
<	1		1		1	1		× *
Kayıt	Ekle	Yenil	e Ara	ima:				

Görsel 6.131: Projenin bitmiş hâli

Görsel 6.131'de kurulumu gerçekleştirilen proje hiçbir hata vermeden çalıştı. Başka bilgisayarlara kurulum yapıldığında hem o bilgisayardaki Server ayarları **config** dosyasına eklenmeli hem de veri tabanları yeni Server'a **import** edilmelidir.



6.16. ENTITY FRAMEWORK



Görsel 6.132: Entity Framework şeması

Entity Framework (Görsel 6.132), yazılım geliştiricilerin uzun SQL sorguları yazmalarını ortadan kaldırarak bir **Object Relational Mapping** (ORM) imkânı sağlayan framework'tür. ORM ise ilişkisel veri tabanı yönetim sistemlerine doğrudan müdahale yerine nesneler aracılığı ile müdahale edilmesini sağlayan bir köprüdür. Entity Framework ile birlikte **Language Integrated Query** (LINQ) sorguları kullanılarak nesneler üzerinde güçlü bir sorgulama imkânına sahip olunur.

Entity Framework'ün temel amacı, uygulama geliştiricinin data işlemleri ile çok vakit kaybetmeden uygulama tarafına odaklanmasını sağlamaktır. Örneğin klasik ADO.NET uygulamalarında bir bağlantının açılmasından ve kapatılmasından tamamen geliştiriciler sorumludur ancak Entity Framework kullanıldığında bu tür işlemlere karışılmaz. Sorgu hazırlanır ve Entity Framework aracılığı ile database'e iletilir. Entity Framework teknolojisi, veri tabanlarıyla ilgili birçok konuda işlemleri kullanıcıların yerine yaparak işi kolaylaştırmaktadır. Entity Framework performans olarak ADO.NET teknolojisinden yavaştır. **Kısaca özetlenirse tablolar sınıflara, satırlar nesnelere ve sütunlar değişkenlere dönüşerek projeye entegre edilir.** Entity Framework yapısını kullanırken LINQ ve Lambda Expressions yapıları da kullanılır.

Lind (Language integrated query) dile entegre edilmiş sorgu olarak Türkçeye çevrilebilir. Lind C# ve VB.NET programlama dilleri ile farklı kaynaklardaki veriler üzerinde **CRUD** (Create - Read - Update - Delete) işlemleri yapılmasını sağlar. Böylelikle veri tabanı uyumsuzluklarını ortadan kaldırarak tek bir sorgu arayüzü elde edilmiş olur (Görsel 6.133 Lind şeması).

Örneğin olarak SQL dili, veri tabanından veriyi almak, kaydetmek vb. işlemlerde kullanılmak üzere yapılandırılmış bir sorgu dilidir. Linq ise koleksiyonlar, ADO.NET Data Set, XML, Web Servisler, MS SQL Server, Entity Framework ve diğer veri tabanı ve ORM teknolojileri kullanılarak veriler üzerinden CRUD işlemleri yapmak için C# ve VB.NET dilleri için oluşturulmuş bir sorgu söz dizimidir.





Görsel 6.133: LINQ şeması

Lambda Expressions (Lambda ifadeleri), yani lambda ifadeleri, verilen filtrelere ve belirlenen kriterlere göre, değişkenlere değer atamaya yarayan fonksiyonlar olarak düşünülebilir. Genellikle diziler veya entity listeleri üzerinde kolayca sorgu yapılmasını sağladığı için, SQL üzerinden yapabilecek pek çok sorgu bu fonksiyonlar sayesinde halledebilir. Önceden oluşturulan "**eokul**" veri tabanını kullanarak yapılacak proje için Görsel 6.134'teki gibi Entity Framework için bir form tasarlanır. Formu tasarlarken "dataGridView" için aşağıdaki işlemler uygulanır.

- Bütün bileşenlere isim veriniz.
 - Datagridview için şu ayarları yapınız:
 - **SelectionMode** = FullRowSelect
 - MultiSelect = False
 - **ReadOnly** = True
 - **EditMode** = EditProgrammatically
 - AutoSizeColumnsMode = Fill

m1.cs Form1.cs [Design] + X		
🖳 Entity Framework Örneği		
Öğrenci No : Adı :	Ekle	Sıralama İşlemleri Id'ye Göre
Soyadı : Bölümü :	Sil	 Ada Göre A-Z Ada Göre Z-A Doğum K->B
Sınıfı : 🗸	Güncelle	○ Doğum B->K
		Siraia

Görsel 6.134: Entity Framework form tasarımı

Form tasarlandıktan sonra projeye gerekli paketler kurulmalıdır. Bu işlem Görsel 6.135'deki gibi Solution Explorer penceresine sağ tıklandıktan sonra "**Manage NuGet Packages**" seçeneği işaretlenerek yapılır.

Solu	utio	n Expl	orer	•	ф,	x	rop
G	Θ		a 'o • 5 C # @ 🖋 🗕 🚳				perties
Sea	rch	Soluti	on Explorer (Ctrl+ş)		۶) -	
-	S	olutio	h 'EntityFrameworkUygulama' (1 of 1 project)				
	C#	Enti	tvFrameworkUvaulama				
	⊳		Build				
	⊳		Rebuild				
			Clean				
	4		Analyze and Code Cleanup				
		œ∙	Publish				
		Ψ.	- donamin				
			Scope to This				
	Þ		New Solution Explorer View				
			Add				+
		Ħ	Manage NuGet Packages				
		**					
		101	Set as Startup Project				
			Debua				•

Görsel 6.135: Nuget paket eklemek



Gelen ekranda iki adet paket kurulumu gerçekleştirilir. Bunun için arama kutusuna sırasıyla "**Entity Framework**" ve "**MySql.Data.EntityFramework**" yazılarak aşağıdaki paketler indirilir (Görsel 6.136).

.NET	EntityFramework by Microsoft Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature development and stabilization.	√6.4.4	
MySQL	MySql.Data.EntityFramework 🖉 by Oracle, 264K downloads MySql.Data.EntityFramework	v8.0.22	

Görsel 6.136: Nuget paketleri kurma

DİKKAT

Benzer isimde birçok paket yer almaktadır. Yükleme yapılırken isimlere dikkat edilmelidir.

DİKKAT

İki paket yüklemesinden kaynaklı uyumsuzluk sorunlarına karşı daha alt sürümler tercih edilebilir. Ekranlar gelmiyorsa ve en başa dönüş oluyorsa sürümler ve MySQL .NET Connector sürümü kontrol edilir. Gerekiyorsa alt sürümler kurulur. Alt sürümler, paketler yüklenmeden önce yanındaki sürüm kutusundan indirilip çıkarılabilir.

Projeye sağ tıklanarak proje son bir kez "Rebuild" edilir.

Forma **Entity Framework** eklenmelidir. Solution Explorer penceresinde projeye sağ tıklanarak "Add → New Item" yolu izlenir (Görsel 6.137).



Görsel 6.137: Entity Framework için yeni bileşen ekleme

Gelen pencerede sol taraftan Data ve sağ taraftan "**ADO.NET Entity Data Model**" seçilerek Add düğmesine basılır (Görsel 6.138).

Add New Item - EntityFrameworkUygulama				? ×
✓ Installed	Sort by:	Default 🔹 🏢 📰		Search (Ctrl+E)
✓ Visual C# Items ▷ Web	Ð	ADO.NET Entity Data Model	Visual C# Items	Type: Visual C# Items A project item for creating an ADO.NET
Windows Forms SQL Server	ſ	DataSet	Visual C# Items	Entity Data Model.
MySQL WPF	- (EF 5.x DbContext Generator	Visual C# Items	
Data Code	包	EF 6.x DbContext Generator	Visual C# Items	
P Online		Service-based Database	Visual C# Items	
	Ũ	XML File	Visual C# Items	
	A	XML Schema	Visual C# Items	
	2A	XSLT File	Visual C# Items	
Name: Model1	_			
				Add Cancel

Görsel 6.138: Projeye Entity Framework ekleme





Gelen pencerede "**EF Designer from database**" seçeneği işaretlenerek Next kutucuğuna tıklanır (Görsel 6.139).

Entity Data M	odel Wizard						×
	Choose Mode	el Contents					
What should	d the model co	ontain?					
EF Designer from database	Empty EF Designer model	Empty Code First model	Code First from database				
Creates a m settings for interact with	odel in the EF [the model, and h are generated	Designer based o l database object from the mode	on an existing its to include i il.	database. You o n the model. Th	an choose the	database conne application will	ection,
			< Previous	Next >	Finish	Car	icel

Görsel 6.139: Entity Framework Model seçim penceresi

Gelen ekranda bağlantının tanımlanması gerekmektedir. "**New Connection**" düğmesine basılır. Veri tabanının ayarları girilir, veri tabanı bağlantısı "**Test Connection**" ile test edilir, "**database name**" kısmından kullanılacak veri tabanı seçilerek "**OK**" düğmesine basılır (Görsel 6.140).

Connection Properties ? ×							
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.							
Data <u>s</u> ource:							
MySQL Database	e (MySQL Data Provider)	<u>C</u> hange					
S <u>e</u> rver name:	localhost						
<u>U</u> ser name:	root						
Password:	•••••						
	Save my password						
<u>D</u> atabase name:	eokul	~					
		Ad <u>v</u> anced					
<u>T</u> est Connectio	on OK	Cancel					

Görsel 6.140: Entity Framework veri tabanı bağlantısı



Gelen ekranda "**Yes, include the sensitive data in the connection string**" seçeneği seçilir. Burada en önemli bölüm, **eokulEntities** kısmıdır. Bu, bağlantı sonucu oluşan bir yapıdır. Projede bu isim ile veri tabanına ulaşılır (Görsel 6.141).

Entity Data Model Wizard X
Choose Your Data Connection
Which data connection should your application use to connect to the database?
localhost(eokul) V New Connection
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?
\bigcirc No, exclude sensitive data from the connection string. I will set it in my application code.
Yes, include the sensitive data in the connection string.
Connection string:
metadata=res://*/Model1.csdl res://*/Model1.ssdl ^ res://*/Model1.mst;provider=MySql.Data.MySqlClient;provider connection string="server=localhost;user id=root;database=eokul"
Save connection settings in App.Config as:
eokulEntities
< Previous Next > Finish Cancel

Görsel 6.141: Entity Framework ayarları

Gelen ekranda veri tabanına eklenecek yapılar seçilir. Tablolarla çalışılacağı için tablolar seçilir ve "**Finish**" düğmesine basılır (Görsel 6.142). Bu işlemin tamamlanması biraz zaman alacaktır.

Choose Your Database Objects and Settings Which database objects do you want to include in your model? View local detsler View of the detsler Views Pluralize or singularize generated object names Include foreign key columns in the model Import selected stored procedures and functions into the entity model Model Namespace: View Include foreign key columns in the model	Choose Your Database Objects and Settings Which database objects do you want to include in your mode!?	Entity Data Model Wizard	X
Which database objects do you want to include in your model? Image: State of the st	Which database objects do you want to include in your model? Image: State of the state of the state object in your model? Image: State of the state object in the state object object in the state		
✓ ✓ ✓ ✓ ✓ ✓ ✓	V views Pluralize or singularize generated object names Include foreign key columns in the model Import selected stored procedures and functions into the entity model	Which database objects do you want to include in your model?	
Pluralize or singularize generated object names Include foreign key columns in the model Import selected stored procedures and functions into the entity model Model Namespace:	Pluralize or singularize generated object names Include foreign key columns in the model Import selected stored procedures and functions into the entity model Model Namespace: eokulModel	 ✓ Initial Contro	
Include foreign key columns in the model Import selected stored procedures and functions into the entity model Model Namespace:	Include foreign key columns in the model Import selected stored procedures and functions into the entity model Model Namespace: eokulModel	Pluralize or singularize generated object names	
Import selected stored procedures and functions into the entity model Model Namespace:	Import selected stored procedures and functions into the entity model Model Namespace: eokulModel	Include foreign key columns in the model	
Model Namespace:	Model Namespace: eokulModel	Import selected stored procedures and functions into the entity model	
	eokulModel	Model Namespace:	
eokuliviodei		eokulModel	

Görsel 6.142: Entity Framework bileşen seçim ekranı





Tabloların ve bağlantıların olduğu diyagram ekranı gelecektir (Görsel 6.143).





Bu diyagram ile işlemler başarılı bir şekilde tamamlanır. Kodlar yazılırken ilk olarak formun **Load()** olayına listele metodu yazılır.



DİKKAT: Bu noktada projeyi çalıştırırken olabilecek bir hataya ve çözümüne ilişkin şu adımlar izlenmelidir:

 Hata mesajında veri tabanı adı istek dışında tekrarlandığı için veri tabanı ile iletişim sağlanamamaktadır. Projedeki veri tabanı adı "eokul" olduğu için burada veri tabanı adının tekrar ettiği görülmektedir (Görsel 6.144 veri tabanı adı tekrarlama hatası).

Exception Unhandled	Ψ×
System.Data.Entity.Core.EntityCommandExecutionException: Ar error occurred while executing the command definition. See the inner exception for details."	
Inner Exception MySqlException: Table 'eokul.eokul.dersler' doesn't exist	-
This exception was originally thrown at this call stack: [External Code]	•
View Details Copy Details Start Live Share session Exception Settings Break when this exception type is thrown Except when thrown from: WindowsFormsApp8.exe Open Exception Settings Edit Conditions	

Görsel 6.144: Veri tabanı adı tekrarlama hatası





Çözüm için ilk olarak projeye geçilir ve Ctrl + F yapılır. Arama penceresine "Schema="eokul"" yazılır, arama yeri olarak "Entire Solution" seçilir ve son olarak arama metodu olarak "Find All" seçilir ve ok işaretine basılır (Görsel 6.145).



Görsel 6.145: Arama penceresi ayarları

Böylece tekrara sebep olan kısımlar tespit edilir ve herhangi birine çift tıklanır.

Find Results 1	
월 월 철 또 × ×	
Find all "Schema="eokul", Find Results 1, Entire Solution, ""	
C:\Users\murat\source\repos\WindowsFormsApp8\WindowsFormsApp8\Model1.edmx(69):	<entityset entitytype="Self.dersler" name="dersler" schema="eokul" store:type="Tables"></entityset>
C:\Users\murat\source\repos\WindowsFormsApp8\WindowsFormsApp8\Model1.edmx(70):	<entityset entitytype="Self.ogrencibilgi" name="ogrencibilgi" schema="eokul" store:type="Tables"></entityset>
C:\Users\murat\source\repos\WindowsFormsApp8\WindowsFormsApp8\Model1.edmx(71):	<entityset entitytype="Self.ogrencinot" name="ogrencinot" schema="eokul" store:type="Tables"></entityset>
Matching lines: 3 Matching files: 1 Total files searched: 21	

Görsel 6.146: Arama sonucu bulunan sonuçlar

• Böylece "Model1.edmx" dosyasının kodlarına erişilir.

```
<EntitySet Name="dersler" EntityType="Self.dersler" Schema="eokul" store:Type="Tables" />
<EntitySet Name="ogrencibilgi" EntityType="Self.ogrencibilgi" Schema="eokul" store:Type="Tables" />
<EntitySet Name="ogrencinot" EntityType="Self.ogrencinot" Schema="eokul" store:Type="Tables" />
```

 Erişim sağladıktan sonra yapılması gereken "Schema="eokul" kısmından eokul bölmünü silmektir. Son durum şöyle olacaktır:

```
<EntitySet Name="dersler" EntityType="Self.dersler" Schema="" store:Type="Tables" />
<EntitySet Name="ogrencibilgi" EntityType="Self.ogrencibilgi" Schema="" store:Type="Tables" />
<EntitySet Name="ogrencinot" EntityType="Self.ogrencinot" Schema="" store:Type="Tables" />
```

Böylece artık bir sıkıntı olmadan proje çalışacaktır.

ogrenciNo	ogrenciAdi	ogrenciSoyadi	ogrenciBolumu	ogrenciSinifi	ogrencidogumtarihi	^
2345	Haydar	DAĞ	Gazetecilik	10	1.01.2002	
3455	Beste	ŞAHİNOĞULLARI	Yiyecek İçecek	9	22.11.2003	
3456	Melike	ASLAN	Yiyecek İçecek	9	22.07.2003	
3457	Ahmet	KOŞUCU	Bilişim Teknolojileri	11	22.09.2001	
4324	İdil	ALTIN	Bilişim Teknolojileri	12	8.09.2000	
4544	Ayça	DAL	Gazetecilik	12	12.12.2002	
4566	Yakup	KUTLU	Bilişim Teknolojileri	12	4.06.2002	
4567	Sedat	KÖSE	Yiyecek İçecek	10	12.12.2002	
4756	Öykü	KILINÇ	Bilişim Teknolojileri	12	13.04.2000	
5456	Sevcan	KUDRET	Gazetecilik	9	23.12.2003	
6554	Nurdeniz	SOYLU	Gazetecilik	12	27.08.2000	
0700	N4-1	DENIZ	Marca - Infance - In	0	17 10 0000	- Y

Görsel 6.147: Entity Framework'ün çalışması





Artık projeye kayıt eklenebilir.

BİRİMİ

private void btnEkle_Click(object sender, EventArgs e)
{
try
{
//projemize veri tabanını Entity Framework olarak ekledikten sonra
//tablolar sınıflara dönüşür. ogrencibilgi tablosundan
//bir nesne türetilir.
ogrencibilgi ogrenciekle = new ogrencibilgi();
ogrenciekle.ogrenciNo = Convert.ToInt32(txtOgrNo.Text);
ogrenciekle.ogrenciAdi = txtOgrAd.Text;
ogrenciekle.ogrenciSoyadi = txtOgrSoyad.Text;
ogrenciekle.ogrenciBolumu = txtOgrBolum.Text;
ogrenciekle.ogrenciSinifi = Convert.ToSByte(cmbSinif.Text);
ogrenciekle.ogrencidogumtarihi = dateTimePicker1.Value.Date;
ogrenci_ent.ogrencibilgi.Add(ogrenciekle);//Nesneye değerleri atadıktan sonra tek satır ile
veri tabanına eklenir.
ogrenci_ent.SaveChanges();//Değişiklikler kaydedilir.
MessageBox.Show(``Öğrenci Eklendi.");
dataGridView1.DataSource = ogrenci_ent.ogrencibilgi.ToList();//Datagrid yenilenir.
}
catch
{
MessageBox.Show(``Lütfen Boş alan geçmeyiniz.\nLütfen doğru değerler giriniz.\nNo benzer-
siz olmalıdır.");
}
}

Görüldüğü gibi kayıt eklerken ADO.Net'teki gibi veri tabanı bağlantısı açmak, SQL komutu yazmak ya da diğer işlemlere gerek kalmamıştır. Üstelik **ogrencibilgi** tablosundan türetilen ogrenciekle nesnesi ile direkt olarak tablodaki alanlarla etkileşime geçilebilecektir.









Entity Framework Ö	rneği					- 0
Öğrenci No	: 2220					Sıralama İşlemleri
Adı	· Murat				Ekle	● ld've Göre
, (4)	Marat					,, ,
Sovadı		FD				O Ada Göre A-Z
	TORTOEV				Sil	• Ada Göre Z-A
Bölümü	: Yivecek İo	ecek				
						O Dogum K->B
Sınıfı	: 11	~		Gü	incelle	○ Doğum B->K
Doğum Tarih	ii: 22 Şuba	t 2002 Cum:	×			Sırala
ogrenciNo	ogrenciAdi	ogrenciSoyad Ö	ğrenci Eklendi.	u	ogrenciSinifi	ogrencidogumtarihi
2345	Haydar	DAĞ	-		10	1.01.2002
455	Beste	ŞAHİNOĞULL		ĸ	9	22.11.2003
456	Melike	ASLAN	ОК	ĸ	9	22.07.2003
	Ahmet	KOŞUCU		bjileri	11	22.09.2001
3457	Idil	ALTIN	Bilişim Teknol	lojileri	12	8.09.2000
1457 1324		DAL	Gazetecilik		12	12.12.2002
1457 1324 1544	Ayça	RUTUL	Bilişim Teknol	ojileri	12	4.06.2002
1457 1324 1544 1566	Ayça Yakup	KUTLU	Misseels is a se	1.	10	12.12.2002
1457 1324 1544 1566 1567	Ayça Yakup Sedat	KUTLU KÖSE	Yiyecek İçece	k Iojilori	12	13.04.2000
1457 1324 1544 1566 1567 1756	Yakup Sedat Öykü	KUTLU KÖSE KILINÇ	Yiyecek İçece Bilişim Teknol	ek lojileri	12	23 12 2003
1457 1324 1544 1566 1567 1756 1456 1554	Yyça Yakup Sedat Öykü Sevcan	KUTLU KÖSE KILINÇ KUDRET	Yiyecek İçece Bilişim Teknol Gazetecilik	k lojileri	12 9 12	13.04.2000 23.12.2003 27.08.2000

Görsel 6.149: Entity Framework ile kayıt ekleme

Kayıt silme işlemine geçilir.





🖳 Entity Framework Ö	rneği				- 🗆 X	
Öğrenci No	: 2220				Sıralama İşlemleri	
				Ekle		
Adı	: Murat	Murat			Id ye Gore	
Sound					O Ada Göre A-Z	
Soyau	· YURI	YURTSEVER		Sil	○ Ada Göre Z-A	
Bölümü	: Yivec	Vivecek İcecek				
	- IJOO	on içobon			O Dogum K->B	
Sınıfı	: 11	~		Güncelle	○ Doğum B->K	
				Guncelle		
Doğum Tarih	i: 22 Ş	ubat 2002 Cuma	a 📭		Sırala	
			×			
ogrenciNo	ogrenciAdi	ogrenciSovadi	11	ogrenciSi	nifi ogrencidogumtarihi ^	
2220	Murat	YURTSEVER	Dğrenci silindi. 🛛 🕅	11	22.02.2002	
2345	Haydar	DAĞ		10	1.01.2002	
3455	Beste	ŞAHİNOĞULLA	OK k	9	22.11.2003	
3456	Melike	ASLAN	k	9	22.07.2003	
3457	Ahmet	KOŞUCU	Bilişim Teknoloj	ileri 11	22.09.2001	
4324	İdil	ALTIN	Bilişim Teknoloj	ileri 12	8.09.2000	
4544	Ayça	DAL	Gazetecilik	12	12.12.2002	
4566	Yakup	KUTLU	Bilişim Teknoloj	ileri 12	4.06.2002	
4567	Sedat	KÖSE	Yiyecek İçecek	10	12.12.2002	
4756	Öykü	KILINÇ	Bilişim Teknoloj	ileri 12	13.04.2000	
5456	Sevcan	KUDRET	Gazetecilik	9	23.12.2003	
LOSEA	Mundania	COVUU	Constantille	10	27.02.2000	
Arama:						

Görsel 6.150: Entity Framework ile öğrenci silme

Burada farklı olarak var türünde bir lambda cümlesi görülmektedir. Silme ve güncelleme işlemlerinde mutlaka "**Where**" kullanılarak sınırlandırılma yapılmalıdır. Burada da yapılan işlem budur.

Güncelleme işlemine geçilir. Önce **datagridview**'in **celldoubleclik** olayına çift tıklanıldığı zaman bilgilerin textboxlara girilmesi için kodlar yazılır.

private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
 txtOgrNo.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
 txtOgrAd.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
 txtOgrSoyad.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
 txtOgrBolum.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
 cmbSinif.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();




Şimdi güncelleme kodları yazılır. private void btnGuncelle_Click(object sender, EventArgs e) { try { if (dataGridView1.SelectedRows.Count != 0) { int secilen_ogrencino = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value.ToString()); //Lambda yapısı ile sorgu cümlesi yazılır. var ogrenciguncelle = ogrenci_ent.ogrencibilgi.Where($w = > w.ogrenciNo = = secilen_ogrencino$). FirstOrDefault(); ogrenciguncelle.ogrenciNo = Convert.ToInt32(txtOgrNo.Text); ogrenciguncelle.ogrenciAdi = txtOgrAd.Text; ogrenciguncelle.ogrenciSoyadi = txtOgrSoyad.Text; ogrenciguncelle.ogrenciBolumu = txtOgrBolum.Text; ogrenciguncelle.ogrenciSinifi = Convert.ToSByte(cmbSinif.Text); ogrenciguncelle.ogrencidogumtarihi = dateTimePicker1.Value.Date; ogrenci_ent.SaveChanges();//Değişiklikler kaydedilir. MessageBox.Show("Öğrenci Güncellendi."); dataGridView1.DataSource = ogrenci_ent.ogrencibilgi.ToList(); } else MessageBox.Show("Lütfen güncellenecek kayda çift tıklayınız."); } catch { MessageBox.Show("Lütfen güncellenecek kayda çift tıklayınız.\nLütfen alanları kontrol ediniz."); } }

🖶 Entity Framework Ö	meği							- 0	×
Öğrenci No	: 6554					Ekle	Sıra	lama İşlemleri	
Adı	: Nurdeniz						0	🖻 ld'ye Göre	
							0	Ada Göre A-Z	
Soyadı	: SOYLU					Sil			
Bölümü	Cozetacilik					011			
Donanna	. dazetecink							Dogum K->B	
Sınıfı	: 12	~			G	üncelle	C	Doğum B->K	
Doğum Tarih	ii : 27 Ağustos	2000 P	azar	 ▼ ×				Sırala	
ogrenciNo	ogrenciAdi	ogrenciSoya	8×	o an an line al l		ogrenciSini	fi	ogrencidogumtarihi	^
2345	Haydar	DAĞ	Ogrena	Guncellendi.		10		1.01.2002	
3455	Beste	ŞAHİNOĞUL			9			22.11.2003	
3456	Melike	ASLAN		ОК		9		22.07.2003	
3457	Ahmet	коşиси			eri	11		22.09.2001	
4324	İdil	ALTIN		Bilişim Teknoloj	nolojileri 12			8.09.2000	
4544	Ayça	DAL		Gazetecilik	zetecilik 12			12.12.2002	
4566	Yakup	KUTLU		Bilişim Teknoloj	Bilişim Teknolojileri 12			4.06.2002	
4567	Sedat	KÖSE		Yiyecek İçecek		10		12.12.2002	_
4756	Öykü	KILINÇ		Bilişim Teknoloj	Bilişim Teknolojileri			13.04.2000	_
5456	Sevcan	KUDRET		Gazetecilik		9		23.12.2003	
6554	Nurdeniz	SOYLU		Gazetecilik	Gazetecilik			27.08.2000	
0700	Makanat	DENÍZ		Mine anti in a anti		0		17 10 0000	
Arama:									





	VERİ TABANI İŞLEMLERİ											
	•				•	۰			;	/		

Böylece temel işlemler (ekleme, listeleme, güncelleme, silme - CRUD) yapılmış olur. Arama işlemine geçilir. Bu işlem arama textbox'ının **textchanged** olayına yazılır.

, , , , , , , , , , , , , , , , , , , ,	5	-	, ,
private void txtArama_TextChanged(object sender, EventArgs e)		
{			
dataGridView1.DataSource = renciSoyadi == txtArama.Text).ToLis	= ogrenci_ent.ogrencibilgi.Where(w = t();	> w.ogrer	nciAdi == txtArama.Text w.og-
}			

🖳 Entity Framework Ör	neği						– 🗆 ×
Öğrenci No	:						Sıralama İşlemleri
Adı	:				Ek	le	● ld'ye Göre
Soyadı	:				s	il	 Ada Göre A-Z Ada Göre Z-A
Bölümü	:						○ Doğum K->B
Sınıfı	:	Güncelle			○ Doğum B->K		
Doğum Tarihi	i: 11	Şubat	2021 Perşembe				Sırala
ogrenciNo	ogrenci	Adi	ogrenciSoyadi	ogrenciBolu	ımu	ogrenciSin	nifi ogrencidogumtarihi
4324	İdil		ALTIN	Bilişim Tekno	olojileri	12	8.09.2000
7858	İdil		LAL	Bilişim Tekno	olojileri	12	2.11.2000
Arama: idil]			

Görsel 6.152: Lambda yapısı ile arama işlemi



•





Son olarak LINQ yapısı kullanılarak sıralama işlemleri yapılır.

```
private void btnSirala_Click(object sender, EventArgs e)
              {
                      if (rdbIdSirala.Checked==true)
                       {
                             //Numaraya göre sıralama
                              List<ogrencibilgi> nosiraliliste = ogrenci_ent.ogrencibilgi.OrderBy(p => p.ogrenciNo).To-
List();
                             dataGridView1.DataSource = nosiraliliste;
                      }
                      else if (rdbAZ.Checked==true)
                       {
                             //Ada göre küçükten büyüğe doğru sıralama
                              List<orrectional control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control control cont
List();
                             dataGridView1.DataSource = azsiraliliste;
                      }
                      else if (rdbZA.Checked == true)
                       {
                             //Ada göre büyükten küçüğe sıralama
                             List<ogrencibilgi> zasiraliliste = ogrenci_ent.ogrencibilgi.OrderByDescending(p => p.ogren-
ciAdi).ToList();
                             dataGridView1.DataSource = zasiraliliste;
                      }
                      else if (rdbDogum.Checked == true)
                       {
                             //Doğum tarihine göre küçükten büyüğe doğru sıralama
                             List<ogrencibilgi> dogumkb = ogrenci_ent.ogrencibilgi.OrderBy(p => p.ogrencidogumtari-
hi).ToList();
                             dataGridView1.DataSource = dogumkb;
                      }
                      else
                       {
                             //Doğum tarihine göre büyükten küçüğe doğru sıralama
                             List<ogrencibilgi> dogumbk = ogrenci_ent.ogrencibilgi.OrderByDescending(p => p.ogrenci-
dogumtarihi).ToList();
                             dataGridView1.DataSource = dogumbk;
                      }
              }
```



/	•	•	•			
Entity Framework	Örneği				>	<
Öğrenci No	2792				Sıralama İşlemleri	
-				Ekle		
Adı	: Havva				Id'ye Göre	
Sound	- DEMIN	- 5			O Ada Göre A-Z	
Soyau	DEMIRE	=R		Sil	○ Ada Göre Z-A	
Bölümü	Gazetec	ilik			Doğum K >R	
					© Doguiii К->в	
Sınıfı	: 12	~	Gi	üncelle	O Doğum B->K	
Doğum Tar	rihi : 27 Ağus	tos 2000 Pazar	·		Sırala	
Doğum Tar	r ihi : 27 Ağus ogrenciAdi	tos 2000 Pazar ogrenciSoyadi	ogrenciBolumu	ogrenciSinifi	Sırala	^
Doğum Tar ogrenciNo 999	rihi : 27 Ağus ogrenciAdi Murat	tos 2000 Pazar ogrenciSoyadi TEOMAN	ogrenciBolumu Bilişim Teknolojileri	ogrenciSinifi 12	Sırala ogrencidogumtarihi 17.01.2000	^
Doğum Tar ogrenciNo 999 756	rihi : 27 Ağus ogrenciAdi Murət Öykü	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri	ogrenciSinifi 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000	^
Doğum Tar ogrenciNo 999 756 845	rihi : 27 Ağus ogrenciAdi Muret Öykü Gizem	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik	ogrenciSinifi 12 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000	^
Doğum Tar ogrenciNo 999 756 845 554	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Gazetecilik	ogrenciSinifi 12 12 12 12 12	ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000	^
Doğum Tar ogrenciNo 999 756 845 554 324	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz İdil	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Gazetecilik Bilişim Teknolojileri	ogrenciSinifi 12 12 12 12 12 12 12	ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 8.09.2000	^
Doğum Tar 999 756 845 554 324 858	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz Idil Idil	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ KILINÇ FIRINCI SOYLU ALTIN LAL	ogrenciBolumu Bilişim Teknolojileri Gazetecilik Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12	Ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 8.09.2000 2.11.2000	^
Doğum Tar 999 756 845 554 324 858 745	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz İdil İdil Zerda	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ	ogrenciBolumu Bilişim Teknolojileri Gazetecilik Gilişim Teknolojileri Bilişim Teknolojileri Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 27.08.2000 2.11.2000 17.05.2001	
Doğum Tar ogrenciNo 999 756 845 554 324 858 745 457	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz İdil İdil Zerda Ahmet	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ KOŞUCU	ogrenciBolumu Bilişim Teknolojileri Gazetecilik Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri Bilişim Teknolojileri Bilişim Teknolojileri	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 8.09.2000 2.11.2000 17.05.2001 22.09.2001	
Doğum Tar ogrenciNo 999 756 845 554 324 858 858 745 457 345	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz Idil Idil Zerda Ahmet Haydar	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ KOŞUCU DAĞ	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 27.08.2000 2.11.2000 17.05.2001 22.09.2001 1.01.2002	
Doğum Tar ogrenciNo 999 756 845 554 324 858 858 745 457 345 345 566	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz Idii Idii Idii Zerda Ahmet Haydar Yakup	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ KOŞUCU DAĞ KUTLU	ogrenciBolumu Bilişim Teknolojileri Gazetecilik Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik	ogrenciSinifi 12 12 12 12 12 12 12 12 11 10 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 27.08.2000 2.11.2000 17.05.2001 22.09.2001 1.01.2002 4.06.2002	
Doğum Tar ogrenciNo 999 756 845 554 324 858 324 858 745 457 345 556 566 544	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz Idii Idii Idii Zerda Ahmet Haydar Yakup Ayça	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ KOŞUCU DAĞ KUTLU DAL	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12 12 12 11 10 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 27.08.2000 2.11.2000 17.05.2001 22.09.2001 1.01.2002 4.06.2002 12.12.2002	
Doğum Tar ogrenciNo 999 756 845 554 324 858 745 457 345 457 345 566 544	rihi : 27 Ağus ogrenciAdi Murat Öykü Gizem Nurdeniz Idil Idil Idil Zerda Ahmet Haydar Yakup Ayça	tos 2000 Pazar ogrenciSoyadi TEOMAN KILINÇ FIRINCI SOYLU ALTIN LAL DENIZ KOŞUCU DAĞ KUTLU DAL	ogrenciBolumu Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri Gazetecilik Bilişim Teknolojileri	ogrenciSinifi 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 11 10 12 12 12	Sırala ogrencidogumtarihi 17.01.2000 13.04.2000 23.07.2000 27.08.2000 8.09.2000 2.11.2000 17.05.2001 22.09.2001 1.01.2002 4.06.2002 12.12.2002 10.12.002	<

-

🚽 Entity Framework Ör	rneği					- 0
Öğrenci No	:	3819				Sıralama İşlemleri
Adı	:	Mustafa			Ekle	○ ld'ye Göre
Soyadı	:	KEMAL			Sil	 Ada Göre A-Z Ada Göre Z-A
Bölümü	:	Bilişim Tek	nolojileri		0	○ Doğum K->B
Sınıfı	:	12	~	G	äüncelle	● Doğum B->K
Doğum Tarih	ii :	17 Ocak 2	2003 Pazartes	și ∎•		Sırala
ogrenciNo	ogr	enciAdi	ogrenciSoyadi	ogrenciBolumu	ogrenciSinifi	ogrencidogumtarihi
5456	Seve	can	KUDRET	Gazetecilik	9	23.12.2003
0/88	Deed	met		Yiyecek içecek	9	17.12.2003
2450	Moli	le		Yiyeeek leesek	9	22.11.2003
4544	Aver	ke			12	12 12 2003
4567	Sed	at	t KÖSE		10	12 12 2002
4566	Yakı			Bilisim Teknolojileri	12	4 06 2002
2345	Hav	dar	DAĞ	Gazetecilik	10	1.01.2002
3457	Ahm	et	KOŞUCU	Bilişim Teknolojileri	11	22.09.2001
8745	Zero	la	DENIZ	Gazetecilik	11	17.05.2001
0740			LAL	Bilişim Teknolojileri	12	2.11.2000
7858	Ídil					

Görsel 6.154: LINQ ile doğum tarihine göre büyükten küçüğe sıralama işlemi

Burada seçilen radiobutton'a göre sıralama işlemleri tek satır kod ile yapılmaktadır. ADO.NET'te bu işlemler daha fazla kod ile sağlanmaktadır fakat unutulmaması gereken husus ADO.NET'te kodlarla çalışmadan diğer yapılara düzgün geçiş yapılamayacağıdır.

10. SINIF



	ÖLÇME VE DEĞERLENDİRME - 6
L. Ver	i tabanları olmasaydı bugün ne tür sorunlar yaşanırdı? Araştırınız.
2. Aşa	ğıdaki SQL komutlarının sonuçlarını yazınız.
Α.	insert into yazar(yazarad,yazarsoyad) values('İdil','İMSİYAT');
В.	select * from kitap where kitapadi like `_p%';
С.	select * from ogrenci where sinif='10A' order by ogrno desc;
D.	select ograd,ogrsoyad,sinif,cinsiyet from ogrenci where (sinif='9A' and cinsiyet='E') or (sinif='9E and cinsiyet='K');
Ε.	select * from kitap where sayfasayisi between 50 and 200;
8. Aşa	ğıda belirtilen durumları SQL komutları ile yazınız.
F.	Öğrenci tablosunda doğum yılı 1989 olan öğrencileri listeleyiniz.
G.	9A sınıfındaki tüm öğrencileri 10A sınıfına aktarınız.
Н.	Öğrenci tablosundaki adı 'A' harfi ile başlayan öğrencileri listeleyiniz.
I.	Öğrenci tablosunda 10A veya 10B sınıfındaki öğrencilerin adını, soyadını ve sınıfını listeleyiniz.
J.	Öğrenci tablosunda 10A sınıfı ve kız öğrencileri listeleyiniz.
i. Bir up bu 5. Ver arını	e-ticaret sitesini inceleyiniz. Urunler için, müşteriler için, siparişler için tablolar oluşt ınların bağlantılarını yapınız. i tabanına kayıtlar ekleyiniz ve bu kayıtlar üzerinde insert, delete, update, select komu kullanınız.
5. Örn	lek bir Connection String yazınız.
5. Örn 7. AD().NET ve Entity Framework yapılarını karşılaştırınız.
5. Örn 7. AD(D.NET ve Entity Framework yapılarını karşılaştırınız.
5. Örn 7. AD(D.NET ve Entity Framework yapılarını karşılaştırınız.
5. Örn 7. AD(D.NET ve Entity Framework yapılarını karşılaştırınız.
5. Örn 7. AD(D.NET ve Entity Framework yapılarını karşılaştırınız.
5. Örn 7. AD(D.NET ve Entity Framework yapılarını karşılaştırınız.



•

CEVAP ANAHTARLARI

Öğrenme Birimi Cevap Anahtarı - 1

- **1.** Y D D Y D D Y
- **2.** using ToolBox veya Araç Kutusu % ToString()
- **3.** Kaydet namespace
- 4. Değişken isimlerinde boşluk karakteri kullanılmaz. Değişken isimleri sayı ile başlamaz. Doğru tanımlanmıştır. Doğru tanımlanmıştır. Doğru tanımlanmıştır. Ondalık sayılar int veri türünde tanımlanamaz. char olarak tanımlı değişkenlere tek tırnak içine alınmış sadece bir karakter atanabilir. Doğru tanımlanmıştır.
- 5. 3,2 sayısı mesaj olarak görülür.
- 6.

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi;
    double sonuc;
    sayi = Convert.ToInt32(textBox1.Text);
    sonuc = sayi * 0.18;
```

MessageBox.Show(sonuc.ToString());

7.

}

```
private void button1_Click(object sender, EventArgs e)
{
    int yariCap;
    double alan,cevre,piSayisi;
    piSayisi = 3.14;
    yariCap = Convert.ToInt32(textBox1.Text);
    alan = piSayisi * yariCap * yariCap;
    cevre = 2 * piSayisi * yariCap;
    MessageBox.Show("Dairenin alanı=" + alan.ToString());
    MessageBox.Show("Dairenin çevresi=" + cevre.ToString());
}
```





Öğrenme Birimi Cevap Anahtarı - 2

- **1.** if(a > b && a > c)
- **2.** if(a < b || a < c)
- **3.** Y Y D D Y
- **4.** 3 6 9 12 15 17 18

5. 1 – 2 – 4 – 5 – 7 – 8 – 10 – 11 – 13 – 14

- **6.** 1. tur sonuç=1
 - 2. tur sonuç=2
 - 3. tur sonuç=6
 - 4. tur sonuç=24
 - 5. tur sonuç=120
 - 6. tur sonuç=720

7. Kodlar çalıştırılmaz ve program başlamaz.

Açıklama: Kodlar derlenirken Hata Listesi panelinde "string türü örtülü olarak int türüne dönüştürülemez." mesajı verir ve kodlar çalıştırılmaz çünkü yapılan hata, derleme hatasıdır. Bu tip hatalar, programın başlatılmasını engelleyen çok kritik hatalardır.

8.

```
private void button1_Click_1(object sender, EventArgs e)
  {
     int sayi1, sayi2, sayi3;
     sayi1 = Convert.ToInt32(textBox1.Text);
     sayi2 = Convert.ToInt32(textBox2.Text);
     sayi3 = Convert.ToInt32(textBox3.Text);
     if(sayi1>sayi2 && sayi1 > sayi3)
     {
        MessageBox.Show("İlk girilen sayı en büyüktür.");
     }
     if(sayi2>sayi1 && sayi2 > sayi3)
     {
        MessageBox.Show("İkinci girilen sayı en büyüktür.");
     }
     if (sayi3 > sayi1 \&\& sayi3 > sayi2)
     {
        MessageBox.Show("Son girilen sayı en büyüktür.");
     }
  }
```



IO. SINIF

```
Öğrenme Birimi Cevap Anahtarı - 3
```

```
1.
class Televizyon
{
   private int sesSeviyesi;
   private double ekranBoyutu;
   private string goruntuTeknolojisi;
   public int SesSeviyesi
      get
         return sesSeviyesi;
      }
      set
      {
         sesSeviyesi = value;
      }
   }
   public double EkranBoyutu
      get
      {
         return ekranBoyutu;
      }
      set
     ł
         ekranBoyutu = value;
      }
  }
   public string GoruntuTeknolojisi
      get
{
         return goruntuTeknolojisi;
      set
      {
         goruntuTeknolojisi = value;
      }
   }
}
2.
private class Bilgisayar
  double ram;
  public double RAMKapasitesi
   {
      get { return ram; }
      set { ram = value; }
  string cpu;
   public string CPU
      get { return cpu; }
set { cpu = value; }
  double hd;
  public double HDKapasitesi
      get { return hd; }
      set { hd = value; }
}
```

```
3.
```

class Televizyon { public int SesSeviyesi { get; set; } public double EkranBoyutu { get; set; } public string GoruntuTeknolojisi { get; set; } } 4. class Bilgisayar { public double RAMKapasitesi { get; set; } public string CPU { get; set; } public double HDKapasitesi { get; set; } } 5. class Televizyon { public int SesSeviyesi { get; set; } public double EkranBoyutu { get; set; } public string GoruntuTeknolojisi { get; set; } bool gucAcik = false; int kanalNo = 1; public void GucAc() { gucAcik = true; } public void GucKapat() { gucAcik = false; } public void KanalDegistir(int kanalNo) { this.kanalNo = kanalNo; } public int SesSeviyesiOku() ł return SesSeviyesi; } }

10. <u>sinif</u>•

```
6.
                                                       7.
class Televizyon
                                                       public: Sınıf ögelerine dışarıdan erişim sağlamak
                                                        için kullanılır.
   public int SesSeviyesi { get; set; }
                                                        private: Bu tür ögelere sadece sınıf içinden erişile-
   public double EkranBoyutu { get; set; }
                                                       bilir.
   public string GoruntuTeknolojisi { get; set; }
                                                       8.
   bool qucAcik = false;
                                                       abstract class Televizyon
   int kanalNo = 1;
                                                       {
   public int KanalŃo
                                                           protected int SesSeviyesi { get; set; }
                                                           public double EkranBoyutu { get; set; }
      get { return kanalNo; }
                                                           public string GoruntuTeknolojisi { get; set; }
   public void GucAc()
                                                           bool gucAcik = false;
                                                           int kanalNo = 1;
      qucAcik = true;
                                                           public int KanalNo
                                                           {
   public void GucKapat()
                                                              get { return kanalNo; }
      gucAcik = false;
                                                           }
                                                           public abstract void GucAc();
   public void KanalDegistir(int kanalNo)
                                                           public abstract void GucKapat();
      this.kanalNo = kanalNo;
                                                           public void KanalDegistir(int kanalNo)
   public void KanalNoArtir()
                                                           ł
                                                              this.kanalNo = kanalNo;
      kanalNo++;
                                                           }
                                                           public void KanalNoArtir()
   public void KanalNoArtir(int artis)
                                                           {
      kanalNo += artis;
                                                              kanalNo++;
                                                           }
   public void KanalNoAzalt()
                                                          public void KanalNoArtir(int artis)
                                                           {
      kanalNo--;
   }
                                                              kanalNo += artis;
   public void KanalNoAzalt(int azalis)
                                                           }
                                                           public void KanalNoAzalt()
      kanalNo -= azalis;
                                                           {
   }
                                                              kanalNo--;
   public int SesSeviyesiOku()
                                                           }
                                                           public void KanalNoAzalt(int azalis)
      return SesSeviyesi;
                                                           ł
                                                              kanalNo -= azalis;
}
                                                           }
private static void Main(string[] args)
                                                           public int SesSeviyesiOku()
   Televizyon tv = new Televizyon();
                                                           {
   tv.KanalDegistir(20);
                                                              return SesSeviyesi;
   System.Console.WriteLine(tv.KanalNo);
                                                           }
                                                        }
   tv.KanalNoArtir();
                                                       class AkilliTelevizyon : Televizyon
   System.Console.WriteLine(tv.KanalNo);
                                                        ł
                                                           public string IsletimSistemi { get; set; }
   tv.KanalNoArtir(5);
                                                           public override void GucAc()
   System.Console.WriteLine(tv.KanalNo);
  tv.KanalNoAzalt();
                                                           }
   System.Console.WriteLine(tv.KanalNo);
                                                          public override void GucKapat()
   tv.KanalNoAzalt(3);
   System.Console.WriteLine(tv.KanalNo);
}
```

10. <u>sinif</u>

9.

```
Televizyon sınıfına statik bir alan eklenir.
public static string Marka = "...";
```

10.

Televizyon sınıfı sealed olarak tanımlanır. sealed class Televizyon

11.

Sınıf içindeki bir değişkeni, dış dünyaya kapatıp sadece sınıf içinde kullanılabilir kılmak için özellik private şeklinde tanımlanmalıdır.

12. Sınıf içindeki bir değişkeni, dış dünyaya kapatıp sadece sınıf içinde ve bu sınıftan türetilen alt sınıflarda kullanılabilir kılmak için özellik protected şeklinde tanımlanmalıdır.

13.

Sınıf içindeki bir değişkeni her yerden erişilebilir kılmak için özellik public şeklinde tanımlanmalıdır.

14. 8

15.

Nesne oluşturulurken otomatik çalışan metot:

Yapıcı metot (Constuctor)

- Sınıf adı ile aynı ada sahip olmalıdır.
- Soyut tanımlanmamalıdır.
- Sadece bir tane statik yapıcı metot olabilir.
- Dönüş tipi olamaz (void dâhil).
- Aşırı yüklenebilir. Nesne yok edilirken otomatik çalışan metot: Yıkıcı metot (Destructor)
- Sadece bir tane olabilir (Aşırı yüklenemez).
- Sınıf adı ile aynı ada sahip olmalıdır.
- Adı ~ (tilde) karakteri ile başlamalıdır.
- Dönüş tipi olamaz (void dâhil).

16. 0 & 125

17.

Sınıftan nesne oluşturmadan doğrudan sınıf adı ile sınıf ögelerine erişim için kullanılır.

18.

Sınıfa ilk erişim sağlandığında çalıştırılır.

19.

ARAYÜZLER	SOYUT SINIFLAR
Bir sınıf birden fazla arayüzden türetilebilir.	Bir sınıf sadece tek bir soyut sınıftan türetilebilir.
Sadece boş (gövdesi olmayan) metotlar tanımla- nabilir.	Hem normal metot hem de boş metotlar tanımla- nabilir.
Çoklu kalıtım özelliği sağlar.	Çoklu kalıtım özelliği sağlamaz.
Tüm ögeler public olarak kabul edilir.	Ögeler public olmak zorunda değildir.
Yapıcı metot içeremez.	Yapıcı metot içerebilir.
Statik ögeler barındıramaz.	Statik ögeler barındırabilir.



10. SINIF

```
20.
interface IGuc
{
  void GucAc();
  void GucKapat();
}
class Televizyon : IGuc
{
  // ...
  public void GucAc()
   {
   }
  public void GucKapat()
   {
  }
}
class Bilgisayar : IGuc
{
  // ...
  public void GucAc()
   {
   }
  public void GucKapat()
   {
  }
}
```

Öğrenme Birimi Cevap Anahtarı - 4

- **1.** int[], numaralar
- **2.** string[],sehirler, string[81]
- **3.** 1
- **4.** B
- 5. A
- 6. D 7. C
- **8.** E
- 9. A
- **10.** B
- **11.** A
- **12.** E
- **13.** D
- **14.** B
- **15.** A
- **16.** B
- 17. E 18. D
- **19.** C





Öğrenme Birimi Cevap Anahtarı - 5

- 1. Arabirim
- 2. Application.Run
- 3. System.Windows.Forms

4.

CenterToScreen				
FormClosed				
ControlBox				
Load				
AcceptButton				
Show				

- **5.** E
- **6.** A
- **7.** C
- 8. B
- 9. D
- **10.** E

Öğrenme Birimi Cevap Anahtarı - 6

- 2.
- A. Yazar isimli veri tabanına yazar ekler.
- B. kitap tablosunda kitap adının 2. harfi p olan kayıtları getirir.
- C. ogrenci tablosundan sınıfı 10-A olanları okul numarasına göre azalan olarak listeler.
- D. öğrenci tablosundan sınıfı 9A ve cinsiyeti erkek olanlar veya sınıfı 9B ve cinsiyeti kız olanları listeler.
- E. kitap tablosundan sayfa sayısı 50 ile 100 arasında olanları listeler.

3.

- A. select * from ogrenci where dtarih between '01/01/1989' and '12/31/1989';
- **B.** update ogrenci set sinif='10A' where sinif='9A';
- C. select * from ogrenci where ograd like 'A%';
- D. select ograd, ogrsoyad, sinif from ogrenci where sinif='10A' or sinif='10B';
- E. select * from ogrenci where cinsiyet='K' and sinif='10A';



10. <u>sinif</u>

GÖRSEL KAYNAKÇA

• ••

:./

KİTAP KAPAĞI GÖRSELLERİ							
Shutterstock	https://www.sh	https://www.shutterstock.com/image-vector/digital-code-background-abstract-vector-illustration-270522938					
123rf	https://tr.123rf site-flyer-desigr	https://tr.123rf.com/photo_52422597_abstract-vector-background-blue-and-violet-waved-lines-for-brochure-web- site-flyer-design-illustratio.html					
Kitap İçinde Kullanıla	Kullanılan Görseller						
Sıra Sizde Görseli	https://www.sh	https://www.shutterstock.com/tr/image-vector/pencil-write-isolated-icon-1200010768					
Uygulama Görseli	https://tr.123rf	https://tr.123rf.com/photo_16516142_netbookdoodle-tarz%C4%B1.html					
Not Görseli	https://tr.123rf	nttps://tr.123rf.com/stok-foto%C4%9Fraf/9043967.html?sti=mlyogs57l6y54x60en &mediapopup=9043967					
Dikkat Görseli	https://tr.123rf	https://tr.123rf.com/profile_johny007pan?mediapopup=13341637					
Sayfa Numarası Görseli	https://tr.123rf.com/klipart-vekt%C3%B6r/computer.html?alttext=1&oriSearch=sign+icon&sti=lvie4ou39uskofb-73d%7C&mediapopup=30644988						
ÖĞRENME BİR	RİMİ-1						
Öğrenme Birimi Kapak Resmi	Shutterstock	https://www.shutterstock.com/tr/image-photo/businessman-working-office-pressing-but-ton-on-598371008					
Görsel 1.1	Shutterstock	https://www.shutterstock.com/tr/image-vector/desktop-computer-laptop-tablet-smart-pho-ne-221566516					
Görsel 1.2	Shutterstock	https://www.shutterstock.com/tr/image-photo/c-sharp-programming-language-web-deve-lopment-604209716					
Görsel 1.3		Yazar tarafından düzenlenmiştir.					
Görsel 1.4		Yazar tarafından düzenlenmiştir.					
Görsel 1.5		Yazar tarafından düzenlenmiştir.					
Görsel 1.6		Yazar tarafından düzenlenmiştir.					
Görsel 1.7		Yazar tarafından düzenlenmiştir.					
Görsel 1.8		Yazar tarafından düzenlenmiştir.					
Görsel 1.9		Yazar tarafından düzenlenmiştir.					
Görsel 1.10		Yazar tarafından düzenlenmiştir.					
Görsel 1.11		Yazar tarafından düzenlenmiştir.					
Görsel 1.12		Yazar tarafından düzenlenmiştir.					
Görsel 1.13		Yazar tarafından düzenlenmiştir.					
Görsel 1.14		Yazar tarafından düzenlenmiştir.					
Görsel 1.15		Yazar tarafından düzenlenmiştir.					
Görsel 1.16		Yazar tarafından düzenlenmiştir.					
Görsel 1.17		Yazar tarafından düzenlenmiştir.					
Görsel 1.18		Yazar tarafından düzenlenmiştir.					
Görsel 1.19		Yazar tarafından düzenlenmiştir.					
Görsel 1.20		Yazar tarafından düzenlenmiştir.					
Görsel 1.21		Yazar tarafından düzenlenmiştir.					
Görsel 1.22		Yazar tarafından düzenlenmiştir.					
Görsel 1.23		Yazar tarafından düzenlenmiştir.					
Görsel 1.24		Yazar tarafından düzenlenmiştir.					
Görsel 1.25		Yazar tarafından düzenlenmiştir.					
Görsel 1.26	Shutterstock	https://www.shutterstock.com/tr/image-photo/c-code-hand-pencil-points-sharp-1099318907					
Görsel 1.27	Shutterstock	https://www.shutterstock.com/tr/image-photo/c-code-hand-pencil-points-sharp- 1099318907					
Görsel 1.28		Yazar tarafından düzenlenmiştir.					
Görsel 1.30		Yazar tarafından düzenlenmiştir.					
Görsel 1.31		Yazar tarafından düzenlenmiştir.					
Görsel 1.32		Yazar tarafından düzenlenmiştir.					



•

•

10. SINIF •

-

-

ÖĞRENME	BİRİMİ-	2
Öğrenme Birimi Kapak Resmi	Shutterstock	https://www.shutterstock.com/tr/image-photo/black-shoes-standing-crossroad-making-decisi- on-1179925327
Görsel 2.1	Shutterstock	https://www.shutterstock.com/tr/image-vector/true-false-green-red-signal-on-307758260
Görsel 2.2		Yazar tarafından düzenlenmiştir.
Görsel 2.3		Yazar tarafından düzenlenmiştir.
Görsel 2.4		Yazar tarafından düzenlenmiştir.
Görsel 2.5		Yazar tarafından düzenlenmiştir.
Görsel 2.6		Yazar tarafından düzenlenmiştir.
Görsel 2.7	Shutterstock	https://www.shutterstock.com/tr/image-vector/man-key-near-computer-account- login-1499141258
Görsel 2.8	Shutterstock	
Görsel 2.9		Yazar tarafından düzenlenmiştir.
Görsel 2.10		Yazar tarafından düzenlenmiştir.
Görsel 2.11		Yazar tarafından düzenlenmiştir.
Görsel 2.12		Yazar tarafından düzenlenmiştir.
Görsel 2.13		Yazar tarafından düzenlenmiştir.
Görsel 2.14		Yazar tarafından düzenlenmiştir.
Görsel 2.15	Shutterstock	https://www.shutterstock.com/tr/image-photo/close-counter-cassette-tape-player-repairing-1781199686
Görsel 2.16		Yazar tarafından düzenlenmiştir.
Görsel 2.17		Yazar tarafından düzenlenmiştir.
Görsel 2.18		Yazar tarafından düzenlenmiştir.
Görsel 2.19		Yazar tarafından düzenlenmiştir.
Görsel 2.20		Yazar tarafından düzenlenmiştir.
Görsel 2.21		Yazar tarafından düzenlenmiştir.
Görsel 2.22		Yazar tarafından düzenlenmiştir.
Görsel 2.23		Yazar tarafından düzenlenmiştir.
Görsel 2.24		Yazar tarafından düzenlenmiştir.
Görsel 2.25		Yazar tarafından düzenlenmiştir.
Görsel 2.26		Yazar tarafından düzenlenmiştir.

• ••

ÖĞRENME BİRİMİ-3						
Öğrenme Birimi Kapak Resmi	Shutterstock	https://www.shutterstock.com/tr/image-vector/object-oriented-programming-oop-273118595				
Görsel 3.1-1	123rf	https://www.123rf.com/stock-photo/55883157.html?sti=nx784oj8xwnh9vplea				
Görsel 3.1-2	123rf	https://www.123rf.com/stock-photo/53600574.html?oriSearch=55883157&sti=od- dw7x1mf2q88i9kfu				
Görsel 3.2		Yazar tarafından düzenlenmiştir.				
Görsel 3.3		Yazar tarafından düzenlenmiştir.				
Görsel 3.4		Yazar tarafından düzenlenmiştir.				
Görsel 3.5		Yazar tarafından düzenlenmiştir.				
Görsel 3.6		Yazar tarafından düzenlenmiştir.				
Görsel 3.7		Yazar tarafından düzenlenmiştir.				
Görsel 3.8		Yazar tarafından düzenlenmiştir.				
Görsel 3.9		Yazar tarafından düzenlenmiştir.				
Görsel 3.10		Yazar tarafından düzenlenmiştir.				



•

/•

_____•

:.

Görsel 3.11	Yazar tarafından düzenlenmiştir.
Görsel 3.12	Yazar tarafından düzenlenmiştir.
Görsel 3.13	Yazar tarafından düzenlenmiştir.
Görsel 3.14	Yazar tarafından düzenlenmiştir.

• ••

:./

ÖĞRENME BİRİMİ-4			
Öğrenme Birimi Kapak Resmi	Shutterstock	https://www.shutterstock.com/image-illustration/neon-light-blockcha in-technology-information-block-1317288464	
Görsel 4.1	Yazar tarafından düzenlenmiştir.	Görsel 4.15	Yazar tarafından düzenlenmiştir.
Görsel 4.2	Yazar tarafından düzenlenmiştir.	Görsel 4.16	Yazar tarafından düzenlenmiştir.
Görsel 4.3	Yazar tarafından düzenlenmiştir.	Görsel 4.17	Yazar tarafından düzenlenmiştir.
Görsel 4.4	Yazar tarafından düzenlenmiştir.	Görsel 4.18	Yazar tarafından düzenlenmiştir.
Görsel 4.5	Yazar tarafından düzenlenmiştir.	Görsel 4.19	Yazar tarafından düzenlenmiştir.
Görsel 4.6	Yazar tarafından düzenlenmiştir.	Görsel 4.20	Yazar tarafından düzenlenmiştir.
Görsel 4.7	Yazar tarafından düzenlenmiştir.	Görsel 4.21	Yazar tarafından düzenlenmiştir.
Görsel 4.8	Yazar tarafından düzenlenmiştir.	Görsel 4.22	Yazar tarafından düzenlenmiştir.
Görsel 4.9	Yazar tarafından düzenlenmiştir.	Görsel 4.23	Yazar tarafından düzenlenmiştir.
Görsel 4.10	Yazar tarafından düzenlenmiştir.	Görsel 4.24	Yazar tarafından düzenlenmiştir.
Görsel 4.11	Yazar tarafından düzenlenmiştir.	Görsel 4.25	Yazar tarafından düzenlenmiştir.
Görsel 4.12	Yazar tarafından düzenlenmiştir.	Görsel 4.26	Yazar tarafından düzenlenmiştir.
Görsel 4.13	Yazar tarafından düzenlenmiştir.	Görsel 4.27	Yazar tarafından düzenlenmiştir.
Görsel 4.14	Yazar tarafından düzenlenmiştir.		

ÖĞRENME BİRİMİ-5				
Öğrenme Birimi Kapak Resmi	Shutterstock	https://www.shutterstock.com/image-vector/software-web-development-programming-con- cept-abstract-1122339353		
Görsel 5.1		Yazar tarafından düzenlenmiştir.	Görsel 5.21	Yazar tarafından düzenlenmiştir.
Görsel 5.2		Yazar tarafından düzenlenmiştir.	Görsel 5.22	Yazar tarafından düzenlenmiştir.
Görsel 5.3		Yazar tarafından düzenlenmiştir.	Görsel 5.23	Yazar tarafından düzenlenmiştir.
Görsel 5.4		Yazar tarafından düzenlenmiştir.	Görsel 5.24	Yazar tarafından düzenlenmiştir.
Görsel 5.5		Yazar tarafından düzenlenmiştir.	Görsel 5.25	Yazar tarafından düzenlenmiştir.
Görsel 5.6		Yazar tarafından düzenlenmiştir.	Görsel 5.26	Yazar tarafından düzenlenmiştir.
Görsel 5.7		Yazar tarafından düzenlenmiştir.	Görsel 5.27	Yazar tarafından düzenlenmiştir.
Görsel 5.8		Yazar tarafından düzenlenmiştir.		
Görsel 5.9		Yazar tarafından düzenlenmiştir.		
Görsel 5.10		Yazar tarafından düzenlenmiştir.		
Görsel 5.11		Yazar tarafından düzenlenmiştir.		
Görsel 5.12		Yazar tarafından düzenlenmiştir.		
Görsel 5.13		Yazar tarafından düzenlenmiştir.		
Görsel 5.14		Yazar tarafından düzenlenmiştir.		
Görsel 5.15		Yazar tarafından düzenlenmiştir.		
Görsel 5.16		Yazar tarafından düzenlenmiştir.		
Görsel 5.17		Yazar tarafından düzenlenmiştir.		
Görsel 5.18		Yazar tarafından düzenlenmiştir.		
Görsel 5.19		Yazar tarafından düzenlenmiştir.		
Görsel 5.20		Yazar tarafından düzenlenmiştir.		



•

/-

_____•

ÖĞRENME B	ÖĞRENME BİRİMİ-6			
Öğrenme Birimi Kapak Resmi	Shutterstock	erstock https://www.shutterstock.com/tr/image-vector/vector-line-web-banner-big-data-795575680		
Görsel 6.1	Shutterstock	https://www.shutterstock.com/tr/image-photo/database-table-technical-concept-girl-poin-ting-92126581		
Görsel 6.2		Yazar tarafından düzenlenmiştir.	Görsel 6.43	Yazar tarafından düzenlenmiştir.
Görsel 6.3		Yazar tarafından düzenlenmiştir.	Görsel 6.44	Yazar tarafından düzenlenmiştir.
Görsel 6.4		Yazar tarafından düzenlenmiştir.	Görsel 6.45	Yazar tarafından düzenlenmiştir.
Görsel 6.5		Yazar tarafından düzenlenmiştir.	Görsel 6.46	Yazar tarafından düzenlenmiştir.
Görsel 6.6		Yazar tarafından düzenlenmiştir.	Görsel 6.47	Yazar tarafından düzenlenmiştir.
Görsel 6.7		Yazar tarafından düzenlenmiştir.	Görsel 6.48	Yazar tarafından düzenlenmiştir.
Görsel 6.8		Yazar tarafından düzenlenmiştir.	Görsel 6.49	Yazar tarafından düzenlenmiştir.
Görsel 6.9		Yazar tarafından düzenlenmiştir.	Görsel 6.50	Yazar tarafından düzenlenmiştir.
Görsel 6.10		Yazar tarafından düzenlenmiştir.	Görsel 6.51	Yazar tarafından düzenlenmiştir.
Görsel 6.11		Yazar tarafından düzenlenmiştir.	Görsel 6.52	Yazar tarafından düzenlenmiştir.
Görsel 6.12		Yazar tarafından düzenlenmiştir.	Görsel 6.53	Yazar tarafından düzenlenmiştir.
Görsel 6.13		Yazar tarafından düzenlenmiştir.	Görsel 6.54	Yazar tarafından düzenlenmiştir.
Görsel 6.14		Yazar tarafından düzenlenmiştir.	Görsel 6.55	Yazar tarafından düzenlenmiştir.
Görsel 6.15		Yazar tarafından düzenlenmiştir.	Görsel 6.56	Yazar tarafından düzenlenmiştir.
Görsel 6.16		Yazar tarafından düzenlenmiştir.	Görsel 6.56-1	Yazar tarafından düzenlenmiştir.
Görsel 6.17		Yazar tarafından düzenlenmiştir.	Görsel 6.57	Yazar tarafından düzenlenmiştir.
Görsel 6.18		Yazar tarafından düzenlenmiştir.	Görsel 6.58	Yazar tarafından düzenlenmiştir.
Görsel 6.19		Yazar tarafından düzenlenmiştir.	Görsel 6.59	Yazar tarafından düzenlenmiştir.
Görsel 6.20		Yazar tarafından düzenlenmiştir.	Görsel 6.60	Yazar tarafından düzenlenmiştir.
Görsel 6.21		Yazar tarafından düzenlenmiştir.	Görsel 6.61	Yazar tarafından düzenlenmiştir.
Görsel 6.22		Yazar tarafından düzenlenmiştir.	Görsel 6.62	Yazar tarafından düzenlenmiştir.
Görsel 6.23		Yazar tarafından düzenlenmiştir.	Görsel 6.63	Yazar tarafından düzenlenmiştir.
Görsel 6.24		Yazar tarafından düzenlenmiştir.	Görsel 6.64	Yazar tarafından düzenlenmiştir.
Görsel 6.25		Yazar tarafından düzenlenmiştir.	Görsel 6.65	Yazar tarafından düzenlenmiştir.
Görsel 6.25-1		Yazar tarafından düzenlenmiştir.	Görsel 6.66	Yazar tarafından düzenlenmiştir.
Görsel 6.26		Yazar tarafından düzenlenmiştir.	Görsel 6.67	Yazar tarafından düzenlenmiştir.
Görsel 6.27		Yazar tarafından düzenlenmiştir.	Görsel 6.68	Yazar tarafından düzenlenmiştir.
Görsel 6.28		Yazar tarafından düzenlenmiştir.	Görsel 6.69	Yazar tarafından düzenlenmiştir.
Görsel 6.29		Yazar tarafından düzenlenmiştir.	Görsel 6.69-1	Yazar tarafından düzenlenmiştir.
Görsel 6.30		Yazar tarafından düzenlenmiştir.	Görsel 6.69-2	Yazar tarafından düzenlenmiştir.
Görsel 6.31		Yazar tarafından düzenlenmiştir.	Görsel 6.69-3	Yazar tarafından düzenlenmiştir.
Görsel 6.32		Yazar tarafından düzenlenmiştir.	Görsel 6.70	Yazar tarafından düzenlenmiştir.
Görsel 6.33		Yazar tarafından düzenlenmiştir	Görsel 6.70-1	Yazar tarafından düzenlenmiştir
Görsel 6 34		Yazar tarafından düzenlenmiştir	Görsel 6 70-2	Yazar tarafından düzenlenmiştir
Görsel 6.35		Yazar tarafından düzenlenmiştir.	Görsel 6.771	Yazar tarafından düzenlenmiştir
Görsel 6.36		Yazar tarafından düzenlenmiştir	Görsel 6.72	Yazar tarafından düzenlenmiştir
Görsel 6.37		Yazar tarafından düzenlenmiştir.	Görsel 6.73	Yazar tarafından düzenlenmiştir.
Görsel 6.38		Yazar tarafından düzenlenmistir.	Görsel 6.74	Yazar tarafından düzenlenmiştir.
Görsel 6.39		Yazar tarafından düzenlenmiştir.	Görsel 6.75	Yazar tarafından düzenlenmiştir.
Görsel 6.40		Yazar tarafından düzenlenmiştir.	Görsel 6.76	Yazar tarafından düzenlenmiştir.
Görsel 6.41		Yazar tarafından düzenlenmiştir.	Görsel 6.77	Yazar tarafından düzenlenmiştir.
Görsel 6.42		Yazar tarafından düzenlenmiştir.	Görsel 6.78	Yazar tarafından düzenlenmiştir.

• ••

:/

•



•

_

	[
Görsel 6.79	Yazar tarafından düzenlenmiştir.	Görsel 6.119	Yazar tarafından düzenlenmiştir.
Görsel 6.80	Yazar tarafından düzenlenmiştir.	Görsel 6.120	Yazar tarafından düzenlenmiştir.
Görsel 6.81	Yazar tarafından düzenlenmiştir.	Görsel 6.121	Yazar tarafından düzenlenmiştir.
Görsel 6.82	Yazar tarafından düzenlenmiştir.	Görsel 6.121-1	Yazar tarafından düzenlenmiştir.
Görsel 6.83	Yazar tarafından düzenlenmiştir.	Görsel 6.122	Yazar tarafından düzenlenmiştir.
Görsel 6.84	Yazar tarafından düzenlenmiştir.	Görsel 6.123	Yazar tarafından düzenlenmiştir.
Görsel 6.85	Yazar tarafından düzenlenmiştir.	Görsel 6.124	Yazar tarafından düzenlenmiştir.
Görsel 6.86	Yazar tarafından düzenlenmiştir.	Görsel 6.125	Yazar tarafından düzenlenmiştir.
Görsel 6.87	Yazar tarafından düzenlenmiştir.	Görsel 6.126	Yazar tarafından düzenlenmiştir.
Görsel 6.88	Yazar tarafından düzenlenmiştir.	Görsel 6.127	Yazar tarafından düzenlenmiştir.
Görsel 6.89	Yazar tarafından düzenlenmiştir.	Görsel 6.128	Yazar tarafından düzenlenmiştir.
Görsel 6.90	Yazar tarafından düzenlenmiştir.	Görsel 6.128-1	Yazar tarafından düzenlenmiştir.
Görsel 6.91	Yazar tarafından düzenlenmiştir.	Görsel 6.128-2	Yazar tarafından düzenlenmiştir.
Görsel 6.92	Yazar tarafından düzenlenmiştir.	Görsel 6.129	Yazar tarafından düzenlenmiştir.
Görsel 6.93	Yazar tarafından düzenlenmiştir.	Görsel 6.129-1	Yazar tarafından düzenlenmiştir.
Görsel 6.94	Yazar tarafından düzenlenmiştir.	Görsel 6.129-2	Yazar tarafından düzenlenmiştir.
Görsel 6.95	Yazar tarafından düzenlenmiştir.	Görsel 6.130	Yazar tarafından düzenlenmiştir.
Görsel 6.96	Yazar tarafından düzenlenmiştir.	Görsel 6.131	Yazar tarafından düzenlenmiştir.
Görsel 6.97	Yazar tarafından düzenlenmiştir.	Görsel 6.132	Yazar tarafından düzenlenmiştir.
Görsel 6.98	Yazar tarafından düzenlenmiştir.	Görsel 6.133	https://oguzcangenc.com.tr/blog/linq-langu-
			age-integrated-query-nedir/
Görsel 6.99	Yazar tarafından düzenlenmiştir.	Görsel 6.134	Yazar tarafından düzenlenmiştir.
Görsel 6.100	Yazar tarafından düzenlenmiştir.	Görsel 6.135	Yazar tarafından düzenlenmiştir.
Görsel 6.101	Yazar tarafından düzenlenmiştir.	Görsel 6.136	Yazar tarafından düzenlenmiştir.
Görsel 6.102	Yazar tarafından düzenlenmiştir.	Görsel 6.136-1	Yazar tarafından düzenlenmiştir.
Görsel 6.103	Yazar tarafından düzenlenmiştir.	Görsel 6.137	Yazar tarafından düzenlenmiştir.
Görsel 6.103-1	Yazar tarafından düzenlenmiştir.	Görsel 6.138	Yazar tarafından düzenlenmiştir.
Görsel 6.104	Yazar tarafından düzenlenmiştir.	Görsel 6.139	Yazar tarafından düzenlenmiştir.
Görsel 6.105	Yazar tarafından düzenlenmiştir.	Görsel 6.140	Yazar tarafından düzenlenmiştir.
Görsel 6.106	Yazar tarafından düzenlenmiştir.	Görsel 6.141	Yazar tarafından düzenlenmiştir.
Görsel 6.107	Yazar tarafından düzenlenmiştir.	Görsel 6.142	Yazar tarafından düzenlenmiştir.
Görsel 6.108	Yazar tarafından düzenlenmiştir.	Görsel 6.143	Yazar tarafından düzenlenmiştir.
Görsel 6.109	Yazar tarafından düzenlenmiştir.	Görsel 6.144	Yazar tarafından düzenlenmiştir.
Görsel 6.110	Yazar tarafından düzenlenmiştir.	Görsel 6.145	Yazar tarafından düzenlenmiştir.
Görsel 6.111	Yazar tarafından düzenlenmiştir.	Görsel 6.146	Yazar tarafından düzenlenmiştir.
Görsel 6.112	Yazar tarafından düzenlenmiştir.	Görsel 6.147	Yazar tarafından düzenlenmiştir.
Görsel 6.113	Yazar tarafından düzenlenmiştir.	Görsel 6.148	Yazar tarafından düzenlenmiştir.
Görsel 6.114	Yazar tarafından düzenlenmiştir.	Görsel 6.149	Yazar tarafından düzenlenmiştir.
Görsel 6.115	Yazar tarafından düzenlenmiştir.	Görsel 6.150	Yazar tarafından düzenlenmiştir.
Görsel 6.116	Yazar tarafından düzenlenmiştir.	Görsel 6.151	Yazar tarafından düzenlenmiştir.
Görsel 6.117	Yazar tarafından düzenlenmiştir.	Görsel 6.152	Yazar tarafından düzenlenmiştir.
Görsel 6.118	Yazar tarafından düzenlenmiştir.	Görsel 6.153	Yazar tarafından düzenlenmiştir.
Görsel 6.118-1	Yazar tarafından düzenlenmiştir.	Görsel 6.154	Yazar tarafından düzenlenmiştir.

•

305

• ••

10. sinif •



KAYNAKÇA

- Bilişim Teknolojileri Alanı Çerçeve Öğretim Programı
- Türk Dil Kurumu Türkçe Sözlük
- Türk Dil Kurumu Yazım Kılavuzu
- C# Belgeleri (2020). Erişim adresi: https://docs.microsoft.com/tr-tr/dotnet/csharp/
- Sharp, J. (2020). Microsoft Visual C# Step by Step. U.S.: Microsoft Press
- Aktaş, V. (2020). Her Yönüyle C# 8.0. İstanbul: Kodlab
- https://medium.com/@nuriyavuz2.71/sql-101-ve-sql-injection-101-d2b35dff52de
- https://www.oracle.com/tr/database/what-is-database.html
- https://www.ozcanbayri.com.tr/big-data-nedir/
- https://kod5.org/t-sql-veri-tipleri/
- https://ceaksan.com/tr/primary-unique-foreign-key
- https://www.dijitalders.com/icerik/2378/veritabani_nedir.html
- https://icon-icons.com/pack/Modern-Education-And-Knowledge-Power/1804 -
- https://icon-icons.com/icon/education-school-lunch-box-break-bag-food/133449
- https://icon-icons.com/pack/Modern-Education-And-Knowledge-Power/1804
- https://medium.com/@kdrcandogan/entity-framework-nedir-6aed4dcf6328
- https://beylikweb.com/veri-madenciligi-nedir-tanimi-amaci-ve-teknikleri/
- http://www.sercangulgeze.com.tr/ado.net-nedir-nasil-kullanilir.html
- https://oguzcangenc.com.tr/blog/linq-language-integrated-query-nedir/



