

10. Sınıf

Nesne Tabanlı Programlama

36 Haftalık Eğitim Kitabı

Mesleki ve Teknik Anadolu Lisesi - Bilişim Teknolojileri Alanı

Konu anlatımı • çözümlü C# örnekleri • günlük yaşam uygulamaları • öğrenci çalışma rehberi

İçindekiler

1. Hafta	Geliştirme ortamı ve ilk C# projesi
2. Hafta	Değişkenler, veri türleri ve operatörler
3. Hafta	Karar yapıları
4. Hafta	Döngüler
5. Hafta	Metotlar ve parametreler
6. Hafta	Diziler ve koleksiyonlara giriş
7. Hafta	Sınıf ve nesne kavramı
8. Hafta	Alanlar, özellikler ve kurucu metotlar
9. Hafta	Kapsülleme
10. Hafta	Uygulama: Akıllı kumbara
11. Hafta	Değer ve referans türleri
12. Hafta	Metot aşırı yükleme
13. Hafta	Statik üyeler
14. Hafta	Kalıtım
15. Hafta	Metot ezme ve virtual/override
16. Hafta	Polimorfizm
17. Hafta	Uygulama: Kütüphane sistemi
18. Hafta	Birinci dönem uygulama değerlendirme
19. Hafta	Soyut sınıflar
20. Hafta	Arayüzler
21. Hafta	Enum ve struct
22. Hafta	Hata yönetimi
23. Hafta	Generic koleksiyonlar
24. Hafta	LINQ temelleri
25. Hafta	Dosya işlemleri
26. Hafta	JSON veri saklama
27. Hafta	Windows Forms proje yapısı
28. Hafta	Form kontrolleri ve olaylar
29. Hafta	Çok formlu uygulamalar
30. Hafta	Katmanlı mimariye giriş

31. Hafta	Veritabanı ve CRUD
32. Hafta	Uygulama: Toplu taşıma kartı
33. Hafta	Test, hata ayıklama ve doğrulama
34. Hafta	Dönem projesi tasarımı
35. Hafta	Dönem projesi geliştirme
36. Hafta	Sunum, değerlendirme ve portfolyo

Geliştirme ortamı ve ilk C# projesi

Odak: IDE, .NET SDK, proje ve Program.cs

Kazanımlar

1	IDE, .NET SDK, proje ve Program.cs kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Günlük su tüketimini hesaplayan ilk konsol uygulamasını oluşturma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
const int hedef = 8; int icilen = 5; Console.WriteLine(hedef - icilen);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Günlük su tüketimini hesaplayan ilk konsol uygulamasını oluşturma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Değişkenler, veri türleri ve operatörler

Odak: Değişken, veri türü ve operatör seçimi

Kazanımlar

1	Değişken, veri türü ve operatör seçimi kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Bir manav alışverişinin ürün adını, miktarını ve tutarını saklama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
string urun = "Elma"; double kilo = 2.5; decimal fiyat = 38.50m; decimal toplam = (decimal)kilo * fiyat;
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Bir manav alışverişinin ürün adını, miktarını ve tutarını saklama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Karar yapıları

Odak: if, else if, else ve karşılaştırma operatörleri

Kazanımlar

1	if, else if, else ve karşılaştırma operatörleri kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Hava sıcaklığına göre kıyafet önerme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
int sıcaklik = 12; string oneri = sıcaklik < 10 ? "Mont" : sıcaklik < 20 ? "Ceket" : "Tişört";
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Hava sıcaklığına göre kıyafet önerme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Döngüler

Odak: for, while, sayaç ve toplam değişkeni

Kazanımlar

1	for, while, sayaç ve toplam değişkeni kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Bir haftanın günlük adım sayılarını toplama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
int[] adimlar = {6500,8200,7100,9000,5400,10300,7800}; int toplam = 0; foreach (int adim in adimlar) toplam
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Bir haftanın günlük adım sayılarını toplama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Metotlar ve parametreler

Odak: Metot tanımlama, parametre ve dönüş değeri

Kazanımlar

1	Metot tanımlama, parametre ve dönüş değeri kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Kargo ücretini ağırlığa göre hesaplama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
decimal KargoUcreti(decimal kg) => 30 + kg * 8; Console.WriteLine(KargoUcreti(4.5m));
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Kargo ücretini ağırlığa göre hesaplama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Diziler ve koleksiyonlara giriş

Odak: Dizi, List ve elemanlara erişim

Kazanımlar

1	Dizi, List ve elemanlara erişim kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Sınıf yoklama listesini saklama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
List<string> ogrenciler = ["Ayşe", "Mert", "Ece"]; ogrenciler.Add("Can");
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Sınıf yoklama listesini saklama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Sınıf ve nesne kavramı

Odak: Sınıf, nesne, özellik ve metot

Kazanımlar

1	Sınıf, nesne, özellik ve metot kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Öğrenci bilgilerini ve not ortalamasını nesne olarak modelleme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
var ogrenci = new Ogrenci("Zeynep", 10); ogrenci.NotEkle(85);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Öğrenci bilgilerini ve not ortalamasını nesne olarak modelleme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Alanlar, özellikler ve kurucu metotlar

Odak: Alan, özellik ve kurucu metot

Kazanımlar

1	Alan, özellik ve kurucu metot kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Bir ürün nesnesini başlangıç değerleriyle oluşturma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
var urun = new Urun("Deftir", 45m); record Urun(string Ad, decimal Fiyat);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Bir ürün nesnesini başlangıç değerleriyle oluşturma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Kapsülleme

Odak: private alan, doğrulama ve kontrollü erişim

Kazanımlar

1	private alan, doğrulama ve kontrollü erişim kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Banka hesabında negatif bakiyeyi engelleme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
class Hesap { public decimal Bakiye { get; private set; } public void ParaYatir(decimal t){ if(t>0) Bakiye+=
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Banka hesabında negatif bakiyeyi engelleme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Uygulama: Akıllı kumbara

Odak: Sınıf tasarımı ve durum değıştiren metotlar

Kazanımlar

1	Sınıf tasarımı ve durum değıştiren metotlar kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Akıllı kumbaraya para ekleme ve çekme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
var kumbara = new Kumbara(); kumbara.ParaEkle(100);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Akıllı kumbaraya para ekleme ve çekme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Değer ve referans türleri

Odak: Değer türü kopyası ve referans paylaşımı

Kazanımlar

1	Değer türü kopyası ve referans paylaşımı kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Bir sayının kopyasıyla listenin referansını karşılaştırma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
int a=5; int b=a; b=9; List<int> x=[1]; List<int> y=x; y.Add(2);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Bir sayının kopyasıyla listenin referansını karşılaştırma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Metot aşırı yükleme

Odak: Aynı metot adında farklı parametreler

Kazanımlar

1	Aynı metot adında farklı parametreler kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Alan hesabını kare ve dikdörtgen için hesaplama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
double Alan(double kenar)=>kenar*kenar; double Alan(double a,double b)=>a*b;
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Alan hesabını kare ve dikdörtgen için hesaplama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Statik üyeler

Odak: static alan ve static metot

Kazanımlar

1	static alan ve static metot kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Oluşturulan öğrenci sayısını sınıf düzeyinde tutma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
class Sayac { public static int Adet; public Sayac(){ Adet++; } }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Oluşturulan öğrenci sayısını sınıf düzeyinde tutma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Kalıtım

Odak: Temel sınıf, türetilmiş sınıf ve yeniden kullanım

Kazanımlar

1	Temel sınıf, türetilmiş sınıf ve yeniden kullanım kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Çalışan sınıfından öğretmen sınıfı türetme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
class Calisan { public string Ad=""; } class Ogretmen : Calisan { public string Brans=""; }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Çalışan sınıfından öğretmen sınıfı türetme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Metot ezme ve virtual/override

Odak: virtual, override ve davranışı özelleştirme

Kazanımlar

1	virtual, override ve davranışı özelleştirme kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Farklı bildirim türlerinin mesaj göndermesi. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
class Bildirim { public virtual void Gonder(){} } class Sms:Bildirim { public override void Gonder()=>Consol
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Farklı bildirim türlerinin mesaj göndermesi için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Polimorfizm

Odak: Ortak tür üzerinden farklı davranışlar

Kazanımlar

1	Ortak tür üzerinden farklı davranışlar kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Ödeme yöntemlerini tek listeden çalıştırma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
List<Odeme> odemeler=[new Kart(),new Nakit()]; foreach(var o in odemeler) o.Ode();
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Ödeme yöntemlerini tek listeden çalıştırma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Uygulama: Kütüphane sistemi

Odak: Nesneler arası ilişki ve ödünç durumu

Kazanımlar

1	Nesneler arası ilişki ve ödünç durumu kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Kitabın ödünç alınmasını ve iadesini yönetme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
var kitap = new Kitap("Nce Memed"); kitap.OduncAl();
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Kitabın ödünç alınmasını ve iadesini yönetme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Birinci dönem uygulama değerlendirmesi

Odak: Dönem kazanımlarını birlikte kullanma

Kazanımlar

1	Dönem kazanımlarını birlikte kullanma kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Küçük bir okul kantini otomasyonu tasarlama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// Menü, sipariş, toplam, bütçe kontrolü ve raporlama adımlarını ayrı metotlara bölün.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Küçük bir okul kantini otomasyonu tasarlama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Soyut sınıflar

Odak: abstract sınıf ve zorunlu davranış

Kazanımlar

1	abstract sınıf ve zorunlu davranış kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Farklı araçların yakıt maliyetini hesaplama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
abstract class Arac { public abstract decimal Maliyet(int km); }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Farklı araçların yakıt maliyetini hesaplama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Arayüzler

Odak: interface ile sözleşme tanımlama

Kazanımlar

1	interface ile sözleşme tanımlama kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Yazdırılabilir rapor türleri oluşturma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
interface IYazdirilabilir { void Yazdir(); } class Fatura:IYazdirilabilir { public void Yazdir(){} }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Yazdırılabilir rapor türleri oluşturma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kardondar.com üzerindeki ilgili derste bulunur.

Enum ve struct

Odak: enum seçenekleri ve struct veri grubu

Kazanımlar

1	enum seçenekleri ve struct veri grubu kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Sipariş durumu ve koordinat bilgisi saklama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
enum Durum { Hazirlaniyor, Yolda, Teslim } record struct Konum(double X, double Y);
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Sipariş durumu ve koordinat bilgisi saklama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Hata yönetimi

Odak: try, catch, finally ve güvenli dönüşüm

Kazanımlar

1	try, catch, finally ve güvenli dönüşüm kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Kullanıcının girdiği yaşı güvenli biçimde okuma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
try { int yas=int.Parse("16"); } catch(FormatException) { Console.WriteLine("Geçersiz sayı"); }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Kullanıcının girdiği yaşı güvenli biçimde okuma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Generic koleksiyonlar

Odak: List, Dictionary, Queue ve Stack

Kazanımlar

1	List, Dictionary, Queue ve Stack kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Telefon rehberini anahtar-değer olarak saklama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
Dictionary<string,string> rehber=new(); rehber["Ali"]="555 0101";
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Telefon rehberini anahtar-değer olarak saklama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

LINQ temelleri

Odak: Where, Select, OrderBy ve toplama işlemleri

Kazanımlar

1	Where, Select, OrderBy ve toplama işlemleri kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Ürünleri fiyata göre filtreleme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
var pahali = fiyatlar.Where(f=>f>100).OrderBy(f=>f).ToList();
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Ürünleri fiyata göre filtreleme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Dosya işlemleri

Odak: Dosya yazma, okuma ve using

Kazanımlar

1	Dosya yazma, okuma ve using kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Günlük çalışma notlarını metin dosyasına kaydetme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
File.WriteAllText("gunluk.txt", "Bugün C# çalıştım"); string metin=File.ReadAllText("gunluk.txt");
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Günlük çalışma notlarını metin dosyasına kaydetme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

JSON veri saklama

Odak: JSON serileştirme ve geri okuma

Kazanımlar

1	JSON serileştirme ve geri okuma kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Ayarları JSON dosyasında saklama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
string json=JsonSerializer.Serialize(new { Tema="Koyu", Dil="tr" });
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Ayarları JSON dosyasında saklama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Windows Forms proje yapısı

Odak: Form, Program.cs ve olay tabanlı yapı

Kazanımlar

1	Form, Program.cs ve olay tabanlı yapı kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Basit not hesaplama formu oluşturma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// TextBox, Button ve Label ekleyin; hesaplamay■ Button Click olay■nda yap■n.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Basit not hesaplama formu oluşturma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Form kontrolleri ve olaylar

Odak: Button, TextBox, Label, ComboBox ve olaylar

Kazanımlar

1	Button, TextBox, Label, ComboBox ve olaylar kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Döviz seçimine göre tutar hesaplayan form. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
// ComboBox seçimi ve TextBox tutarın Click olayında okuyun.
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Döviz seçimine göre tutar hesaplayan form için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Çok formulu uygulamalar

Odak: Formlar arası veri taşıma ve pencere yönetimi

Kazanımlar

1	Formlar arası veri taşıma ve pencere yönetimi kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Giriş ekranından ana menüye kullanıcı adı aktarma. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// AnaForm kurucusuna kullanıcının adı adn parametre olarak gönderin.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Giriş ekranından ana menüye kullanıcı adı aktarma için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Katmanlı mimariye giriş

Odak: Sunum, iş ve veri katmanlarını ayırma

Kazanımlar

1	Sunum, iş ve veri katmanlarını ayırma kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Öğrenci kayıt uygulamasını katmanlara bölme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
// UI -> OğrenciService -> OğrenciRepository bağımlılık yönünü kurun.
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Öğrenci kayıt uygulamasını katmanlara bölme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Veritabanı ve CRUD

Odak: Bağlantı, sorgu, ekleme, listeleme, güncelleme ve silme

Kazanımlar

1	Bağlantı, sorgu, ekleme, listeleme, güncelleme ve silme kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Öğrenci kayıtlarını veritabanında yönetme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// Parametrelili INSERT/SELECT/UPDATE/DELETE sorguları kullanın.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Öğrenci kayıtlarını veritabanında yönetme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Uygulama: Toplu taşıma kartı

Odak: Kalıtım, polimorfizm ve bakiye kontrolü

Kazanımlar

1	Kalıtım, polimorfizm ve bakiye kontrolü kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Öğrenci ve tam ulaşım kartlarını modelleme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
abstract class Kart { public decimal Bakiye; public abstract decimal Ucret {get;} }
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Öğrenci ve tam ulaşım kartlarını modelleme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Test, hata ayıklama ve doğrulama

Odak: Breakpoint, adım adım çalışma ve doğrulama

Kazanımlar

1	Breakpoint, adım adım çalışma ve doğrulama kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Hatalı indirim hesabını test ederek düzeltme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
decimal Indirimli(decimal fiyat,int oran){ if(oran<0||oran>100) throw new ArgumentException(); return fiyat*
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Hatalı indirim hesabını test ederek düzeltme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirondar.com üzerindeki ilgili derste bulunur.

Dönem projesi tasarımı

Odak: İhtiyaç analizi, kullanıcı hikâyesi ve sınıf diyagramı

Kazanımlar

1	İhtiyaç analizi, kullanıcı hikâyesi ve sınıf diyagramı kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Dönem projesinin kapsamını ve ekranlarını planlama. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

```
// Kullanıcı, problem, özellikler, veri modeli ve kabul ölçütlerini yazın.
```

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Dönem projesinin kapsamını ve ekranlarını planlama için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Dönem projesi geliştirme

Odak: Parçalara ayırma, sürümleme ve düzenli test

Kazanımlar

1	Parçalara ayırma, sürümleme ve düzenli test kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Planlanan dönem projesini çalışan parçalara dönüştürme. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// Önce temel ak■■■ bitirin; her özellikten sonra derleyip test edin.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Planlanan dönem projesini çalışan parçalara dönüştürme için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.

Sunum, değerlendirme ve portfolyo

Odak: Sunum, demo, öz değerlendirme ve portfolyo

Kazanımlar

1	Sunum, demo, öz değerlendirme ve portfolyo kavramlarını açıklayabilme
2	Problemi girdi, işlem ve çıktı adımlarına ayırabilme
3	C# ile okunabilir bir çözüm geliştirip test edebilme

Günlük yaşam problemi

Dönem projesini kanıtlarıyla tanıtmak. Önce gerekli verileri, yapılacak işlemleri ve beklenen çıktıyı belirleyin.

Çözümlü örnek

// Problem, çözüm, canlı demo, karşılaşılan hata ve geliştirme önerisini sunun.

Çözüm yolu

1. Problemi küçük parçalara ayırın.
2. Uygun veri türü ve programlama yapısını seçin.
3. Kodunuzu anlamlı adlarla yazın.
4. Normal, sınır ve hatalı değerlerle test edin.

Üç seviyeli çalışma

Kolay	Dönem projesini kanıtlarıyla tanıtmak için sabit örnek değerlerle çalışan çözüm hazırlayın.
Orta	Çözümü farklı girdileri doğrulayacak ve hatalı durumda açıklayıcı mesaj verecek şekilde geliştirin.
İleri	Birden fazla kayıtlı çalışan, özet sonuç ve rapor üreten küçük bir uygulamaya dönüştürün.

Not: Etkileşimli teslim, ipucu ve çözüm ekranları kadirdondar.com üzerindeki ilgili derste bulunur.