

3. Hafta Ödev Çözümleri

Ödevler-1 Çözümleri

Ödev 1: Liste Ortalaması Hesaplama

```
public double CalculateAverage(List<int> numberList)
{
    if (numberList == null || numberList.Count == 0)
    {
        return 0;
    }

    double toplam = 0;
    foreach (int sayi in numberList)
    {
        toplam += sayi;
    }

    return toplam / numberList.Count;
}
```

Ödev 2: Sesli Harf Sayacı

```
public int CountVowels(string text)
{
    if (string.IsNullOrEmpty(text))
    {
        return 0;
    }

    int sesliHarfSayisi = 0;
    string turkceSesliHarfler = "aeıioöü";
    string kucukHarfMetin = text.ToLower();
```

```
foreach (char karakter in kucukHarfMetin)
{
    if (turkceSesliHarfler.Contains(karakter))
    {
        sesliHarfSayisi++;
    }
}
return sesliHarfSayisi;
}
```

Ödev 3: Üs Alma (Döngü ile)

```
public int CalculatePower(int baseNumber, int exponent)
{
    if (exponent < 0) return 0; // Negatif üsler bu metot için kapsam dışı.
    if (exponent == 0) return 1;

    int sonuc = 1;
    for (int i = 0; i < exponent; i++)
    {
        sonuc *= baseNumber;
    }
    return sonuc;
}
```

Ödev 4: Asal Sayı Avcısı

```
public bool IsPrime(int number)
{
    if (number <= 1) return false;
    if (number == 2) return true;
    if (number % 2 == 0) return false;
```

```
// Sayının kareköküne kadar olan tek sayıları kontrol etmek
yeterlidir.
for (int i = 3; i * i <= number; i += 2)
{
    if (number % i == 0)
    {
        return false;
    }
}
return true;
}
```

Ödev 5: Palindromik Kelimeler

```
public bool IsPalindrome(string text)
{
    string küçükHarfMetin = text.ToLower();
    string tersMetin = "";
    for (int i = küçükHarfMetin.Length - 1; i >= 0; i--)
    {
        tersMetin += küçükHarfMetin[i];
    }
    return küçükHarfMetin.Equals(tersMetin);
}
```

Ödev 6: Mükemmel Sayı Bulucu

```
public bool IsPerfectNumber(int number)
{
    if (number <= 1)
    {
        return false;
    }
    int bolenlerToplami = 1; // 1 her zaman bir bölendir.
    // Bir sayının bölenleri yarısından büyük olamaz (kendisi hariç).
    // Döngüyü sayının kareköküne kadar optimize edebiliriz.
```

```

for (int i = 2; i * i <= number; i++)
{
    if (number % i == 0)
    {
        bolenlerToplami += i;
        if (i * i != number) // i'nin karesi sayı değilse, diğer
bölteni de ekle.
        {
            bolenlerToplami += number / i;
        }
    }
}
return bolenlerToplami == number;
}

```

Ödev 7: Listedeki En Büyük İkinci Sayı

```

public int FindSecondLargest(List<int> numberList)
{
    // Listenin en az iki eleman içerdiğini varsayıyoruz.
    int enBuyuk = int.MinValue;
    int ikinciEnBuyuk = int.MinValue;

    foreach (int sayi in numberList)
    {
        if (sayi > enBuyuk)
        {
            ikinciEnBuyuk = enBuyuk;
            enBuyuk = sayi;
        }
        else if (sayi > ikinciEnBuyuk && sayi != enBuyuk)
        {
            ikinciEnBuyuk = sayi;
        }
    }
    return ikinciEnBuyuk;
}

```

Ödev 8: Metin Şifreleme (Sezar Şifrelemesi)

```
public string Encrypt(string text, int shiftAmount)
{
    char[] buffer = text.ToCharArray();
    for (int i = 0; i < buffer.Length; i++)
    {
        char harf = buffer[i];
        if (char.IsLetter(harf))
        {
            char baslangic = char.IsUpper(harf) ? 'A' : 'a';
            // Harfi kaydır ve alfabe dışına çıkarsa başa dönmesini sağla
            (mod 26).
            buffer[i] = (char)(baslangic + (harf - baslangic +
shiftAmount) % 26);
        }
    }
    return new string(buffer);
}
```

Ödev 9: Basamakları Toplama

```
public int SumDigits(int number)
{
    int toplam = 0;
    // Sayı negatifse pozitive çevirerek işlem yapalım.
    number = Math.Abs(number);

    while (number > 0)
    {
        toplam += number % 10; // Son basamağı al ve toplama ekle
        number /= 10;          // Son basamağı at
    }
    return toplam;
}
```

Ödev 10: Fibonacci Dizisi Oluşturucu

```
public List<int> GenerateFibonacci(int count)
{
    List<int> fibDizisi = new List<int>();
    if (count <= 0)
    {
        return fibDizisi;
    }

    if (count >= 1) fibDizisi.Add(0);
    if (count >= 2) fibDizisi.Add(1);

    for (int i = 2; i < count; i++)
    {
        int sonrakiSayi = fibDizisi[i - 1] + fibDizisi[i - 2];
        fibDizisi.Add(sonrakiSayi);
    }
    return fibDizisi;
}
```

Ödevler-2 Çözümleri

Bu bölüm, “Refactoring” ve “Hata Avı” görevlerinin çözümlerini içerir.

Ödev-1: Refactoring (Yeniden Düzenleme) Çözümü

Aşağıdaki kod, istenen görevleri yerine getirmek için `KullaniciBilgileriniAl`, `YasHesapla` ve `SonucuYazdir` olmak üzere üç ayrı metoda bölünmüştür. `KullaniciBilgileriniAl` metodu, birden fazla değeri döndürmek için `tuple` yerine `out` parametrelerini kullanmaktadır.

```
using System;

public class Program
{
    public static void Main(string[] args)
    {
        // 1. Kullanıcıdan bilgileri alan ve bir string dizisi olarak
        döndüren metodu çağır.
        string[] kullaniciBilgileri = KullaniciBilgileriniAl();

        // 2. Diziden verileri geri al. Unutma: Sayıları metinden tam
        sayıya geri çevirmeliyiz.
        string ad = kullaniciBilgileri[0];
        string soyad = kullaniciBilgileri[1];
        int dogumYili = Convert.ToInt32(kullaniciBilgileri[2]);

        // 3. Alınan bilgilerle yaşı hesapla.
        int yas = YasHesapla(dogumYili);

        // 4. Elde edilen sonuçları ekrana yazdır.
        SonucuYazdir(ad, soyad, yas);
    }

    /// <summary>
    /// Kullanıcıdan bilgileri alır ve bu bilgileri içeren 3 elemanlı bir
    string dizisi döndürür.
    /// Dizi Yapısı: [0] = Ad, [1] = Soyad, [2] = Doğum Yılı (metin
    olarak)
    /// </summary>
    /// <returns>Kullanıcı bilgilerini içeren bir string
    dizisi.</returns>
    public static string[] KullaniciBilgileriniAl()
    {
        Console.WriteLine("Hoş geldiniz!");
        Console.WriteLine("-----");

        string[] bilgiler = new string[3];

        Console.WriteLine("Lütfen adınızı girin:");
        bilgiler[0] = Console.ReadLine();

        Console.WriteLine("Lütfen soyadınızı girin:");
        bilgiler[1] = Console.ReadLine();
    }
}
```

```
        Console.WriteLine("Lütfen doğum yılınızı girin:");
        bilgiler[2] = Console.ReadLine(); // Doğum yılını da metin olarak
        alıp diziye koyuyoruz.
```

```
        return bilgiler;
    }
}
```

```
/// <summary>
/// Verilen doğum yılına göre yaşı hesaplar.
/// </summary>
```

```
public static int YasHesapla(int dogumYili)
{
    return 2025 - dogumYili;
}
```

```
/// <summary>
/// Kullanıcının adını, soyadını ve yaşını alarak sonuçları ekrana
yazdırır.
```

```
/// </summary>
```

```
public static void SonucuYazdir(string ad, string soyad, int yas)
{
    Console.WriteLine("-----");
    Console.WriteLine("Bilgileriniz işleniyor...");

    string tamAd = ad + " " + soyad;
    Console.WriteLine($"Merhaba {tamAd}! Siz {yas} yaşındasınız.");
}
```

```
if (yas < 18)
{
    Console.WriteLine("Sistemi kullanmak için reşit
değilsiniz.");
}
else
{
    Console.WriteLine("Sisteme girişiniz onaylandı.");
}
}
}
```


Ödev-2: Hata Avı (Bug Hunt) Çözümü

Verilen koddaki mantıksal hata, pozitif sayıları toplamak yerine üzerine yazmasıydı. Hata, = operatörünün += ile değiştirilmesiyle düzeltilmiş ve yorum satırında açıklanmıştır.

```
using System;
```

```
public class Program
```

```
{
```

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

```
    {
```

```
        int[] sayilar = { 10, -5, 20, 8, -15, 30 };
```

```
        int pozitifToplam = 0;
```

```
        for (int i = 0; i < sayilar.Length; i++)
```

```
        {
```

```
            // Sadece pozitif sayıları işleme al
```

```
            if (sayilar[i] > 0)
```

```
            {
```

```
                /*
```

```
                * HATA AÇIKLAMASI:
```

```
                * Hata, bu satırda bulunmaktadır. Orijinal kod, pozitif  
bir sayı bulduğunda
```

```
                * bu sayıyı 'pozitifToplam' değişkenine eklemek yerine,  
doğrudan atama yapıyordu (`=`).
```

```
                * Bu yüzden döngünün her adımında 'pozitifToplam'ın  
değeri, o anki pozitif
```

```
                * sayıyla güncelleniyor ve önceki değerler kayboluyordu.  
Döngü tamamlandığında
```

```
                * 'pozitifToplam' değişkeni, dizideki son pozitif  
sayının (30) değerini tutuyordu.
```

```
                *
```

```
                * DÜZELTME:
```

```
                * Değişkenin değerini atamak yerine, mevcut pozitif  
sayıyı toplama eklemek için
```

```
                * atama operatörü `=` yerine toplama ve atama operatörü  
`+=` kullanılmıştır.
```

```
                */
```

```
                pozitifToplam += sayilar[i]; // Hata düzeltildi: = yerine  
+= kullanıldı.
```

```
        }  
    }  
  
    // Beklenen: 10 + 20 + 8 + 30 = 68  
    // Düzeltilmiş kod ile Çıkan: 68  
    Console.WriteLine("Pozitif sayıların toplamı: " + pozitifToplam);  
}  
}
```