

ÖDEVLER-1

AÇIKLAMA:

Bir console projesi oluşturun. Program.cs içindeki Main metodunun dışında(tıpkı bizim derste yaptığımız gibi) aşağıdaki metotları yazın.

Not: Yazdığınız Main metodu içinde kullanabilir, çağırabilir, denemelerinizi yapabilirsiniz. (Yine derste bizim yaptığımız gibi). Bu çalışmayı yaparken `C#'ta Region Kullanımı` konusunu araştırabilirsiniz isterseniz. Mecburi değil bu tabi ki, ama kodunuzu organize etme konusunda yardımcı olabilecek basit bir teknik. Ayrıca her ödevle ilgili olarak 3.sayfada ipuçları verilmiştir. Öncelikle o ipuçlarına bakmadan yapmaya çalışıp, ihtiyacınız olursa bakmanızı öneririm. Diğer yandan yapay zekadan destek almamak için direnin bu çalışmaları yaparken 😊 Araştırma yapın elbette internette ama, direkt yapay zekadan cevap istemeyin. Bunu konuştuk derste zaten, ne demek istediğimi biliyorsunuz 😊

Ödev 1: Liste Ortalaması Hesaplama

- **Metot:** double CalculateAverage(List numberList)
- **Açıklama:** Parametre olarak bir tam sayı listesi (List) alan ve bu listedeki sayıların aritmetik ortalamasını double türünde döndüren bir metot yazın.
- **Bilgi:** Aritmetik ortalama, bir veri setindeki tüm değerlerin toplamının, veri adedine bölünmesiyle elde edilir.

Ödev 2: Sesli Harf Sayacı

- **Metot:** int CountVowels(string text)
- **Açıklama:** Parametre olarak bir metin (string) alan ve bu metnin içindeki Türkçe sesli harflerin (a, e, ı, i, o, ö, u, ü) toplam sayısını int olarak döndüren bir metot yazın.
- **Bilgi:** Sesli harfler, dilin temel ses birimleridir. Bu ödevde Türkçe'deki 8 sesli harfi saymanız beklenmektedir.

Ödev 3: Üs Alma (Döngü ile)

- **Metot:** int CalculatePower(int baseNumber, int exponent)
- **Açıklama:** Parametre olarak bir taban ve bir üs değeri alan ve Math.Pow kullanmadan, bir döngü yardımıyla tabanın üssünü hesaplayıp sonucu int olarak döndüren bir metot yazın.
- **Bilgi:** Üs alma işlemi, bir sayının (taban) kendisi ile belirli bir sayıda (üs) çarpılmasıdır. Örneğin, $4^3 = 4 * 4 * 4 = 64$.

Ödev 4: Asal Sayı Avcısı

- **Metot:** bool IsPrime(int number)
 - **Açıklama:** Parametre olarak pozitif bir tam sayı alan ve bu sayının asal olup olmadığını bool (true/false) olarak döndüren bir metot yazın.
 - **Bilgi:** Asal sayı, yalnızca 1'e ve kendisine tam olarak bölünebilen, 1'den büyük pozitif tam sayılardır. Örneğin, 2, 3, 5, 7, 11 asaldır.
-

Ödev 5: Palindromik Kelimeler

- **Metot:** bool IsPalindrome(string text)
 - **Açıklama:** Parametre olarak bir kelime (string) alan ve bu kelimenin palindrom olup olmadığını bool olarak döndüren bir metot yazın.
 - **Bilgi:** Palindrom, tersten okunuşu da kendisiyle aynı olan kelime veya cümlelerdir. Örneğin: "kayak", "neden", "madam".
-

Ödev 6: Mükemmel Sayı Bulucu

- **Metot:** bool IsPerfectNumber(int number)
 - **Açıklama:** Parametre olarak pozitif bir tam sayı alan ve bu sayının "mükemmel sayı" olup olmadığını bool olarak döndüren bir metot yazın.
 - **Bilgi:** Mükemmel sayı, kendisi hariç pozitif tam bölenlerinin toplamı kendisine eşit olan sayıdır. Örneğin, 6'nın bölenleri (1, 2, 3) toplamı 6'dır, bu yüzden 6 mükemmel bir sayıdır.
-

Ödev 7: Listedeki En Büyük İkinci Sayı

- **Metot:** int FindSecondLargest(List numberList)
 - **Açıklama:** Parametre olarak bir tam sayı listesi (List) alan ve listedeki en büyük ikinci sayıyı int olarak döndüren bir metot yazın. Hazır sıralama metotları kullanmayın.
 - **Bilgi:** Bir listedeki en büyük ikinci eleman, en büyük elemandan bir küçük olan sıradaki elemandır. Listenin en az iki eleman içerdiğini varsayabilirsiniz.
-

Ödev 8: Metin Şifreleme (Sezar Şifrelemesi)

- **Metot:** string Encrypt(string text, int shiftAmount)
- **Açıklama:** Parametre olarak bir metin ve bir tam sayı (kaydırma miktarı) alan ve metindeki her harfi alfabede belirtilen miktar kadar ileri kaydırarak sonucu string olarak döndüren bir metot yazın.
- **Bilgi:** Sezar şifrelemesi, en basit şifreleme tekniklerinden biridir. Her harfin, alfabede belirli bir sayı kadar kaydırılmasıyla oluşturulur. Örneğin, 2 birim kaydırma ile "alp", "cnp" olur.

Ödev 9: Basamakları Toplama

- **Metot:** `int SumDigits(int number)`
 - **Açıklama:** Parametre olarak pozitif bir tam sayı alan ve bu sayının basamaklarının toplamını `int` olarak döndüren bir metot yazın.
 - **Bilgi:** Bir sayının basamakları toplamı, sayıyı oluşturan rakamların tek tek toplanmasıdır. Örneğin, 542 sayısı için basamaklar toplamı $5 + 4 + 2 = 11$ 'dir.
-

Ödev 10: Fibonacci Dizisi Oluşturucu

- **Metot:** `List GenerateFibonacci(int count)`
 - **Açıklama:** Parametre olarak bir eleman sayısı alan ve bu sayıda eleman içeren Fibonacci dizisini bir `List` olarak döndüren bir metot yazın.
 - **Bilgi:** Fibonacci dizisi, her sayının kendinden önceki iki sayının toplamı olduğu bir sayı dizisidir. Genellikle 0 ve 1 ile başlar: 0, 1, 1, 2, 3, 5, 8, 13...
-

İPUÇLARI

(Bu bölüme ancak bir süre kendi başınıza denedikten sonra bakın!)

1. **CalculateAverage:** Bir foreach döngüsü ile listedeki tüm elemanları bir değişkende toplayın. Döngü bittiğinde, toplamı listenin eleman sayısına (Count özelliği) bölün.
2. **CountVowels:** Metnin tamamını ToLower() metodu ile küçük harfe çevirerek işinizi kolaylaştırın. Ardından bir foreach döngüsü ile her karakterin sesli harf olup olmadığını kontrol edin.
3. **CalculatePower:** Bir sonuc değişkeni oluşturup başlangıç değerini 1 olarak atayın. Bir for döngüsünü exponent değeri kadar döndürerek her adımda sonuc değişkenini baseNumber ile çarpın.
4. **IsPrime:** Bir sayının asal olup olmadığını anlamak için, 2'den başlayıp sayının kareköküne kadar olan sayılara bölünüp bölünmediğini kontrol etmek yeterlidir. Eğer bu aralıkta hiçbir sayıya tam bölünmüyorsa, asaldır.
5. **IsPalindrome:** Boş bir reversedText string'i oluşturun. Bir for döngüsü ile orijinal metnin son karakterinden başına doğru gelin ve her karakteri reversedText'e ekleyin. Sonunda iki metni karşılaştırın.
6. **IsPerfectNumber:** Bir döngü ile 1'den başlayıp sayının yarısına kadar olan tüm sayıları kontrol edin. Eğer bir sayı, ana sayıyı tam bölüyorsa ($\text{number} \% i == 0$), bu bölüneni bir toplam değişkenine ekleyin.
7. **FindSecondLargest:** largest ve secondLargest adında iki değişken oluşturun ve başlangıçta onlara listenin ilk elemanlarını veya çok küçük bir değeri atayın. Liste içinde tek bir döngü ile gezerken, eğer mevcut eleman largest'tan büyükse, eski largest'ı secondLargest yapın ve largest'ı güncelleyin. Değilse, mevcut elemanın secondLargest'tan büyük olup olmadığını kontrol edin.
8. **Encrypt:** Bir char değişkenini int'e dönüştürerek ASCII değerini alabilirsiniz. Bu değere kaydırma miktarını ekleyip tekrar char'a dönüştürerek yeni harfi bulabilirsiniz.
9. **SumDigits:** Sayı 0'dan büyük olduğu sürece devam eden bir while döngüsü kullanın. Döngünün her adımında, sayının 10'a bölümünden kalanını ($\text{number} \% 10$) bir toplama ekleyin ve sayıyı 10'a bölün ($\text{number} = \text{number} / 10$).
10. **GenerateFibonacci:** Boş bir List oluşturun. Dizinin ilk iki elemanını (0 ve 1) listeye ekleyin. Ardından, istenen eleman sayısına ulaşana kadar bir döngü içinde listenin son iki elemanını toplayıp Add() metodu ile listeye ekleyin.