

C# TEMELİNDE HAZIRLANAN YAZILIMA YENİ BAŞLAYAN VE ÖĞRENMEKTE OLANLAR İÇİN BASİT PROGRAMLAMA DİLİ

TANIMLAMA KILAVUZU VE TASARIM SÜRECİ :

Firat Üniversitesi Yazılım Mühendisliği 3. Sınıf 185541052 numaralı öğrencisi Recep YAVUZ tarafından hazırlanmıştır.

TANIM : Yazılımın hemen her platformda kullanıldığı şu dönemde özellikle çocuklar için ve yazılıma yeni adım atan bireyler için basitleştirilmiş Türkçe komut ve kaba kod üzerinde C# temelinde sadece matematikte 4 işlemi yaptıran bir programlama dilidir.

İşlem 4 sınıfından bu projede kullanılmak üzere bir sınıf nesnesi oluşturuyoruz.

```
İslem_4 islem = new İslem_4();
```

Bu kütüphane bir C# kütüphanesi olduğu için ve console üzerinde işlem yaptığı için işlemler bittikten sonra console kendini kapatacaktır. Bu komut ile bu durumun önüne geçebiliriz ve console u program süresince istediğimiz zaman açık tutabiliriz. Yani bu komut C# dilindeki yazdırılan komutun ekranda durmasını sağlayan Console.ReadKey in hazırladığımız kütüphanede aynı işleme sahip komutudur.

```
islem.EkranAcıkTuttu();
```

Bu komut ise C# dilindeki yazdırma komutu olan Console.WriteLine nin hazırladığımız kütüphanede aynı işleme sahip komutudur.

```
islem.EkrandaGoster()
```

TOPLAMA İŞLEMİ :

Bu, girilen tek bir sayının toplamını veren işlem kodudur. Örneğin işlemi yaptırırken sadece 5 rakamı girilirse çıktı olarak 5 yazdıracaktır.

```
public int Toplamaİslemi(int a)
{
    return a;
}
```

Bu , girilen iki sayının toplamını veren işlem kodudur.

```
public int Toplamaİslemi(int a, int b)
{
    return a + b;
}
```

Bu, girilen üç sayının toplamını veren işlem kodudur.

```
}  
public int Toplamaİslemi(int a, int b, int c)  
{  
    return a + b + c;  
}
```

Bu, girilen dört sayının toplamını veren işlem kodudur.

```
}  
public int Toplamaİslemi(int a, int b, int c, int d)  
{  
    return a + b + c + d;  
}
```

Bu ,girilen beş sayının toplamını veren komuttur.

```
}  
public int Toplamaİslemi(int a, int b, int c, int d, int e)  
{  
    return a + b + c + d + e;  
}
```

Bu, girilen altı sayının toplamını veren komuttur.

```
}  
public int Toplamaİslemi(int a, int b, int c, int d, int e, int f)  
{  
    return a + b + c + d + e + f;  
}
```

Bu, girilen yedi sayının toplamını veren komuttur.

```
}  
public int Toplamaİslemi(int a, int b, int c, int d, int e, int f, int h)  
{  
    return a + b + c + d + e + f + h;  
}
```

Bu , kütüphanenin basitleştirilmiş hali olduğu için 7 sayıya kadar işlem yapılabilir. İstenilen durumda işlemi yapılan sayılar fazlalaştırılabilecektir.

ÇIKARMA İŞLEMİ :

Bu, girilen tek bir sayının çıkarma işlemini veren işlem kodudur. Örneğin işlemi yaptırırken sadece 3 rakamı girilirse çıktı olarak 3 yazdıracaktır.

```
}  
public int Cikarmaİslemi(int a)  
{  
    return a;  
}
```

Bu , girilen iki sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b)  
{  
    return a - b;  
}
```

Bu, girilen üç sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b, int c)  
{  
    return a - b - c;  
}
```

Bu, girilen dört sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b, int c, int d)  
{  
    return a - b - c - d;  
}
```

Bu, girilen beş sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b, int c, int d, int e)  
{  
    return a - b - c - d - e;  
}
```

Bu, girilen altı sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b, int c, int d, int e, int f)  
{  
    return a - b - c - d - e - f;  
}
```

Bu, girilen yedi sayının çıkarma işlemini veren işlem kodudur.

```
}  
public int Cıkarmaİslemi(int a, int b, int c, int d, int e, int f, int h)  
{  
    return a - b - c - d - e - f - h;  
}
```

Bu ,kütüphanenin basitleştirilmiş hali olduğu için 7 sayıya kadar işlem yapılabilir. İstenilen durumda işlemi yapılan sayılar fazlalaştırılabilecektir.

ÇARPMA İŞLEMİ :

Bu, girilen tek bir sayının çarpma işlemini veren işlem kodudur. Örneğin işlemi yaptırırken sadece 4 rakamı girilirse çıktı olarak 4 yazdıracaktır.

```
}  
public double Carpmaİslemi(int a)  
{  
    return a;  
}
```

Bu, girilen iki rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b)  
{  
    return a * b;  
}
```

Bu, girilen üç rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b, double c)  
{  
    return a * b * c;  
}
```

Bu, girilen dört rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b, double c, double d)  
{  
    return a * b * c * d;  
}
```

Bu, girilen beş rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b, double c, double d, double e)  
{  
    return a * b * c * d * e;  
}
```

Bu, girilen altı rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b, double c, double d, double e, double f)  
{  
    return a * b * c * d * e * f;  
}
```

Bu, girilen yedi rakamın çarpma işlemini veren işlem kodudur.

```
}  
public double Carpmaİslemi(double a, double b, double c, double d, double e, double f, double h)  
{  
    return a * b * c * d * e * f * h;  
}
```

Çarpma işleminde double kullanmamızın nedeni , kesirli sayıların çarpımı durumunda çıkacak virgüllü sonuçların da yazdırılabilmemesidir. Bu kütüphanenin basitleştirilmiş hali olduğu için 7 sayıya kadar işlem yapılabilmektedir. İstenilen durumda işlemi yapılan sayılar fazlalaştırılabilecektir.

BÖLME İŞLEMİ :

Bu, girilen tek bir sayının bölme işlemini veren işlem kodudur. Örneğin işlemi yaptırırken sadece 7 rakamı girilirse çıktı olarak 7 yazdıracaktır.

```
}  
public double Bölmeİslemi(double a)  
{  
    return a;  
}
```

Bu, girilen iki sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b)  
{  
    return a / b;  
}
```

Bu, girilen üç sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b, double c)  
{  
    return a / b / c;  
}
```

Bu, girilen dört sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b, double c, double d)  
{  
    return a / b / c / d;  
}
```

Bu, girilen beş sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b, double c, double d, double e)  
{  
    return a / b / c / d / e;  
}
```

Bu, girilen altı sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b, double c, double d, double e, double f)  
{  
    return a / b / c / d / e / f;  
}
```

Bu, girilen yedi sayının bölme işlemini veren işlem kodudur.

```
}  
public double Bölmeİslemi(double a, double b, double c, double d, double e, double f, double h)  
{  
    return a / b / c / d / e / f / e;  
}
```

Bölme işleminde double kullanmamızın nedeni kesirli sayılarla yapılan işlemlerde çıkacak virgüllü sonuçların sorunsuz olarak yazdırılabilmesidir. Bu kütüphanenin basitleştirilmiş hali olduğu için 7 sayıya kadar işlem yapılabilir. İstenilen durumda işlemi yapılan sayılar fazlaştırılabilecektir.

SON :

Genel olarak hazırladığımız programlama dilinde C# temelinde hazırladığımız kütüphanenin bazı komutlarını Türkçeleştirerek ve daha basit hale getirerek çocuk ve yazılıma yeni başlayan bireylerin kolayca programlama dillerinin genel olarak algoritmasını az da olsa anlayabilmesini sağlamaktır.