4. SIK KULLANILAN SINIFLAR

4.1. STRİNG SINIFI

- lenght: string verinin uzunluğunu verir.
- ToUpper: Büyük harfe çevirir.
- ToLower: Küçük harfe çevirir.
- StarsWith: Bir stringin belli karekterler ile başlayıp başlamadığına bakar. Geri dönüş değeri boolean'dir.
  Örn: TextBox1.Text.StarsWith(http://);
- EndsWith: Stringin sonunu kontrol eder. Geri dönüş değeri boolean'dir.
  Örn: TextBox1.Text.EndsWith (".com");
- Substring(ilk karakter, karakter sayısı): Stringin içinden parça alır. Stringin kaçıncı karakterinden başlayıp kaç karakter alacağımızı belirleriz.
- Replace("eski", "yeni"): Metnin belli bir parçasını başka bir metin ile değiştirir. Tek bir karekteride değiştirebilir. Yani karakter, boş ta bırakılabilir(null).
- Insert(sıra, "metin"): Metnin içinde bir yere bir metin, karakter ya da boşluk ekler. Araya girer.
- Remove(Sıra, Karakter sayısı): Metnin içinde bir kısmı siler. Kaçıncı karakterden başlayıp kaç karakter sileceğimizi belirleriz.
  Örn: Remove(2, 5);
- IndexOf("metin"): Bir metnin içinde bir metin arama yapar. Boşluk ya da tek bir karakterde aranabilir. Geriye metnin ilk karakterinin sırasını verir yoksa geri -1 gönderir.
- LastIndexOf("metin"): Aranan metnin en son nerede geçtiğinin sırasını verir.
  Not: Bu işlemlerde nereden itibaren aranacağı söylenebilir.
  Örn: IndexOf("metin", 4);
- String.Concat(Str1, Str2): İki farklı metini birleştirip tek metin haline getirir.
- Split(' '): Cümleyi böler. Verilen belli bir karaktere göre metni parçalar. Karakter boşluk olabilir.
- Contains("değer"): Bir string içerisinde bir metin veya karakter olup olmadığını boolean oarak geri döndürür.
- PadRight(karakter sayısı;' karakter'): Girilen bir metnin belli bir karakter sayısına kadar belli bir karakter ile doldurulmasını sağlar. PadRight ile metnin sağına yani sonuna istenilen karakter eklenir.
- PadLeft(karakter sayısı;'karakter'): PadRight ile aynı amaç için kullanılır. PadLeft ile metnin soluna yani başına istenen karakter eklenir.

4.2. CHAR SINIFI

- CharGetNumericValue(' '); Karakterin sayı olup olmadığını kontrol eder değilse -1 verir.
- Char.ToUpper(' ');
- Char.ToLower(' ');
- Char.IsNumber(' '); Sayımı değil mi diye kontrol eder boolean olarak geri döner.
- Char.IsLetter(' '); Harf mi değil mi kontrol eder boolean olarak geri döner.
- Char.IsUpper(' '); Karakter büyük mü değil mi diye kontrol eder.
- Char.IsLower(' ') Karakter küçük mü değil mi diye kontrol eder.
- Char.Parse("karakter"): Stringi karaktere (char) tipine dönüştürür.

4.3. MATEMATİK (MATH) SINIFI

C# İçerisindeki Matematiksel Fonksiyonların Kullanımı

C# uygulamalarında operatörler yardımıyla temel matematiksel işlemler gerçekleştirilebilir. Fakat operatörlerle yapabileceğimiz işlemler sınırlıdır, buna bağlı olarak .Net içerisinde matematiksel hesaplamaları yapmamızı sağlayan **Math** sınıfı tasarlanmıştır.

Math sınıfı *static* olarak tanımlandığı için yavru değişken oluşturulmadan direk olarak içerisindeki metot ve değerlere erişerek işlemleri gerçekleştirebiliriz.

Abs: Bu fonksiyon verilen herhangi bir sayının pozitif değerini döndürür. Diğer bir ifadeyle kendisine parametre olarak verilen sayının, sayı doğrusunda ki 0 noktasına olan uzaklığını gösterir. Örnek;

```
int deger1 = -30;
int pozitifDeger = Math.Abs(deger1);
Console.WriteLine("Sonuç:" + pozitifDeger);
```

Ceiling: Kendisine parametre olarak verilen ondalıklı değeri, üsteki sayıya yuvarlayarak döndürür. Girilen sayı "9.3" veya "9.8" olmasının bir önemi yoktur. İki işlemde de sonuç 10 çıkacaktır. Örnek;

```
double deger1 = 9.3;
double yuvarla = Math.Ceiling(deger1);
Console.WriteLine("Sonuç:" + yuvarla);
```

Exp: Parametre olarak girilen sayıyı, e sayısının kuvveti olarak hesaplar. e sayısının yaklaşık değeri 2.71'dir ve logaritmik hesaplamalarda sıklıkla kullanılır.

```
double sayi = Math.Exp(3);
Console.WriteLine("Sonuç:" + sayi); //Yaklaşık 2.71'nin 3.kuvvetini
hesaplar
```

E: Matematikteki e sayısını ifade eder, yaklaşık değeri 2.71'dir. Bu değer bir sabit olmakla birlikte, metot değildir.

```
double sayi = Math.E;
Console.WriteLine("Sonuç:" + sayi);// 2.71828182845905
```

Floor: Kendisine parametre olarak girilen değeri kendisinden küçük olan tamsayıya yuvarlar. Ceiling kendisinden büyük sayıya yuvarlama işlemi yaparken, Floor kendisinden küçük sayıya yuvarlama yapar.

```
double sayi = 9.8;
double yuvarla = Math.Floor(sayi);
Console.WriteLine("Sonuç:" + yuvarla);
```

Log: Birinci parametrede girilen sayının, ikinci parametrede girilen sayıya göre logaritmasını alır.

Max: Kendisine parametre olarak girilen iki değerden büyük olanı döndürür.
Min: Kendisine parametre olarak girilen değerden küçük olanı döndürür.

```
double buyuk = Math.Max(2, 3.45);
Console.WriteLine("Maksimum:" + buyuk);
double kucuk = Math.Min(-4.3, 3);
Console.WriteLine("Minimum:" + kucuk);
```

Pow: Matematikteki üst hesaplama işlemini gerçekleştirir. Birinci parametre taban, ikinci parametre üs olacak şekilde üslü ifadeyi hesaplar.

```
double yaricap = 4.6;
double alan = Math.PI * Math.Pow(yaricap, 2);// Alan= pi*r²
Console.WriteLine("Dairenin Alanı:" + alan);
```

Round: Kendisine verilen ondalıklı sayının ondalık hassasiyetini ayarlar. Birinci parametre asıl sayıyı, ikinci parametre kaç basamak ondalık gösterilecekse bunu belirler.
`double sayi = 4.698678;`
`double ayarla = Math.Round(sayi, 3);`
`Console.WriteLine("Sonuç:"+ayarla);`

Sign: Parametre olarak verilen sayının pozitif, negatif veya sıfır olma durumunu gösterir. Yani sayı pozitif ise "1", negatif ise "-1", sıfır ise "0" değerini döndürür.

Sqrt: En çok kullanılan matematiksel metotlardan biridir. Kendisine verilen sayının karekökünü hesaplar.
`double sayi = 625;`
`double kok = Math.Sqrt(sayi);`
`Console.WriteLine("Sonuç:"+kok);`

PI: Matematikteki π (pi) sayısını döndürür. Buda e gibi bir sabit olmakla birlikte, metot değildir. PI sayısının yaklaşık değeri 3.14'tür.
Sin – Cos – Tan – Sinh – Cosh - Tanh: Trigonometrik açıların değerlerini hesaplamak için kullanılır. Açılar Radyan cinsinden girilmelidir.

4.4. DATE-TIME SINIFI

DateTime metodları sistem saatine göre çalışır. Tarih ve saat bilgilerini sistemden alır.

- DateTime.Now: Bugünün tarihi ve saatini verir.
- DateTime.Now.Add : Bugünün zamanına istenilen birimde ekleme ya da çıkarma yapmak için kullanılır. Saat, Gün, Ay, Yıl v.b. eklenebilir veya çıkarılabilir. Çıkarmak için negatif değer verilir.

  Örnek: .AddDays(miktar)
- DateTime.Now.AddMonths(miktar)
- .AddYears(miktar)
- .AddHours(miktar)
- TimeSpan sınıfı: Tarih saat bilgisine ekleme ya da çıkarma yapmak için metotları ayrı ayrı kullanmak yerine TimeSpan nesnesi ile toplu ekleme/çıkarma yapılabilir.
  Örnek:
  TimeSpan trh = new TimeSpan:(2, 10, 20) (Saat, Dakika, Saniye)
  Label1.Text=DateTime.Now.Add.(trh).ToString();
- DateTime.Day
  .Month
  .Yaer
  .DayofYear
  .DayofWeek(gün adı)
  Geçerli tarih için gün, ay, yıl, yılın kaçıncı günü, haftanın hangi günü gibi bilgiler yukarıdaki metotlar kullanılarak ayrı ayrı alınabilir.
- DateTime.Now.ToShortDateString(): Sadece tarih bilgisini kısa formatta verir.
  (Örn: 05.12.2018)
- DateTime.Now. ToLongDateString(): Sadece tarih bilgisini uzun formatta verir.
  (Örn: 15 Aralık 2014 Pazartesi)
- DateTime.Now. ToShortTimeString(): Sistem saatinden sadece saati kısa formatta alır.
  Örnek: 17.19.25 – saat.dakika.saniye
- DateTime.Now. ToLongTimeString(): Sistem saatinden sadece saati uzun formatta alır.
  Örnek: 17.19.25.40 – saat.dakika.saniye.milisaniye