Masaüstü Uygulama Geliştirme

Hafta-4

Öğr. Gör. Zafer SERİN

Console.ReadLine() Kullanımı

- C# dilinde Console.ReadLine() ile kullanıcıdan değer alınabilir. Bu kod varsayılan olarak string tipte değer alır. Bu nedenle eğer aritmetiksel işlem yapılmak isteniyorsa; ilgili değerin uygun bir sayısal tipe(int, long vb.) çevrilmesi gerekir.

```
Console.WriteLine("Lütfen 1. sayiyi giriniz..:");
int sayi1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(sayi1);
```

Aritmetik Operatörler

- C# ile aritmetiksel işlemleri yapabilmek için aşağıdaki operatörler kullanılır.

Operatör	İşlevi
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Mod Alma

Karşılaştırma Operatörleri

- C# ile karşılaştırma işlemleri yapabilmek için aşağıdaki operatörler kullanılır.

Operatör	İşlevi
<	Küçüktür
>	Büyüktür
<=	Küçük veya eşittir
>=	Büyük veya eşittir
==	Eşittir
!=	Eşit değildir

Mantıksal Operatörler

- C# ile mantıksal işlemleri yapabilmek için aşağıdaki operatörler kullanılır.

Operatör	İşlevi
&&	ve
||	veya
^	Ya da

İşlem Önceliği

- C#'ta işlem önceliği matematikte olduğu gibidir.

En Yüksek
()
!
*, /, %
+, -
<, >
= =, !=
&&
||
En Düşük

Artırma ve Azaltma Operatörleri

- C# ile artırma ve azaltma işlemlerini yapmak için ++ ve - - operatörleri kullanılır. Bunlar ilgili değeri 1 artırma veya 1 azaltma işlemine tabi tutarlar. Operatörün değişkenin solunda olması durumunda ilgili değer o satırda işlem görürken, operatör değişkenin sağında olursa ilgili değer bir sonraki satırda işlem görür.

Üzerine Ekleme Yığma Operatörleri

Operatör	Örnek Kullanım	Dengi
+=	İ += 5	İ = i+5
-=	İ -= 3	İ = i-3
*=	İ *= 4	İ = i*4
/=	İ /= 7	İ = i/7
%=	İ %= 2	İ = i%2

sizeof Operatörü

- C#'ta sizeof operatörü ile değişken tiplerinin kaç bytelık yer kapladığını integer olarak geriye döndürür. Doğrudan değişken adı ile çağrılamaz!

```
Console.WriteLine(sizeof(int));
Console.WriteLine(sizeof(long));
Console.WriteLine(sizeof(double));
```

typeof Operatörü

- Typeof operatörü verilen değerin tipini getirir. O tür ile ilgili bilgileri edinmek için kullanılan bir operatördür. Doğrudan değişken adı ile çağrılamaz!

```
string isim = "Zafer";
Type t = typeof(string);
Console.WriteLine(t);
```

is ve is null Operatörleri

- Boxing'e tabi tutulmuş bir değerin öz türünü öğrenebilmek için is operatörü kullanılır. True veya False olarak değer döndürür.

```
object sayi = 25;
Console.WriteLine(sayi is int);
```

- Değer tipli değişkenler (string hariç şu ana kadar gördüklerimiz) null değer alamaz. Null değer alabilen bir değişkenin değerinin null olup olmadığını kontrol etmek için is null operatörünü kullanırız.

```
string isim = null;
Console.WriteLine(isim is null);
```

as Operatörü

- Boxing edilmiş bir değeri dışarı çıkarmayı sağlar. () -> cast işleminden farkı eğer değer uymuyorsa hata vermek yerine null değer döndürür. Değer tipli değişkenler normal şartlarda null değer alamayacağı için buna uygun kullanılmalıdır.

```
object isim = 15;
string y = isim as string;
Console.WriteLine(y);
```

Değer tipli bir değişkeni null yapma

- Değer tipli bir değişkeni null yapabilmek için ? Kullanılır.

```
int? sayi = null;
```

?? Operatörü

- Elimizdeki değişkenin değerinin null olma durumuna istinaden farklı bir değeri göndermemizi sağlayan operatördür. Sol ve sağdaki türler aynı olmalıdır.

```
string isim = null;
Console.WriteLine(isim ?? "SERİN");
```

??= Operatörü

- Elimizdeki değişkenin değeri null ise yeni değeri değişkene atamayı sağlar. Sol ve sağdaki türler aynı olmalıdır.

```
string isim = null;
Console.WriteLine(isim ?? "SERİN");
Console.WriteLine(isim);

int? id = null;
id ??= 1;
```