

ANKARA ÜNİVERSİTESİ GAMA MESLEK YÜKSEKOKULU

* BMT255 ALGORİTMA VE PROGRAMLAMA

* Öğr.Gör.Uğur YEDEKÇİOĞLU

DEĐİŐKENLER VE VERİ TÜRLERİ

DeđiŐken Tanımlama Kuralları

DeđiŐken tanımlarken (Aynı zamanda ad alanları, yöntemler ve sınıflar tanımlanırken de bu kurallara uyulmalıdır.) aŐađıdaki söz dizimi kurallarına uyulmalıdır.

DeđiŐken adlarında yalnızca harf (büyük ve küçük), rakam ve alt çizgi(_) karakteri kullanılabilir, boşluk içermez.

DeđiŐken adı, bir harf ile (Alt çizgi harf olarak kabul edilir.) başlamak zorundadır, rakamlarla başlayamaz.

Türkçe karakterler kullanılabilir ama önerilmez(ç,đ,Ő,ö,ü,ı,i vb.).

Programlama dili bazı tanımlayıcı kelimeleri kendisi kullanmak için ayırmıŐtır.

* **Değişken Tanımlama**

- * Değişkenler değerleri tutar. Programlama dili depolayabileceği ve işleyebileceği çok sayıda farklı değer türüne sahiptir(Tam sayı, kayan nokta sayısı ve karakterler dizesi gibi).
- * Bir değişken bildirildiğinde içinde ne tür bir değişken tutacağı da belirtilmelidir.
- * Örneğin, aşağıdaki ifade “*yas*” adında ve “*int*” (tamsayı) değerler taşıyan bir değişkeni bildirir. İfade sonunda noktalı virgül yer alır.
- * `int yas;`
- * Değişken bildirdikten sonra değişkene değer atanabilir. Aşağıdaki deyim *yas* değişkenine 30 değerini atar.
- * `yas = 30;`
- * Ayrıca, bildirim ve atama işlemleri aynı anda yapılabilir.
- * `int yas = 30;`

Ayrıca, bildirim ve atama işlemleri aynı anda yapılabilir.

```
int yas = 30;
```

Eşittir işareti(=), sağ taraftaki değeri, sol taraftaki değişkene atayan bir **atama** işlemidir. Bu atamadan sonra **yas** değişkeni tuttuğu değere başvurmak için kod da kullanılabilir.

Aşağıdaki deyim, konsola **yas** değişkeninin değeri olarak 30 yazar:

```
Console.WriteLine(yas);
```

* Temel Veri Türleri

| Veri Türü | Açıklama | Boyut(bit) | Aralık | Örnek Kullanım |
|-----------|---|------------------------|--|----------------------------------|
| int | Tam sayılar | 32 | -2^{31} den $2^{31}-1$ ' e kadar | int numara; numara = 30; |
| long | Tam sayılar (daha büyük aralık) | 64 | -2^{63} den $2^{63}-1$ ' e kadar | long saniye; saniye = 22L; |
| float | Kayan noktalı sayılar | 32 | $\pm 1,5 \times 10^{45}$, ten $\pm 3,4 \times 10^{-38}$, e kadar | float ölçüm; ölçüm = 0.52F; |
| double | İki kat daha hassas(daha fazla kesin) kayan noktalı sayılar | 64 | $\pm 5,0 \times 10^{-324}$, ten $\pm 1,7 \times 10^{308}$, e kadar | double miktar; miktar = 0.15; |
| decimal | Parasal değerler | 128 | 28 basamak | decimal fiyat; fiyat = 25M; |
| string | Karakter sıraları | Karakter başına 16 bit | Geçersiz | string ad; ad = "Ali"; |
| char | Tek karakter | 16 | 0'dan $2^{16}-1$ ' e kadar | char secim; secim = '*'; |
| bool | Boolean | 8 | Doğru yada Yanlış | Bool tercih; tercih = true; |

- * İşlem Sırası ve Operatörler
- * Aşağıdaki tabloda verilmiştir.

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

İf Yapısı

Kullanımı:

```
if(koşul)
```

```
{
```

➔ Koşul gerçekleşirse bu satırlar işletilir.

```
}
```

Örnek

```
if(not>60)
```

```
{
```

```
Console.WriteLine("Geçti");
```

```
}
```

Açıklaması: Eğer not 60 dan yüksekse ekrana geçti yaz.

İf Else Yapısı

Kullanımı:

```
if(koşul)
```

```
{
```

→ Koşul gerçekleşirse bu satırlar işletilir.

```
}
```

```
Else
```

```
{
```

→ Koşul gerçekleşmez ise bu satırlar işletilir.

```
}
```

* For Yapısı

Kullanımı:

```
For(başlangıç;bitiş;artış/azalış)
```

```
{
```

→ Bu satırlar yukarıda tanımlanan adet kadar tekrar eder.

```
}
```

Örnek:

```
for(int i=1;i<=10;i++)  
{  
Console.WriteLine("Öğr.Gör.Uğur YEDEKÇİOĞLU");  
}
```

Açıklama: Başlangıç 1, Bitiş 10 ve artış miktarı 1 için for döngüsü 10 defa tekrar eder. Dolayısıyla ekrana 10 defa 'Öğr.Gör.Uğur YEDEKÇİOĞLU' yazar.

Ekran çıktısı aşağıdadır.

Kaynakça

<http://immibilisim.com/moduller/1-%20Nesne%20Tabanlı%20Programlamaya%20Giriş.pdf>