

Bazı Önemli Kavramlar

Emre Altunbilek Java Dersleri

İlk Proje Öncesi Şunları Bilelim

1. Her java programında en azından bir tane sınıf olmalıdır.
Her sınıfın bir adı vardır ve her zaman büyük harfle başlayalım ve ikinci kelimemiz de büyük harfle başlasın.
Bu zorunlu değildir. Lütfen türkçe karakter de kullanmayalım.
Sınıf adı örnekleri : HelloWorld, IlkSinif, OgrenciDetay vb...
2. Bir sınıfın çalışması için main isimli bir metota sahip olmalıdır. Burası bütün uygulamanın başlangıç noktasıdır.
3. Statement: bir aksiyonu veya aksiyonlar sırasını belirtir.
Merhaba Dünya ifadesinin gösterilmesine statement denir. Her statement ; ile sonlanır. O yüzden her satırın sonunda mutlaka ; olmalıdır.
4. Javada bazı kelimeler reserved words olarak geçer. Bu kelimelerin java için bir anlamı vardır. O yüzden bu kelimeleri kendi kodlarımızın içinde kullanmamalıyız. Çünkü bunlar java için önemli anlamlara sahiptir.
Örnek : class , static , public , void ...
5. Bir kod bloğunu belirtmek için { } süslü parantezleri kullanılır. Bu bloklar programımızı oluşturan parçalardır.

Bunların dışında da önemli özel semboller bulunur. Dersler ilerledikçe hepsinin açıklamasını öğreneceğiz ama şuan için bu semboller :

{ } --> statement blokları

() --> metotlarda kullanılır

[] --> dizi (array) lerde kullanılır

/ / --> yorum satırı için

" " --> metinsel ifadeler için

; --> statementları sonlandırmak için kullanılan bazı özel sembollerdir.

Değişken Variable Kavramı

Emre Altunbilek Java Dersleri

Değişkenler

- Girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı veri tutucu yapılardır.
- Alacakları değerlere göre türü belirlenmelidir.
- Değişken adları rakamla başlayamaz, bu isimlerde büyük-küçük harf duyarlılığı vardır, örneğin sayi1 ve Sayi1 farklı değişkenleri temsil eder.
- Reserved wordleri kullanarak değişken adı oluşturulamaz.
- Aynı kod bloğu içinde aynı isimden birden fazla değişken tanımlanamaz.
- Boşluk ve /, *, +, - gibi özel karakterler içermemelidir.
- Değişken tanımlamanın genel formülü :
- Değişken türü değişken adı = değeri
`int sayi1 = 10 ;`

[Peki javada bulunan değişken türleri nelerdir?](#)

Numerik Veri Tipleri

Emre Altunbilek Java Dersleri

Javadaki Sayısal Veri Tipleri

Bu veri tipi temel, ilkel (primitif) veri tiplerinden biridir. Genel olarak primitif veri tipleri bellekte sabit yer tutarlar.

Range of numeric data types in Java

Type	Size	Range
byte	8 bits	-128 .. 127
short	16 bits	-32,768 .. 32,767
int	32 bits	-2,147,483,648 .. 2,147,483,647
long	64 bits	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
float	32 bits	$3.40282347 \times 10^{38}$, $1.40239846 \times 10^{-45}$
double	64 bits	$1.7976931348623157 \times 10^{308}$, $4.9406564584124654 \times 10^{-324}$

Literal Kavramı

```
int i = 36;
long y = 524200;
double v = 5.9;
```

Direk olarak programın içinde tanımlı olan sabit değerlere (36, 524200, 5.9) literal denir.

Bir literalı bir değişkene atarken veri türüne dikkat etmeli ve bu veri türünün taşıyabileceği büyüklükte değer atamalısınız. Örneğin byte veri türü maksimum 127 sayısını alır, 128 atayamazsınız.

Float ve Double değerleri belirtirken yanlarına f veya F, d veya D belirtmelisiniz.

Örneğin float bir değer için 250.5F veya double bir değer için 524.985d gibi.

Sayısal veri tiplerini kullanırken ilk değer atanmaz ise java otomatik olarak bu değişkenlere 0 değerini atar.

int sayi1 = 5; bu sayi1 değişkenine 5 değerini atarken int sayi2 ; gibi bir tanım yaparsak bellekte sayi2 için ayrılan alana 0 değeri yazılır.

Char ve Boolean Veri Tipleri

Emre Altunbilek Java Dersleri

Char Veri Tipi

Java'da tek tırnak içinde basit karakterlerin saklanması için char veri tipi kullanılır.

```
char birHarf = 'e';  
char birSayi = '4';  
birSayi isimli değişkene tek tırnaklar içinde 4 değeri  
atanırsa bu artık rakam ve tam sayı olan 4 olarak değil bir  
karakter olarak algılanır.
```

Eğer bir char değişkenine ilk atama yapılmazsa java otomatik olarak '\u0000' unicode değerini atar.

char değerler bellekte 16 bitlik yer kaplar.

Boolean Veri Tipi

Sadece true veya false değerini alan bir veri türüdür.

Bellekte 1 bit yer kaplar.

İlk değeri atanmazsa java varsayılan olarak false değerini atar.

```
boolean ciftSayiMi = true ;
```

Wrapper Kapsayıcı Sınıflar

Emre Altunbilek Java Dersleri

Wrapper Sınıflar Nedir ?

Şu ana kadar öğrendiğimiz tüm veri türleri java tarafından nesne olarak algılanmamaktadır. Belki de bu yüzden bu veri türlerine ilkel veri tipi diyoruz.

İlerleyen zamanlarda sınıf ve nesne kavramlarını öğrenince bu konu çok daha iyi algılanacaktır ancak şuan için bu tanımı yapabiliriz.

Her ilkel veri türünü sarmalayan bir sınıf bulunmaktadır ve veri türünün büyük harfle başlayan versiyonudur.

Örneğin int => Integer, double => Double vb..

Bu sınıf nesneleri null değer alabilir ve özellikle nesneye yönelik programlar yazarken çok yerde işimize yarayacaktır.

Şuan için veri türlerimizin alacağı minimum ve maksimum değerler için bu sınıfların kullanımlarını göreceğiz.

```
System.out.println("En büyük integerdeğer :" + Integer.MAX_VALUE);  
System.out.println("En büyük float değer :" + Float.MAX_VALUE);
```

Tip Dönüşümleri Type Casting

Emre Altunbilek Java Dersleri

Değişkenler ve Türleri

Javada küçük olan tamsayı veri türü otomatik olarak büyük olan veri türüne dönüştürülür. Yani siz long veri tipine sahip bir değişkene integer bir sayı atayabilirsiniz.

Ama tam tersi bir durumda veri kaybı da yaşanabileceği için tip dönüşümü yani type casting işlemi yapmanız gereklidir.

Ayrıca tam sayı tutan byte, short, int ve long veri tipine sahip değişkenlere noktalı ifadeler içeren float veya double değerler atanamaz. Böyle durumlarda virgülden sonraki kısmın kaybedileceği gerçeğini kabul ederek tip dönüşümü yapılabilir.

```
int sayi = (int) 7,75; // sayi değişkenine atanan değer 7 olur.
```

Aşağıdaki ifadeler ne anlama geliyor ?

```
int a1=Integer.valueOf(s1); //a1=12  
int a2=Integer.parseInt(s2); //a2=20
```

```
System.out.println(a1+a2);
```

```
int b1 = 80;  
int b2 = 90;  
System.out.println(b1+b2);
```

```
String t1=String.valueOf(b1);  
String t2=String.valueOf(b2);
```

```
System.out.println(t1+t2);
```

Aşağıdaki ifadeleri hesaplayıp ekrana bastırın ve sonuçları yorumlayın.

```
int sayi1 = 5 / 3 ;  
float sayi2 = 5f / 3f ;  
double sayi3 = 5d / 3d;
```

Aşağıdaki ifadelerin sonucu nedir ve sonucu yorumlayın.

```
System.out.println(1.0 - 0.1 - 0.1 - 0.1 - 0.1 - 0.1);
```

```
System.out.println(1.0 - 0.9);
```

Aşağıdaki ifadelerin sonuçlarını ekrana yazdırın ve sonucu yorumlayın.

```
int number1 = 1;  
int number2 = 2;  
double average = (number1 + number2) / 2;  
System.out.println(average);
```

(a)

```
int number1 = 1;  
int number2 = 2;  
double average = (number1 + number2) / 2.0;  
System.out.println(average);
```

(b)