Sabitler

Program içerisine doğrudan yazılan sayılara sabit denir. Yalnızca değişkenlerin değil sabitlerin de türleri vardır. Bir sabitin türü sayının niceliğine ve sonuna getirilen eklere bağlıdır. Sabitlere ilişkin kurallar şöyledir:

1. Sayı nokta içermiyorsa ve sonuna herhangi bir ek almamışsa ve int türünün sınırları içerisinde kalıyorsa sabit int türdendir. Sayı int türünün sınırları dışında kalırsa error oluşur. Örneğin:

1008 → int

2. Sayı nokta içermiyorsa ve sayının sonunda küçük veya büyük harf L varsa (L veya l), sabit long türdendir. Sayı long türünün sınırları dışında kalırsa error oluşur. Örneğin:

1008L → long
67l → long

3. Java programlama dilinde byte ve short türlerinden sabit yoktur. Ancak int türden bir sabit byte veya short türünün sınırları içerisinde kalıyorsa sırasıyla byte veya short türüne doğrudan atanabilir. Örneğin:

short a;

a = 100;

4. Sayı nokta içeriyorsa ve sonunda ek yoksa sabit double türdendir. Örneğin:

11.8 → double
11. → double
.34 → (0.34) double

5. Sayı nokta içersin ya da içermesin sonunda küçük harf veya büyük harf f varsa (F veya f) sabit float türdendir. Örneğin:

1008F → float
11.8f→ float

6. Sayı nokta içersin ya da içermesin sonunda küçük harf ya da büyük harf d varsa (D veya d) sabit double türdendir. Örneğin:

1008D → double
11.8d → double

7. Tek tırnak içerisindeki karakterler char türden sabit belirtir. Örneğin:

'a' → char
':' → char
' ' → char

Tek tırnak içerisine yalnızca tek bir karakter yerleştirilebilir. Örneğin:

'ab' → error!

Tek tırnak içerisindeki karakterler aslında o karakterin UNICODE tablodaki sıra numarasını belirten sayılardır. Örneğin:

'a' + 1 → 98 elde edilir.

Anahtar Notlar: Java programlama dilinde int türden bir sabit char türünün sınırları içerisinde kalıyorsa char türden bir değişkene doğrudan atanabilir. Atanan bu sayı UNICODE tablosundaki sıra numarası olarak ele alınır.

Bazı karakterlerin görüntü karşılığı yoktur. Bunlar ekrana yazdırılmak istenirse bir olay gerçekleşir. Bu tür karakterlere bastırılamayan (nonprintable-nongraphic) karakterler denilmektedir. Bastırılamayan karakterlerin bazıları özel bir yöntemle belirtilmektedir. Bu karakterlere escape sequence denmektedir. Tek tırnak içerisinde önce bir ters bölü sonra özel bazı karakterler bazı UNICODE karakterleri temsil eder. Java programlama dilinde bu karakterlerin listesi söyledir:

'\b' → backspace
'\t' → horizontal tab
'\n' → line feed
'\f' → form feed
'\r' → carriage return
'\0' → null character

Ters bölü karakterleri iki tırnak içerisinde (string atomlarında) tek bir karakter olarak kullanılır.

Örneğin:

```
package csd;

public class App {
	public static void main(String [] args)
	{
		System.out.println("C:\\user\\csystem"); // C:\user\csystem
	}
}
```

Ters bölü karakterinin kendisi '\\' biçiminde belirtilir. Tek tırnak karakterinin kendisi çift tırnak içerisinde doğrudan kullanılabilir. Ancak karakter sabiti olarak '\'' biçiminde yazılmalıdır. Örneğin:

```
package csd;

class App {
    public static void main(String [] args)
    {
        char ch = '\'';
        System.out.println("'ankara'istanbul");
        System.out.println(ch);
    }
}
```

Çift tırnak karakterinin kendisi çift tırnak içerisinde doğrudan kullanılamaz. Çift tırnak karakteri çift tırnak içerisinde \" biçiminde yazılmalıdır. Çift tırnak karakteri tek tırnak içerisinde doğrudan kullanılabilir. Çift tırnak karakter sabiti ayrıca '\"' biçiminde de yazılabilir. Örneğin

```
package csd;

class App {
    public static void main(String [] args)
    {
        char ch1 = '"', ch2 = '\"';

        System.out.println("\"ankara\"");
        System.out.println(ch1);
        System.out.println(ch2);
    }
}
```

8. boolean türden iki sabit vardır. Bunlar true ve false sabitleridir.

Tam Sayıların Bazı Sayı Sistemlerde Yazılışı

Tam sayı sabitleri çeşitli sayı tabanlarında (sistemlerinde) yazılabilmektedir:

- Onluk Sistem (Decimal): Doğrudan rakamlar (0, 1, …, 9) kullanarak yazılan sabitler. Örneğin:

1008 → decimal

- Onaltılık Sistem (Hexadecimal) : Sabit onaltılık tabanda yazılacaksa önce sabitin önüne 0x veya 0X (sıfır ve x/X) yazılmalıdır. Daha sonra 0 ve 9 arasındaki rakamlar ve A, B, C, D, F karakterleri kullanılabilir. Bu karakterler küçük harf olarak da yazılabilir. Örneğin:

0x1FC0 →hexadecimal
0x1fc0 → hexadecimal

- Sekizlik Sistem (Octal): Sabit sekizlik tabanda yazılacaksa önce sabitin önüne 0(sıfır) yazılır daha sonra 0 ve 7 (7 dahil) arasındaki rakamlar kullanılabilir. Örneğin:

067 → octal

- İkilik sistemde (Binary): Bu şekilde yazılış Java 7 den itibaren Java programlama diline eklenmiştir. Sabit ikilik sistemde yazılacaksa önce sabitin önüne 0b veya 0B (sıfır ve b/B) yazılmalıdır. Daha sonra 0 ve 1 rakamları kullanılabilir. Örneğin:

0b0001111111000000 → binary

Gerçek Sayı Sabitlerinin Üstel Gösterilişi

Gerçek sayı sabitleri bilimsel gösteriliş (scientific notation) diye de adlandırılan üstel biçimde de yazılabilir. Bu yazım için küçük harf ya büyük harf e (E veya e) kullanılır. E veya e den sonra gelen

sayı 10 sayısının kuvvetini göstermektedir. Sabitin değeri E veya e den önce gelen sayı ve E veya e den sonra gelen sayı ile 10 un kuvveti alınarak elde edilen sayının çarpımıdır. Örneğin:

3.145E2 → 314.5
108e-2 → 1.08
3.145E+2 → 314.5

Karakter Sabitlerini \u ile yazılması

Karakter sabitleri tek tırnak içerisinde \u ile UNICODE tablosundaki sıra numarası kullanılarak yazılabilir. Örneğin:

'\u00F1 → \u ile hexadecimal yazılış

Burada 00F1 onaltılık sistemde belirtilmiştir ve ilgili karakterin UNICODE tablosundaki sıra numarasıdır. Örneğin, bu sabit bir değişkene atanacaksa 00F1 sayısı değişkene aktarılır.

Sabitler Yazılırken _(alttire) karakterinin kullanımı

Java 7' den itibaren herhangi bir türden sabit içerisinde ve herhangi bir yazılışında _(alttire) karakteri kullanılabilmektedir. Bu kullanım okunabilirliği artırmak için dile eklenmiştir. Örneğin:

0b0001_1111_1100_0000

Aşağıdaki durumlarda alttire kullanımı geçersizdir. Bunlar dışında sabitin herhangi bir yerinde istenilen sayıda kullanılabilir:

- Karakter sabitlerinde

'\u00_FF' → geçersiz

- Sayının başında veya sonunda. Örneğin:

67_ → geçersiz
_67 → geçersiz
1.56_ → geçersiz
_1.56 → geçersiz

- Sayı nokta içeriyorsa noktadan önce veya sonra. Örneğin:

11._08 → geçersiz
11_.08 → geçersiz

- Sayının sonuna eklenen tür belirten eklerden önce veya sonra. Örneğin:

1008_L → geçersiz
1008L_ → geçersiz

11.08_F → geçersiz
11.08F_ → geçersiz
3.1459_D → geçersiz
3.1459D_ → geçersiz
0x1FC0_ → geçersiz

- Tam sayı sabitlerinin onluk sistem dışında yazılışlarında sayının sistemini belirten öneklerden sonra veya önekler arasında veya öneklerin başında. Örneğin:

0x_1FC0 → geçersiz
0_x1FC0 → geçersiz
_0x1FC0 → geçersiz

- Gerçek sayı sabitlerinin üstel yazılışında E veya e den önce veya sonra

1.23_E+2 → geçersiz
1.23e_+2 → geçersiz

- Yazılan sabitin herhangi basamağı yazısal olarak elde edilecekse: Bu konu ileride ele alınacaktır.

Sabitlerin içerisinde alttire kullanımına ilişkin geçerli örnekler:

0b1111_1010_0011_1100_0011_1100_0011_1100 → geçerli
0b11111010_00111100_00111100_00111100 → geçerli
3.141_592_653_589_793 → geçerli
10_____8 → geçerli
0x1ABC_1FC0 → geçerli
2_000_000L → geçerli