

İlkel Veri Türleri (primitive types)

Tablo 1.1. Tamsayı tipleri

| Tip | Büyüklik | Kapsadığı Değerler |
|-------|----------|--|
| 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 |

Karakter Sapmaları

Kesme işareti ya da tek tırnak diye adlandırdığımız (') simgesi, java için özel anlam taşır. Bu nedenle, Yukarıdaki merinde geçen "Türkiye'nin" ifadesini, "Türkiye\'nin" biçiminde yazmalıyız. (') simgesinin önüne gelen (\) simgesi, java'ya (') nin javada ön-tanımlı işlevinden farklı bir anlam taşıdığını bildirir ve onun karakter olarak yazılmasını sağlar. Hemen her programlama dilinde olduğu gibi, asıl işlevinden sapıran karakter kodları (escape character codes) vardır. Bunları aşağıdaki Tablodan görebiliriz: Şimdi, yukarıdaki programı biraz değiştirelim. Bu işi yaparken java.lang paketi içindeki String class'ını kullanacağız. String class'ı, adından da anlaşıldığı üzere, metin işlemleri için gerekli olan bütün işlemleri yapmamızı sağlayacak değişkenlere ve metotlara (fonksiyonlar) sahiptir.

Tablo 1.2. Karakter Sapmaları

| Sapma (Escape) | Anlamı |
|----------------|--|
| \n | Yeni satır |
| \t | Tab |
| \b | Geri tuşu |
| \r | Satırbaşı |
| \f | Form al (formfeed) |
| \\ | Ters bölü çizgisi |
| \' | Tekli kesme işareti |
| \" | Çifte kesme (tırnak) |
| \ddd | Octal (Sekiz tabanlı) |
| \xdd | Hexadecimal (onaltı tabanlı) |
| \udddd | Birleşik (Unicode) karakter Kodu (16 bitlik) |

Arithmetic İşleçler

Java beş tane temel aritmetik işlecine (operatör) sahiptir.

Tablo 1.3. Aritmetik işleçler

| Operator | Anlamı | Örnek |
|----------|---------|-------|
| + | Toplama | 3 + 4 |

| | | |
|---|---------|--------|
| - | Çıkarma | 5 - 7 |
| * | Çarpma | 5 * 5 |
| / | Bölme | 14 / 7 |
| % | Modulus | 20 % 7 |

Atama işlemleri

Tablo 1.4. Atama işlemleri

| Deyim | Anlamı |
|--------|-----------|
| x += y | x = x + y |
| x -= y | x = x - y |
| x *= y | x = x * y |
| x /= y | x = x / y |

Karşılaştırma (Comparison)

Java'da mantıksal (boolean) deyimleri oluşturmak için yeterli sayıda karşılaştırma operatörleri vardır. Bütün boolean işlemleri sonunda bir boolean değer (False, True) ortaya çıkar.

Tablo 1.5. Karşılaştırma operatörleri

| Operator | Anlamı | Örnek |
|----------|---------------------|--------|
| == | Eşit | x == 3 |
| != | Eşit değil | x != 3 |
| < | (Soldaki) küçüktür | x < 3 |
| > | (Soldaki) büyüktür | x > 3 |
| <= | Küçük ya da eşittir | x <= 3 |
| >= | Büyük ya da eşittir | x >= 3 |

İkidelik Operatörler

Tablo 1.6. İkidelik operatörler

| Operator | Anlamı |
|----------|--|
| & | İkidelik VE (bitwise AND) |
| | İkidelik VEYA (Bitwise OR) |
| ^ | İkidelik XOR |
| << | Sola kay (Left shift) |
| >> | Sağa kay (Right shift) |
| >>> | Sağa kay ve sıfırla (Zero fill right shift) |
| ~ | İkidelik tümler (Bitwise complement) |
| <<= | Sola kay ve ata (Left shift assignment (x = x << y)) |
| >>= | Sağa kay ve ata (Right shift assignment (x = x >> y)) |
| >>>= | Sağa kay-sıfırla ve ata (Zero fill right shift assignment (x = x >>> y)) |
| x&=y | Ve ata (AND assignment (x = x & y)) |
| x =y | Veya ata (OR assignment (x = x y)) |
| x^=y | XOR ata (XOR assignment (x = x ^ y)) |

Tablo 1.7. Operatör öncelikleri

| Operator | Açıklamalar |
|----------------------|--|
| . [] () | Deyimleri gruplamak için () parantezleri kullanılır, öncelik iç parantezden başlar; nokta operatörü (.) sınıf ve nesne içindeki değişken ve metotlara erişimi sağlar; köşeli parantezler array tanımlar |
| ++ -- ! ~ instanceof | instanceof operatörü, nesne o sınıftan türetilmişse true değilse false değerini alır |
| new (type)deyim | new operatörü, bir sınıftan nesne yaratır; deyim (tip) içine yazılan tipe dönüşür |
| * / % | Çarpma, bölme, modulus |
| + - | Toplama , çıkarma |
| << >> >>> | İkilik kaymalar |
| < > <= >= | Bağlantılar |
| == != | Eşitlik |
| & | AND |
| ^ | XOR |
| | OR |
| && | Logical AND |
| | Logical OR |
| ? : | if...then...else yerine geçen koşullu yapı |
| = += -= *= /= %= ^= | Atama operatörleri |
| &= = <<= >>= >>>= | Atama operatörleri |