Android Mobil Uygulama Geliştirme Eğitimi | Java

Değişkenler ve Veri Tipleri

Kasım ADALAN
Elektronik ve Haberleşme Mühendisi
Android - IOS Developer and Trainer

Eğitim İçeriği

1. Değişkenler
2. Primitif Tipler
3. Referans Tiplere Giriş
4. Aritmetik Operatörler
5. Tip dönüşümleri
6. Konsol Girdisi

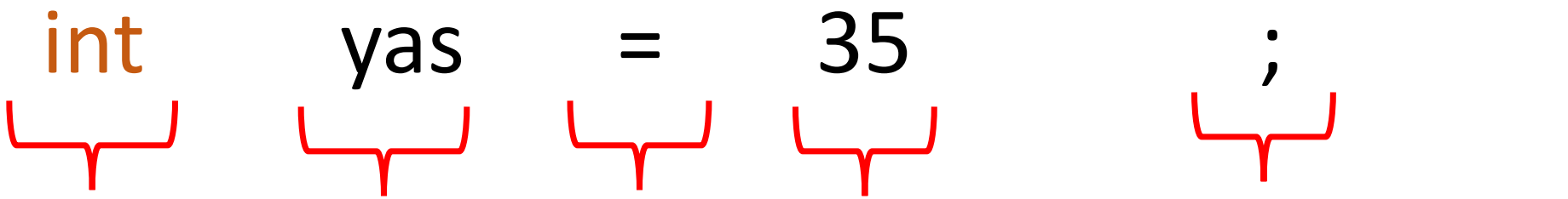
DEĞİŞKENLER

- Modern diller hafızada saklanan değerleri değişkenler ile ifade etmektedir.

- Değişkenler hafızada geçici olarak saklanan değerleri temsil eder .

Not : Değişkenler kalıcı değildir. Programdan çıkıldığında değerler kaybolur. Kalıcı değerler için değişkenlerin değerleri diske yazılmalıdır.

Değişken Oluşturma



yas

Yas 35 değerini tutan bir kutudur, ve yalnızca tam sayı tutabilir

Data Tipleri

Tam Sayılar

Long
Int
Short
Byte

Ondalıklı Sayılar

Double
Float

Metinsel İfadeler

String : Yazılar
Char : Harfler

Mantıksal İfadeler

Boolean : True veya False

Literals – Değerlerin Yazılma Kuralları

- Literals değişkenler için kullanılan değerlerin nasıl yazılması gerektiğini temsil eder.

```
"Ahmet" //String ( Metinsel İfade )
'a' //Char ( Harfsel İfade )
18 // Tam sayı
1.78 // Double ( Ondalıklı Sayı )
1.78f // Float ( Ondalıklı Sayı )
```

Değişkenlere isim verme kuralları

- Case sensitive'dir.Büyük küçük harf farkı vardır.
- Rakamla başlayamaz.
- Boşluk içeremez.
- Türkçe karakter içeremez.
- @, $, ve % değişken içerisinde kullanılmaz.
- Sınıf isimleri büyük harf ile başlar.

Bazı örnekler

```
Azad        zara   abc    move_name    a_123
myname50    _temp  j      a23b9        retVal
```

Örnek

Bir öğrencinin

adını ,yaşını ,boyunu ve adının baş harfininin tutulduğu değişken oluşturunuz.

Örnek

- Bir şirketin ürünlerinin bilgilerinin tutulduğu ürünler tablosunu temsil eden değişkenleri oluşturunuz.

ürün_id	ürün_adi	ürün_adet	ürün_fiyat	ürün_tedarikci
3416	Kol saati	100	149.99	rolex

`System.out.println();` metodu ile Çıktı Alma

- Bu metodları kodlama yaparken sıkça kullanırız.
- Kodlama yaparken kodların çalışma sonuçlarını bu metod ile takip edebiliriz.
- print() yan yana , println() alt alta yazmak için kullanılır.

```
System.out.print("Deneme1");
System.out.println("Deneme1");
System.out.print("Deneme1");
```

```
Deneme1Deneme1
Deneme1
```

Kısayol

Değişkenleri Yazdırma

String ifade içine + ifadesi kullanılarak çıktıya değişken eklenebilir.

```
String ad = "Ahmet";
int yil = 10;

System.out.println(ad+" Bursada "+yil+" yıldır yaşamaktadır");
```

```
Ahmet Bursada 10 yıldır yaşamaktadır
```

```
int a = 10;
int b = 20;

System.out.println(a+" ve "+b+" nin toplamı : "+(a+b)+" dur");
```

```
10 ve 20 nin toplamı : 30 dur
```

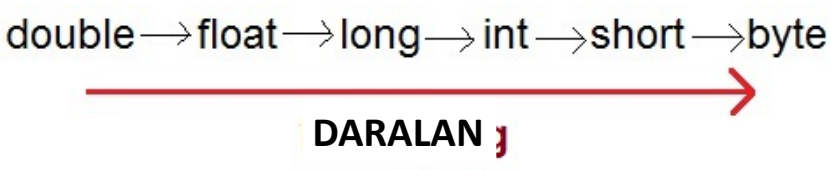
Tür Dönüşümü

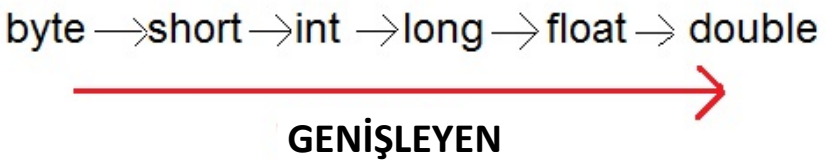
1. Sayısal ifadelerin kendi aralarında dönüşümü.
   - Explicit (açıktan dönüşüm)
   - Unexplicit (açıktan olmayan dönüşüm)

2. Sayısal ifadeler ile String'ler arasındaki dönüşümler.
   - Sayısal ifadeden String'e dönüşüm.
   - String'den Sayısal ifadeye dönüşüm..

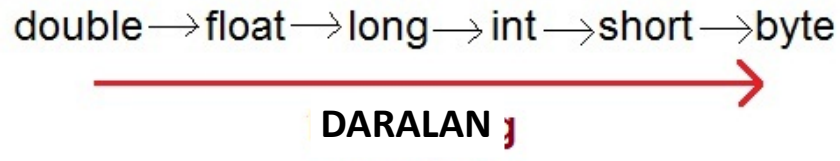
SayısalDönüşümler

- Reel sayılar (float, double) tam sayılara dönüşürken ondalık kısmı gider.
- **NOT :** Genelde küçük veri tipleri büyük veri tiplerine çevrilirken sorun yok ama büyük veri tipinden küçük veri tipine çevrilirken **bilgi kaybı** olabilir.





Explicit (açıktan değişim)



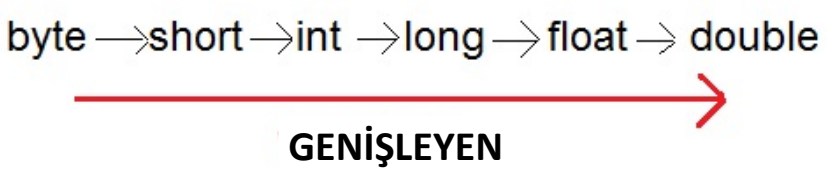

```
double d = 100.04 ;
long l = (long)d;
int i = (int)l;

System.out.println("Double deger : " + d);
System.out.println("Long   deger : " + l);
System.out.println("Int    deger : " + i);
```

Çıktı

```
Double deger : 100.04
Long   deger : 100
Int    deger : 100
```

Unexplicit(açıktan olmayan değişim)

```
int i = 100 ;
long l = i;
double d = l;

System.out.println("Int    deger : " + i);
System.out.println("Long   deger : " + l);
System.out.println("Double deger : " + d);
```



Çıktı

```
Int    deger : 100
Long   deger : 100
Double deger : 100.0
```

Sayısaldan String'e dönüşüm

```
int sayi = 56 ;

String kelime1 = String.valueOf(sayi);
String kelime2 = Integer.toString(sayi);
String kelime3 = sayi + "" ;
```

String'den Sayısal ifadeye dönüşüm

```
String kelime  = "56" ;

int sayi1 = Integer.parseInt(kelime);
int sayi2 = Integer.valueOf(kelime);
```

```
String kelime  = "56" ;

float sayi1 = Float.parseFloat(kelime);
float sayi2 = Float.valueOf(kelime);

float sayi3 = 12.5f;

String kelime1 = String.valueOf(sayi3);
String kelime2 = Float.toString(sayi3);
String kelime3 = sayi3 + "";
```

Açıklama

- Integer.valueOf() ile Integer.parseInt() metotları arasında temel fark valueOf() metodu Integer tipinde bir **nesne**,parseInt() metodu ise int tipinde bir **veri tipi dönderir**.Ve parseInt() metodu temel veri tipi döndürdügü için daha hızlı çalışır.
- (Aynı durum Byte.valueOf()-Byte.parseByte(),Short.valueOf()-Short.parseShort(),Long.valueOf()-Long.parseLong(),Float.valueOf()-Float.parseFloat(),Double.valueOf()-Doble.parseDouble() için de geçerlidir)

Android Kullanım Alanı

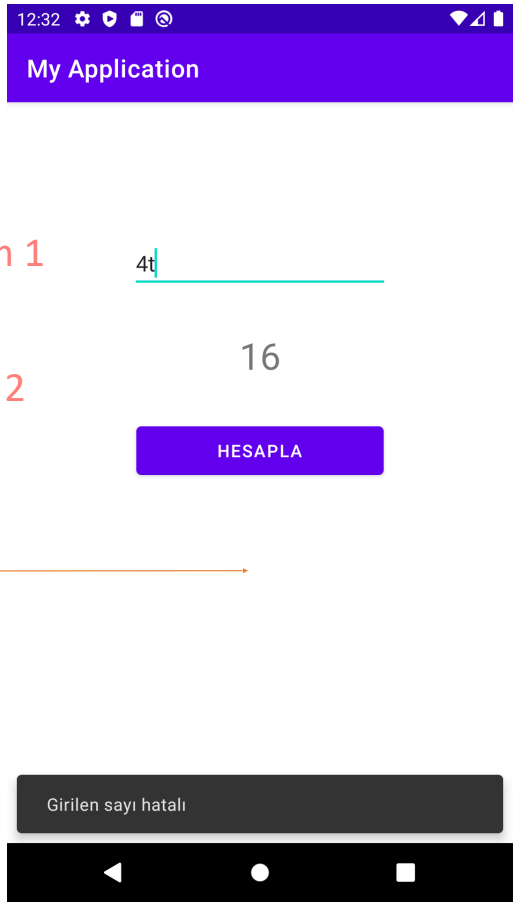
1. Durum


```
buttonHesapla.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String gelenVeri = editTextGirdi.getText().toString();

        try {
            int sayi = Integer.parseInt(gelenVeri);//String to Int
            int hesap = sayi * sayi;
            String sonuc = String.valueOf(hesap); //String to Int
            textViewCikti.setText(sonuc);

        }catch (Exception e){
            Snackbar.make(buttonHesapla
                    , text: "Girilen sayı hatalı",Snackbar.LENGTH_SHORT).show();
        }

    }
});
```

Dönüşüm 1

Dönüşüm 2

2. Durum



Teşekkürler...

kasım-adalan

kasimadalan@gmail.com

kasimadalan