

NESNE YÖNELİMLİ PROGRAMLAMA(Object Oriented Programming/OOP)

Öğr. Gör. Celil ÖZTÜRK

Marmara Üniversitesi

Teknik Bilimler Meslek Yüksekokulu

İçerik

- ✓ Veri Dönüşümü
- ✓ Kullanıcı Girdileri
- ✓ Java Uygulamaları
- ✓ Import, Package kavramları
- ✓ Nesne Oluşturma ve Nesne Referansları
- ✓ String Sınıfı ve Metodları

Veri Dönüşümü

- Bazı zamanlar veri bir tipten diğerine dönüştürülmeye elverişli olur. Örneğin, özel bir durumda bir tam sayının ondalıklı sayı gibi davranması istenebilir.
- Bu dönüşümler bir değişkenin tipini ya da içinde depolanan verinin değerini değiştirmez, sadece bir hesaplamanın parçası olarak bir değeri dönüştürür.

Veri Dönüşümü

- Veri kaybını önlemek için dönüşümler dikkatle yürütülmelidir.
- **Genişleme dönüşümleri** en güvenlisidir çünkü küçük veriden büyük olana doğru gitmeye eğilimlidirler.
(örneğin, short'tan int'e doğru)
- **Daralma dönüşümleri** veri kaybedebilirler çünkü büyük veriden küçük olana doğru gitmeye eğilimlidirler.
(örneğin, int'ten short'a doğru)
- Java'da veri dönüşümünün üç yolu:
 - Atama dönüşümü
 - Yükselme (Promotion)
 - Kalıplama (Casting)

Veri Dönüşümü

Atama dönüşümü bir tipteki değerin başka tipteki bir değişkene atanması ile meydana gelir.

Örnek:

Eğer **para** bir **float** değişken ve **ytl** bir **int** değişken ise, şu ifade ytl içindeki değeri float'a dönüştürür.

`para = ytl`

- Sadece **genişleme dönüşümleri** atama yoluyla meydana gelebilir.
- ytl'nin değerinin ya da tipinin değişmediğine dikkat ediniz.

Veri Dönüşümü

Yükselme, ifadelerdeki operatörler işlenenlerini dönüştürdüğünde kendiliğinden meydana gelir.

Örnek:

Eğer **toplam float** ise ve **sayi int** ise,

sayi şu işlemin gerçekleştirilmesi için ondalıklı bir değere dönüştürülür:

sonuc = toplam / sayi;

Veri Dönüşümü

Kalıplama(Casting)

- Kalıplama, dönüşümde en güçlü ve en tehlikeli tekniktir.
- Bir değerin açıkça dönüştürülmesi ile hem genişleme hem daralma dönüşümleri başarılı olabilir.
- Dönüştürmek için, dönüştürülecek değerin önüne parantezler içinde dönüştürülecek tip yazılır.

Örnek:

`toplam` ve `sayi` tamsayılarsa, fakat bölme işlemi yapılırken ondalıklı bir sonuç isteniliyor ise `toplam`'ı kalıplayabiliriz (cast):

`sonuc = (float) toplam / sayi;`

Etkileşimli(Interactive) Programlar

- Programlar genellikle işletecekleri girdilere ihtiyaç duyarlar.
- **Scanner** sınıfı çeşitli tiplerde girdileri okumaya elverişli metodlar sağlar.
- Bir **Scanner** nesnesi kullanıcının klavyeden girdiği değerleri de içeren çeşitli kaynaklardan girdi okuyabilecek şekilde hazırlanabilir.
- Klavye girdileri System.in nesnesi tarafından temsil edilir.

Girdi Okuma

Aşağıdaki satır klavyeden okuma yapacak bir Scanner nesne oluşturur:

- `Scanner oku = new Scanner(System.in);`
- `new` operatörü Scanner nesnesini oluşturur
- Bir defa oluşturulduktan sonra, Scanner nesnesi aşağıdaki gibi çeşitli girdi metodlarını çağırmak için kullanılabilir:

```
cevap = oku.nextLine();
```

Scanner Sınıfı

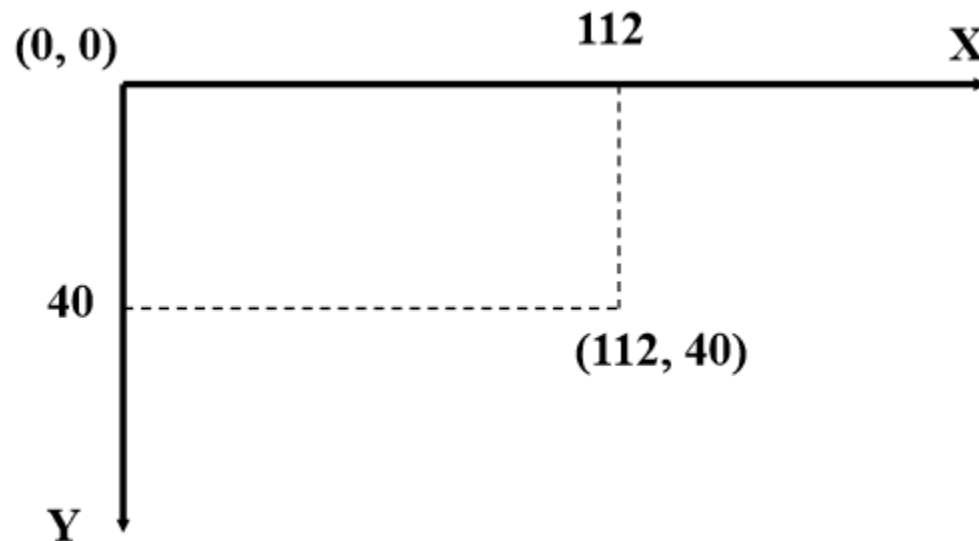
- Scanner sınıfı **java.util** sınıf kütüphanesinin bir parçasıdır ve kullanılabilmesi için programın içine aktarılması (import) gerekir .
- **nextLine** metodu bulunduğu satırın sonuna kadar tüm girdileri okur.
- Aksi belirtilmediği müddetçe, alfabe dışı karakterler (white space) girdinin elemanlarını (simgelerini) ayırmak için kullanılır.
- Alfabe dışı karakterler boşluk karakterlerini, tabları ve yeni satır karakterlerini içerir.
- **Scanner** sınıfının **next** metodu bir sonraki girdi simgesini okur ve bir string olarak geri çevirir.
- **nextInt** ve **nextDouble** gibi metodlar belirli tipteki verileri okurlar.

Grafiğe Giriş

- Bir görüntü ya da çizimin bilgisayarda depolanabilmesi için sayısallaştırılması gereklidir.
- Bir görüntü piksellerden (görüntü elemanı) oluşur ve her bir piksel ayrı ayrı depo edilir.
- Bir görüntüyü oluşturmak için kullanılan piksellerin sayısına **görüntü çözünürlüğü** denir.
- Bir ekran tarafından gösterilebilen piksel sayısına **ekran çözünürlüğü** denir.

KOORDİNAT SİSTEMLERİ

- Her piksel iki boyutlu bir koordinat sistemi kullanılarak tanımlanabilir.
- Java programında bir pikselden bahsederken, orijini üst sol köşede olan bir koordinat sistemi ile ifade edilir.



Renk Sınıfı

- Siyah beyaz bir görüntü her bir piksel için bir bit kullanılarak depolanabilir. (0=beyaz ve 1=siyah)
- Renkli bir görüntü daha fazla bilgi gerektirir; renkleri belirtmek için çeşitli teknikler vardır.

Örneğin, her renk üç ana rengin karışımı olarak belirtilebilir.

Kırmızı, Yeşil ve Mavi

- Her renk 0 ile 255 arasından (toplu halde RGB değeri olarak adlandırılan) 3 sayı ile belirtilebilir.

Renk Sınıfı

- Bir Java programındaki bir renk **Color** sınıfından oluşturulmuş bir nesne ile temsil edilir.
- **Color** sınıfı aşağıdakilerin de dahil olduğu çeşitli önceden tanımlı renkler de içerir:

<u>Nesne</u>	<u>RGB Değeri</u>
<code>Color.black</code>	<code>0, 0, 0</code>
<code>Color.blue</code>	<code>0, 0, 255</code>
<code>Color.cyan</code>	<code>0, 255, 255</code>
<code>Color.orange</code>	<code>255, 200, 0</code>
<code>Color.white</code>	<code>255, 255, 255</code>
<code>Color.yellow</code>	<code>255, 255, 0</code>

APPLET

- Bir Java uygulaması **main metodu** olan bağımsız bir programdır. Ancak applet'ler main metodu içermezler.
- Java applet'i web üzerinde taşınmak ve bir web tarayıcısı kullanılarak çalıştırılmak üzere tasarlanmış bir programdır.
- Bir applet , Java Software Development Kit'te bulunan appletviewer tool kullanılarak da çalıştırılabilir.
- Bunun yerine, belirli amaçlara hizmet eden çeşitli özel metodlar vardır.

APPLET

- Bir applet tanımlayan sınıf , Applet sınıfını genişletir (extend).
- İleride daha ayrıntılı incelenecek olan kalıtımın (inheritance) kullanımından faydalanır.
- Bir applet, appletin makine kodu dosyasına işaret eden bir tag kullanılarak HTML dosyasının içine gömülür.
- Programın makine kodu versiyonu web üzerinden iletilir ve tarayıcının bir parçası olan bir Java yorumlayıcısı tarafından çalıştırılır.

HTML Applet Tag

- HTML Applet Tag

```
<html>
```

```
<title>The Ornek X</title>
```

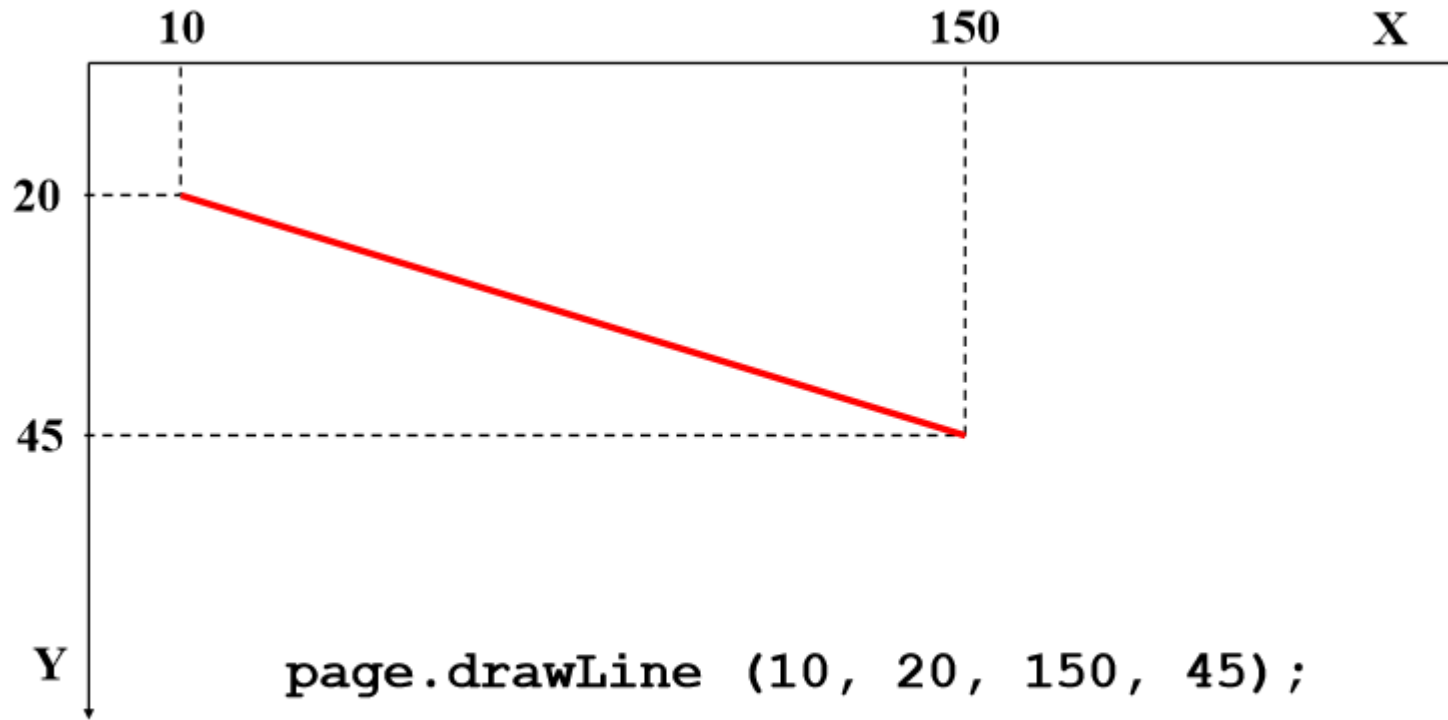
```
<applet code="MyApplet.class" width=300 height=250 > </applet>
```

```
</html>
```

Şekil Çizimi

- Graphics sınıfının şekiller çizen metodları daha ayrıntılı incelenek olursa:
- Bir şekil hangi metodun çağırıldığına bağlı olarak, doldurulmuş ya da doldurulmamış olabilir.
- Metod parametreleri koordinatları ve büyüklükleri belirler.
- Oval gibi eğrileri olan şekiller, genellikle şekli sınırlayan dikdörtgenin belirlenmesi ile çizilir.
- Bir yay bir ovalin bir bölümü olarak düşünülebilir.

Şekil Çizimi-Çizgi

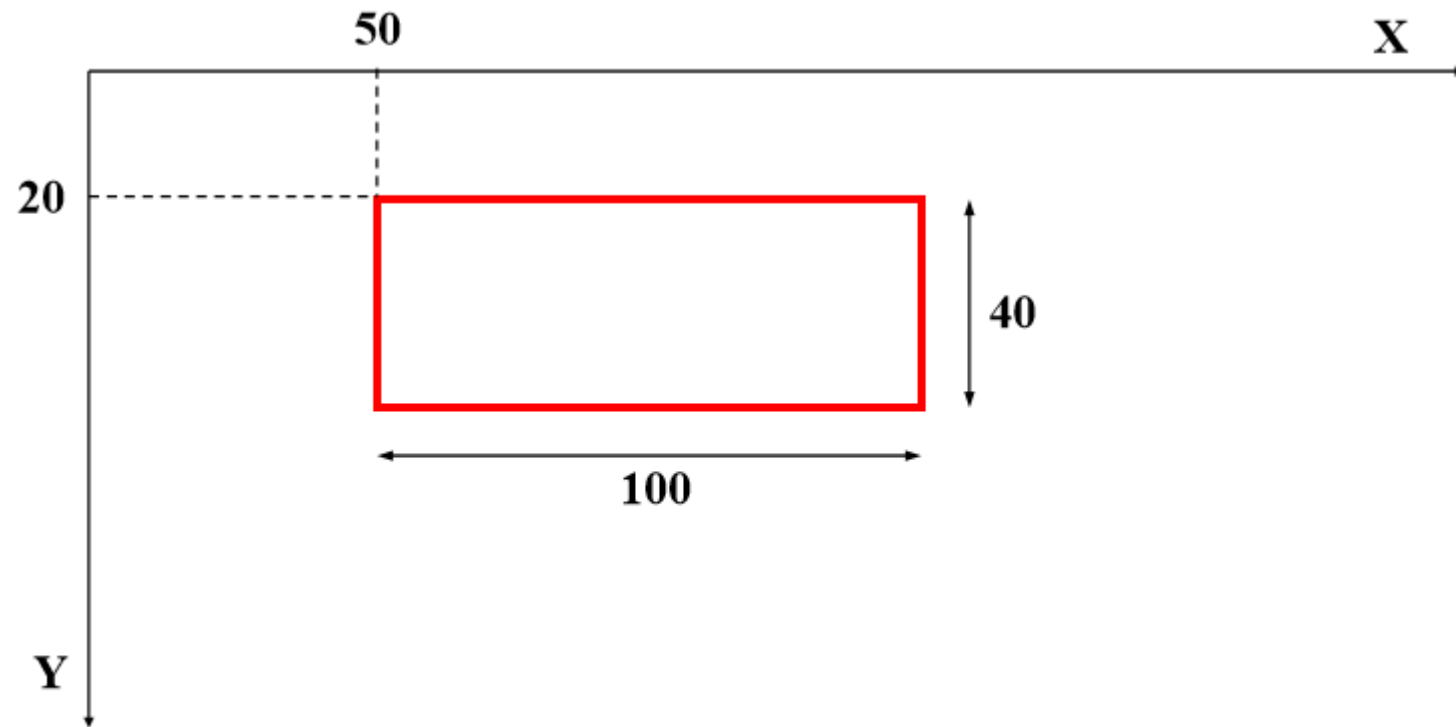


```
page.drawLine (10, 20, 150, 45) ;
```

veya

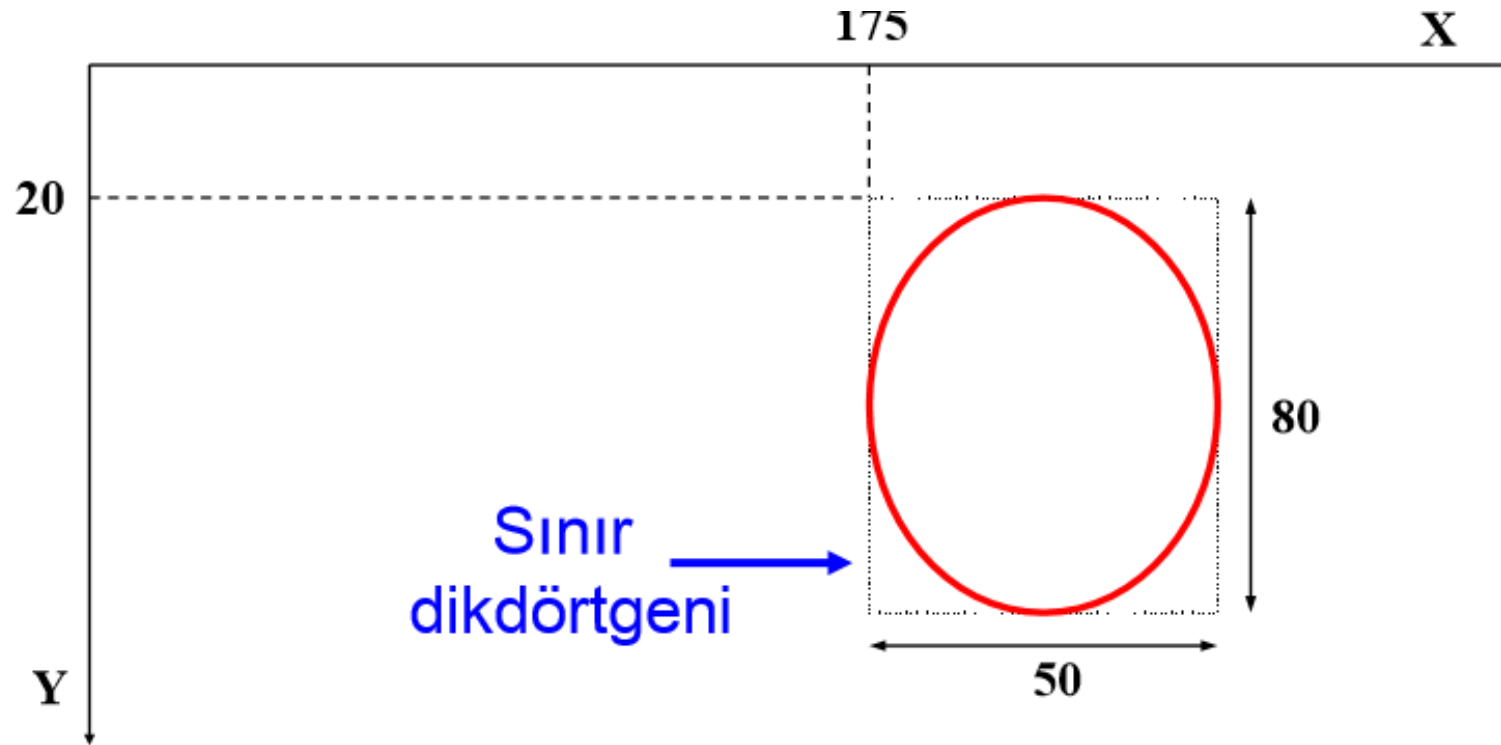
```
page.drawLine (150, 45, 10, 20) ;
```

Şekil Çizimi-Dikdörtgen



```
page.drawRect (50, 20, 100, 40) ;
```

Şekil Çizimi-Oval



```
page.drawOval (175, 20, 50, 80) ;
```

Paket(Package) Kavramı

- Java dilinde modülerliği sağlayan en önemli kavramlardan biri paket kavramıdır.
- Birbiriyle işlev veya kullanım açısından ilişkili sınıflar (class) paket (package) adı verilen bir birim altında birleştirilir.
- Bir sınıfın (class) ait olduğu paketi belirtmek için
- `java.paketin_adi.sınıfın_adi` anahtar kelimesi kullanılır.
- Örnek: `java.util.Scanner`

Paket(Package) Kavramı

- Standart Java sınıf kütüphanesinin sınıfları paketler halinde düzenlenmiştir.
- Standart sınıf kütüphanesindeki paketlerden bazıları:

Paket

java.lang
java.applet
java.awt
javax.swing
java.net
java.util
javax.xml.parsers

Amaç

Genel destek
Web için küçük uygulamalar oluşturmak
Grafikler ve grafiksel kullanıcı arayüzleri
İlave grafik yetenekleri
Ağ haberleşmesi
Yardımcı yazılımlar
XML dosya işleme

Özel Paket Tanımlama

- **Package** **paket_ismi** söz dizimi ile paket tanımlanabilir.
- Tanımlanan paketler içindeki sınıflar bu paketin import edilmesi ile kullanılabilir.

```
package packageornek;  
import erisimpaketi.ErisimSinifi;  
// import erisimpaketi.*; «ile tüm sınıflar eklenebilir.»  
  
public class PackageOrnek {  
  
    public static void main(String[] args) {  
        ErisimSinifi nesne = new ErisimSinifi();  
        nesne.yazdir();  
    }  
}
```



```
package erisimpaketi;  
  
public class ErisimSinifi {  
    public void yazdir()  
    {  
        System.out.println("Erisim  
sınıfına erisim sağlandı !");  
    }  
}
```


IMPORT

- Bir sınıfın, kendisiyle aynı pakette olmayan başka bir sınıfı kullanabilmesi için, sınıf (class) tanımından önce import edilmesi gerekir.
- Bu sadece kullanılan sınıfların tam olarak hangi pakette olduğunu söylemek anlamına gelir. import kelimesi, kaynak kod o paketi kullanmadan önce yazılmalıdır.

import java.paket_adı.sınıf_adı; şeklinde kullanılır.

Örnek: import javax.swing.JApplet;

IMPORT

- Java aynı dizin içindeki sınıfları, bir paketin içindeymiş gibi görür. import deyimi ile birlikte * joker karakteri kullanılabilir.

Örnek : `import java.util.*;`

Anlamı : “java.util paketindeki tüm sınıfları çağır”

Not: * joker karakterini ilgili bir paket için kullanabilirsiniz fakat java.* şeklinde Java’nın altındaki tüm paketlere erişemezsiniz.

Nesne Oluşturma

- Bir değişken bir ilkel tip ya da bir nesneye ait bir referans tutabilir. •
- Bir sınıf ismi bir nesne referansı değişkenini bildirmek için bir tip gibi kullanılabilir.
 - `String baslik;`
- Bu bildirim ile bir nesne oluşturulmaz.
- Bir nesne referansı değişkeni, bir nesnenin adresini tutar.
- Nesnenin kendisi ayrıca oluşturulmalıdır.

Nesne Oluşturma

Genellikle, bir nesne oluşturmak için **new** operatörünü kullanırız

```
baslik = new String ("Java Yazilim Cozumleri");
```



Bu, nesneyi kuran özel bir metod olan
String yapıcısını çağırır

Bir nesnenin oluşturulmasına örnekleme
(*instantiation*) denir

Bir nesne belirli bir sınıfın örneğidir

Metotları Çağırma

- Bir nesne örneklendiğinde, nokta operatörü ile metodları çağırılır.
sayi = baslik.length();
- Bir metod, bir atama ya da ifadede kullanılacak bir değer geri çevirebilir.
- Bir metodun çağırılması bir nesneden bir vazifeyi yerine getirmesini istemek gibi düşünülebilir.

REFERANSLAR

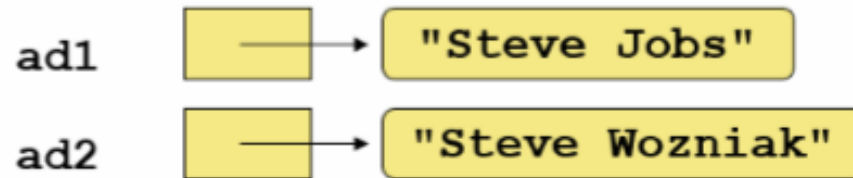
- Bir ilkel değişkenin değeri kendisini içerdiğine fakat bir nesne değişkenin nesnenin adresini içerdiğine dikkat edin.
- Bir nesne referansı, nesnenin bulunduğu yere bir işaretçi olarak düşünülebilir.
- Gelişigüzel adreslerle uğraşmak yerine, çoğu kez referansı grafiksel olarak gösteririz.



Referans Atama

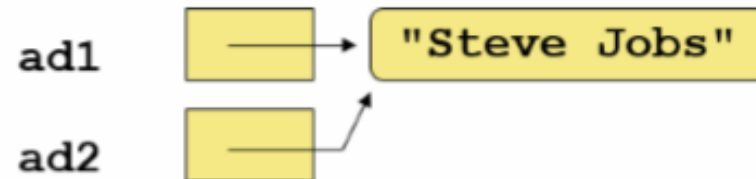
- Nesne referansları için, atama adresi kopyalar:

Önce:

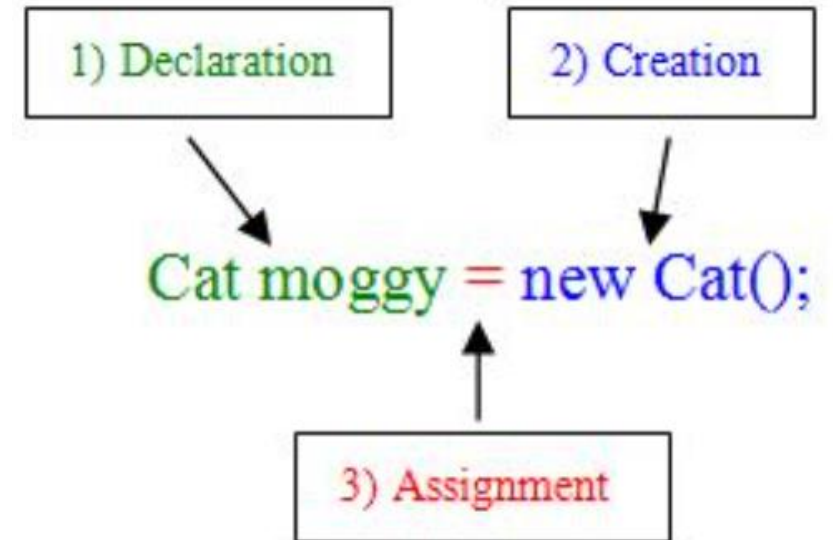
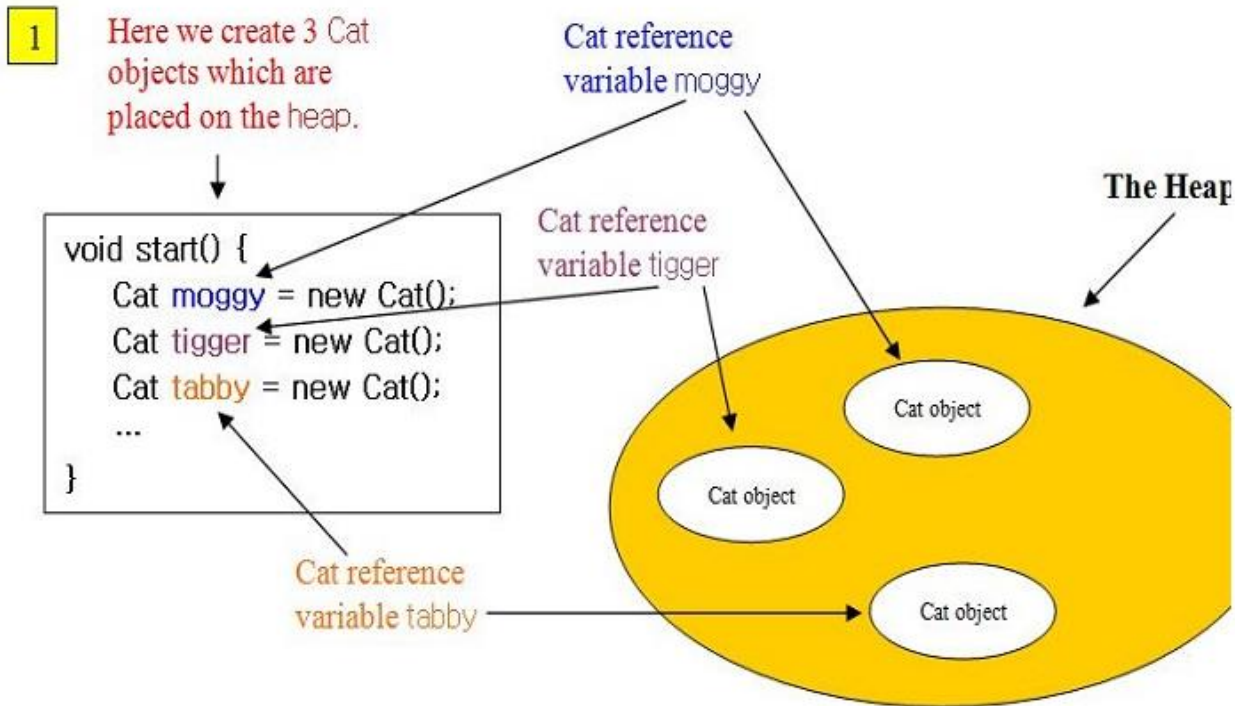


```
ad2 = ad1;
```

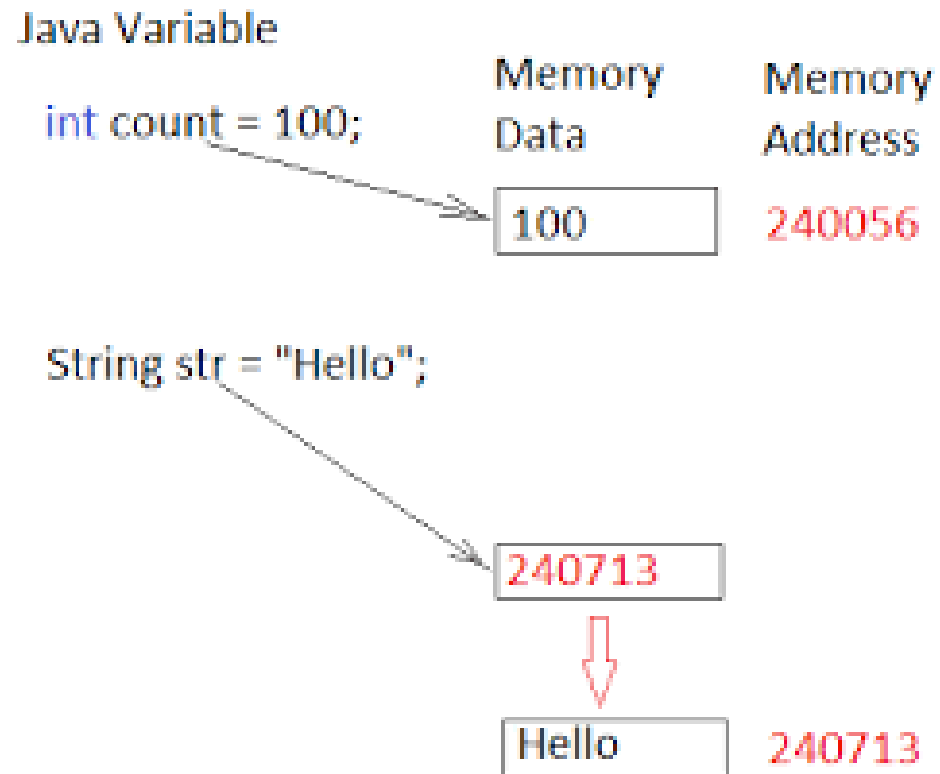
Sonra:



Referans Atama



Referans Atama



ARTIKLARIN TOPLANMASI (Garbage Collection)

- Bir nesnenin hiç geçerli referansı kalmadığında, program için artık erişilebilir olamaz.
- Nesne işe yaramazdır, bu yüzden atık olarak adlandırılır.
- Java, bir nesnenin belleğini sisteme geri çevirerek, otomatik atık toplamaı gerçekleştirir.
- Diğer dillerde, programcı atık toplama işleminden sorumludur.

String Sınıfı

- Stringler çok yaygın olduğundan, bir String nesnesi oluşturmak için new operatörünü kullanılması zorunlu değildir.

`baslik = "Java Yazılım Çözümleri";`

- Bu sadece stringler için işleyen bir söz dizimidir.
- Her bir karakter değişkeni (çift tırnaklar arasına alınmış) bir String nesnesi belirtir.

String Metodları

- Bir String nesnesi oluşturulduğunda, ne değeri ne de uzunluğu değiştirilemez.
- Böylece String sınıfına ait bir nesnenin sabit olduğunu söylemiş oluyoruz.
- Bununla birlikte, String sınıfının çeşitli metodları, asıllarının değiştirilmiş versiyonları olan String nesnelerini döndürürler.

String Sınıfı

String Metodları

`String s=new String()`

- Bu ifade içinde karakter olmayan bir „s“ String'i oluşturur.
- `String(char chars[])`
- Bir Array'i karakter elemanlarını başlangıç değer olarak alan bir String oluşturulabilir.

String Sınıfı

- String Uzunluğu: `Int Length()`

Örnek:

```
class deneme {  
    public static void main(String args[])  
    {  
        char uzun[]={'a','b','c'};  
        String s=new String(uzun);  
        System.out.println(s.length( ));  
    } }
```

String Sınıfı

String İndeksleri

- Bir stringin içinden belirli bir karaktere başvurmak bazen yararlıdır.
- Bu karakterin sayısal indeksi belirtilerek yapılabilir.
- Her string'de indeksler sıfırdan başlar.
- “Selam” stringinde ‘S’ karakteri 0. indekste ve ‘m’ karakteri de 4. indekstedir.

String Sınıfı

Stringlerin araştırılması

- `indexOf()`: karakter veya substring'in string içinde geçen ilk konumunu verir.
- `lastIndexOf()`: karakter veya substring'in string içinde geçen son konumunu verir.
- Araştırılmaya nereden hangi konumdan başlanacağı bilgisi şu form ile verilebilir.

`Int indexOf(string s, int baslangic_indeksi)`

`Int lastIndexOf(string s, int baslangic_indeksi)`

String Sınıfı

- Örnek Kod:

```
import java.io.*;
public class Test {
    public static void main(String args[]) {
        String Str = new String("Welcome to Tutorialspoint.com");
        String SubStr1 = new String("Tutorials");
        String SubStr2 = new String("Sutorials");
        System.out.print("Found Index : " );
        System.out.println(Str.indexOf( 'o' ));
        System.out.print("Found Index : " );
        System.out.println(Str.indexOf( 'o', 5 )); // 5, Aramanın başlatılacağı konumu belirtir.
        System.out.print("Found Index : " );
        System.out.println( Str.indexOf( SubStr1 ));
        System.out.print("Found Index : " );
        System.out.println( Str.indexOf( SubStr1, 15 )); //Bulunmadığında -1 döner. !
        System.out.print("Found Index : " );
        System.out.println(Str.indexOf( SubStr2 ));
    } }
```

ÇIKTI

```
Found Index :4
Found Index :9
Found Index :11
Found Index :-1
Found Index :-1
```

****STRING METOTLARI-
JAVA Kodları !!

Random Sınıfı

- Random sınıfı `java.util` paketinin bir parçasıdır.
- Yalancı rasgele sayılar üreten metodlar sağlar.
- Bir Random nesnesi, görünüşte rastgele değerlerden oluşan bir akım üretmek için bir çekirdek değere dayanan karmaşık hesaplamalar yapar.

****Random Uygulama Örnek Kod !!

Math Sınıfı

Math sınıfı `java.lang` paketinin bir parçasıdır.

Math sınıfı çeşitli matematiksel işlevler yerine getiren metodlar içerir.

Şunları içerirler:

- ☐ Mutlak değer
- ☐ Karekök
- ☐ Üst alma
- ☐ Trigonometrik fonksiyonlar

Math Sınıfı

- Math sınıfının metodları statik metodlardır (sınıf metodları olarak da adlandırılır) .
- Statik metodlar sınıf ismi ile çağırılabilirler – Math sınıfına ait bir nesneye gerek yoktur.

değer = Math.cos(90) + Math.sqrt(delta);

- Math sınıfında (class) iki tane sabit tanımlanmıştır. Bu sabitler:
- Math.PI=3.14159265358979323846 (pi) sayısına eşittir.
- Math.E=2.7182818284590452354 (e) sayısına eşittir.

Math Sınıfı

Yöntem	Açıklama	Örnek
abs(x)	x sayısının mutlak değerini verir.	Math.abs(-23.7)=23.7
ceil(x)	x sayısını bir üst tamsayıya dönüştürür. Sonuç double türündedir.	Math.ceil(9.2)=10.0 Math.ceil(-9.8)=-9.0
floor(x)	x sayısını bir alt tamsayıya dönüştürür. Sonuç double türündedir.	Math.floor(9.2)=9 Math.floor(-9.8)=-10
round(x)	x sayısını en yakın tamsayıya tamamlar. Sonuç int ya da long türündedir.	Math.round(9.2)=9 Math.round(9.8)=10
rint(x)	x sayısını en yakın tamsayıya tamamlar. Sonuç double türündedir.	Math.rint(9.2)=9.0 Math.rint(9.8)=10.0
cos(x)	x'in radyan cinsinden kosinüsünü hesaplar.	Math.cos(0.0)=1.0
sin(x)	x'in radyan cinsinden sinüsünü hesaplar.	Math.sin(0.0)=0.0
tan(x)	x'in radyan cinsinden tanjantını hesaplar.	Math.tan(1.0)=1.557040724
acos(x)	x'in radyan cinsinden kosinüsünün tersini hesaplar. x, 1'den büyük ise NaN sonucunu üretir.	Math.acos(1.0)=0.0
asin(x)	x'in radyan cinsinden sinüsünün tersini hesaplar. x, 1'den büyük ise NaN sonucunu üretir.	Math.asin(1.0)=1.570796

Math Sınıfı

Yöntem	Açıklama	Örnek
atan(x)	x'in radyan cinsinden arc tanjantını hesaplar.	Math.atan(45)=1.54857776
exp(x)	Doğal logaritma parametresi olan e'nin üstel değerini (e^x) hesaplar.	Math.exp(1.0)=2.7182818284590455
log(x)	e tabanına göre x'in logaritmasını alır.	Math.log(2.7182818284590455)=1.0
max(x,y)	x ve y'nin büyük olanını verir.	Math.max(2.0,4.0)=4.0
min(x,y)	x ve y'nin küçük olanını verir.	Math.min(2.0,4.0)=2.0
pow(x,y)	x'in y'ninci kuvvetini verir.	Math.pow(2,3)=8.0
sqrt(x)	x'in karekökünü verir.	Math.sqrt(9.0)=3.0
random()	0 ile 1 arasında rastgele sayı üretir.	Math.random()
toDegrees(x)	x'i radyandan dereceye çevirir.	Math.toDegrees(1.0)=57.29577951308232
toRadians(x)	x'i dereceden radyana çevirir.	Math.toRadians(57.29577951308232)=1.0

Kaynaklar

- Java ve Java Teknolojileri, *Tevfik KIZILÖREN* – Kodlab Yayınları
- Yrd. Doç. Dr. Erbil Akbay –Marmara Üniversitesi / Object Oriented Programming 1-2 Ders notları
- Yrd. Doç. Dr. Deniz KILINÇ-Celal Bayar Üniversitesi / Nesneye Yönelik Programlama Ders Notları
- Dr. Öğr. Üyesi Zehra Aysun ALTIKARDEŞ – Nesne Yönelimli Programlama Notları
- http://javayaz.com/?page_id=84