BBS515 Nesneye Yönelik Programlama

Ders 7 - Sınıflar2

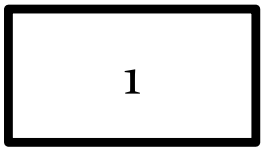
Zümra Kavafoğlu
https://zumrakavafoglu.github.io/

Nesnelerin metotlara parametre olarak verilmesi

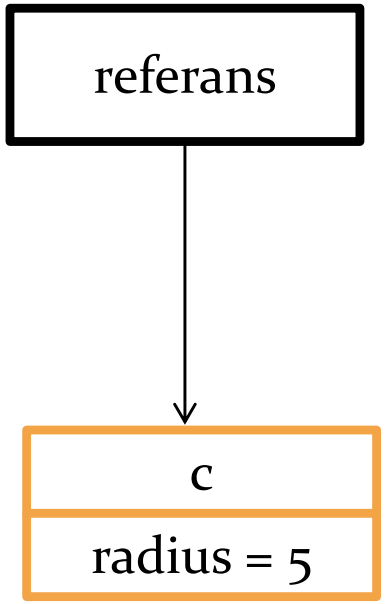
```
public class TestPassingObject {

    public static void main(String[] args) {
        Circle myCircle = new Circle(5.0);
        printCircle(myCircle);
    }

    static void printCircle(Circle c)
    {
        System.out.println("The area of the circle of radius " + c.radius+ "is "+c.findArea());
    }
}
```

Primitif tiplerle Nesneler arasındaki farklar:

- int i=1;
- i

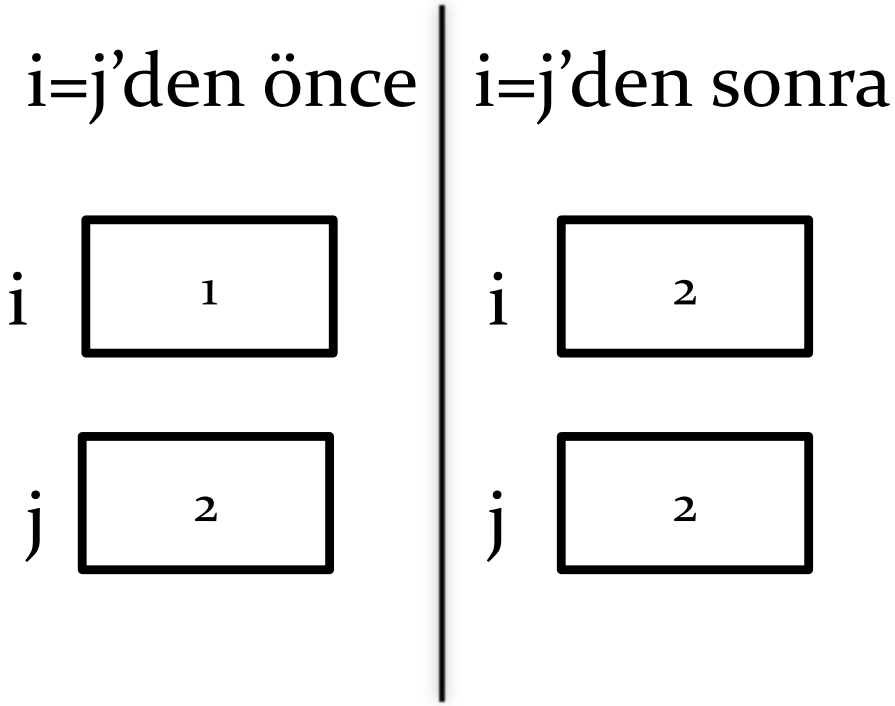


- Circle c = new Circle(5);
- c

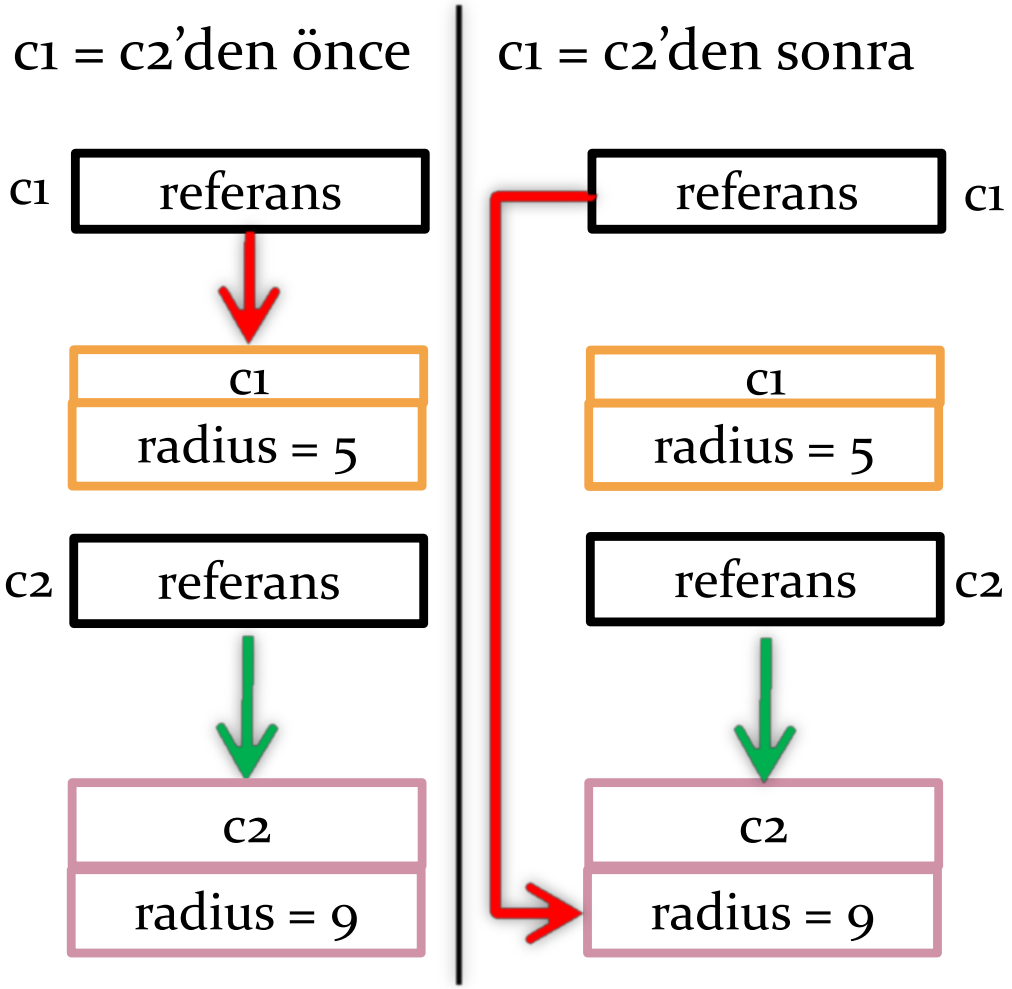


Primitif tiplerle Nesneler arasındaki farklar:

- int i=1;
- int j=2;



- Circle c1 = new Circle(5);
- Circle c2 = new Circle(9);

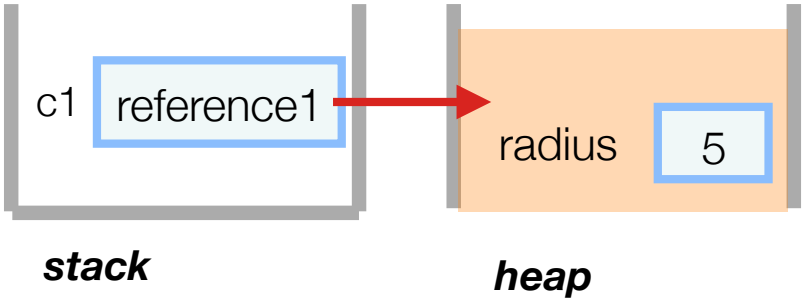


c1

Nesnelerin hafızada saklanması

Bir sınıfın nesnesi oluşturulduğunda nesnenin içerdiği veriler için heap'te alan açılır. stack'de ise heapteki bu alanı gösteren bir referans tutulur.

Circle c = new Circle(5)

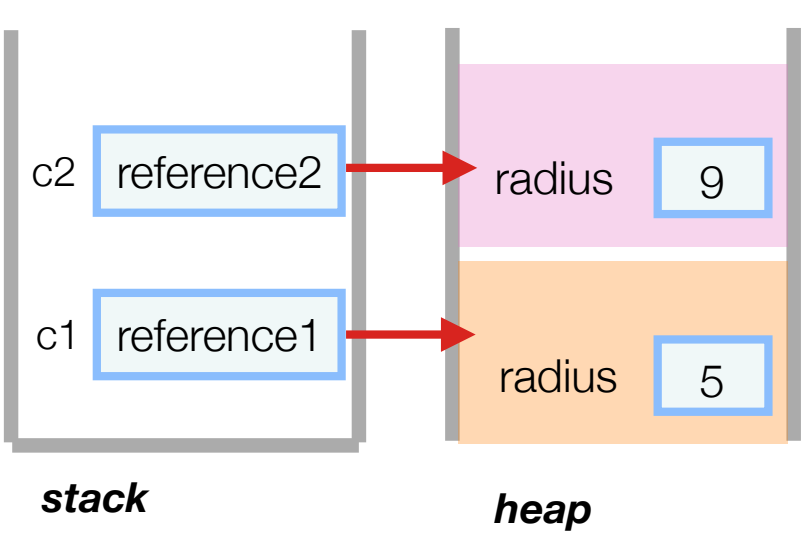


Nesnelerin hafızada saklanması

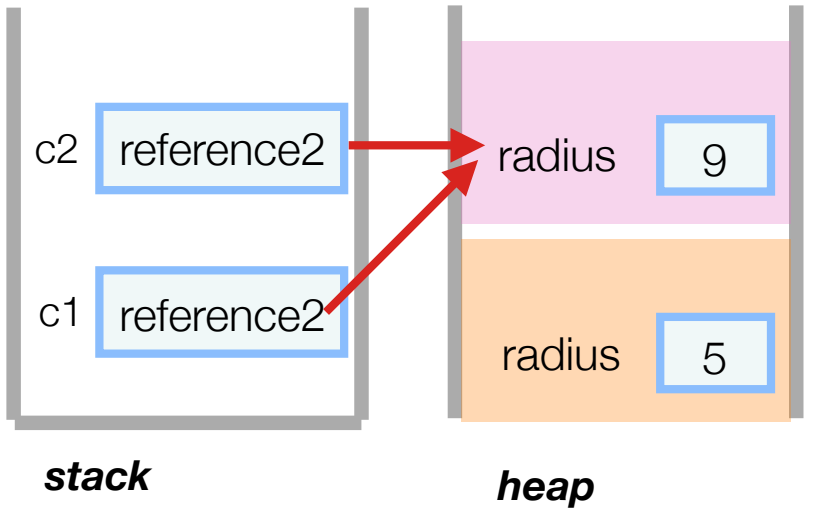
= operatörü nesnenin içeriğini değil, nesneye referans değeri kopyalar.

Circle c1 = new Circle(5)

Circle c2 = new Circle(9)



c1 = c2



Primitif tiplerle Nesneler arasındaki farklar:

Primitif tipler → Pass by value

Nesneler → Pass by reference

```
static void changeValues(Circle c, int x)
{
    x=5;
    c.radius = 5;
}
```

```
Circle myCircle = new Circle(7);
int num = 7;
```

=

```
myCircle.radius = 7.0
num = 7
```

```
changeValues(myCircle, num);
```

```
myCircle.radius = 5.0
num = 7
```

→ myCircle.radius değişti ancak num değişmedi.

GÖRÜNÜRLÜK NİTELEYİCİLERİ (VISIBILITY MODIFIERS)

- Java, verilere, metotlara ve sınıflara ulaşımı kontrol eden çeşitli niteleyiciler barındırır. Bunlardan public, private ve varsayılan niteleyiciyi göreceğiz.
  - public: Sınıf, metot ve verileri tüm programlar onlara erişebilecek biçimde niteler.
  - private: Metot ve verileri sadece tanımlandıkları sınıfta erişilebilecek biçimde niteler.
  - varsayılan niteleyici: Sınıf, metot ya da veriler tanımlanırken başlarına herhangi bir görünürlük niteleyicisi yazılmazsa, aynı paket içindeki herhangi bir sınıf tarafından erişilebilirdirler.

get ve set yöntemleri.

- Çoğu zaman bir sınıf değişkeninin değiştirilmesi ve sınıf değişkenine erişimde, sınıfı tasarlayanın kontrolünün olması gerekir.
- Örneğin her zaman pozitif değerli olması gereken bir sınıf değişkeni, public ile tanımlanıp direkt değiştirilebilir veya erişilebilir olursa, bu değişkene negatif değerler de atanabilir ve bu sınıfın diğer elemanlarını kötü yönde etkileyebilir.
- Bu sebeple
  - sınıf değişkenleri private olarak tanımlanır.
  - private değişkene ulaşmak için public bir get yöntemi yazılır.
  - private değişkeni değiştirmek için public bir set yöntemi yazılır.

```java
public class Circle {

    private double radius;

    public Circle() { radius = 1.0; }

    public Circle(double r) { setRadius(r); }

    public double getRadius() { return radius; }

    public void setRadius(double newRadius){
        if(newRadius<=0){
            System.out.println("Warning: Radius must be positive" +
                    ", it's set to default value 1 ");
            radius = 1;
        }else{
            radius = newRadius;
        }
    }

    public double findArea() { return radius*radius*Math.PI; }
}
```

JAVA PAKETLERİ (PACKAGES)

- Java paketleri Java sınıflarını düzenlemek için kullanılan yapılardır.
- Her Java sınıfı aslında bir paketin içinde yer alır.
- Şimdiye kadar yazdığımız sınıflar hep otomatik olarak o anki klasörün(default package)'ın içinde oluşturuluyordu. Ancak sınıflar başka paketler altında da oluşturulabilir.
- Bir sınıf default package dışında bir paketin içinde oluşturulmak isteniyorsa aşağıdaki satır, açıklama ya da boşluk olmayan ilk satıra yazılmalıdır. (eclipse bunu otomatik olarak yapar.)

```
package packagename
```

```java
package packageDiffCircle;

public class Circle {
    private double radius;

    public Circle() { radius = 1.0; }

    public Circle(double r) { setRadius(r); }

    public double getRadius() { return radius; }

    public void setRadius(double newRadius){
        if(newRadius<=0){
            System.out.println("Warning: Radius must be positive, " +
                    "it's set to default value 1 ");
            radius = 1;
        }else{
            radius = newRadius;
        }
    }

    public double findArea() { return radius*radius*Math.PI; }

    public double findPerimeter() { return 2*Math.PI*radius; }
}
```

JAVA PAKETLERİ (PACKAGES)

- Bir sınıftan, başka bir paketin içindeki public bir sınıfa ulaşabilmek için aşağıdaki satır, programın başına yazılmalıdır.

```
import paketadı.sınıfadı
```

- Örneğin myPackage'ın içindeki Circle sınıfını çağırmak için aşağıdaki satır yazılır.

```
import myPackage.Circle
```

- Bir paketin içindeki tüm sınıflara erişmek isteniyorsa aşağıdaki satır yazılır.

```
import paketadı.*
```

```java
import packageDiffCircle.Circle;

public class TestDiffCircle {
    public static void main(String[] args){

        //Create a circle with radius 5.0
        Circle myCircle = new Circle(5.0);

        System.out.println("Radius of circle: "+myCircle.getRadius());
        System.out.println("Area of circle: "+myCircle.findArea());
        System.out.println("Perimeter of circle: "+myCircle.findPerimeter());
    }
}
```