

Repaso referencias (Anexo)



UT4

Circulo y Punto

```
public class Circulo
{
    private static final double PI = 3.14;
    private double radio;
    private Punto centro;

    /**
     * constructor para los objetos de la clase Circulo
     */
    public Circulo(double radio)
    {
        this.radio = radio;
        this.centro = new Punto(0, 0);
    }
}
```

Circulo y Punto

```
/**
 * constructor
 */
public Circulo(double radio, Punto centro)
{
    this.radio = radio;
    this.centro = centro;
    // this.centro = new Punto(centro.getX(), centro.getY());
}
```

```
/**
 * constructor |
 */
public Circulo(Circulo otro)
{
    this.radio = otro.getRadio();
    this.centro = otro.getCentro();
    // this.centro = new Punto(otro.getCentro().getX(),
    //                          otro.getCentro().getY());
}
```

Circulo y Punto

```
/**
 *
 * accesor para el radio del círculo
 */
public double getRadio()
{
    return radio;
}
```

```
/**
 *
 * accesor para el centro del círculo
 */
public Punto getCentro()
{
    return centro;
}
```

Circulo y Punto

```
public class TestCirculo
{
    public static void main(String[] args){
        Circulo c1 = new Circulo(17, new Punto(4, 6));
        System.out.println(c1.toString());

        Circulo c2 = new Circulo(c1);
        System.out.println(c2.toString());
    }
}
```

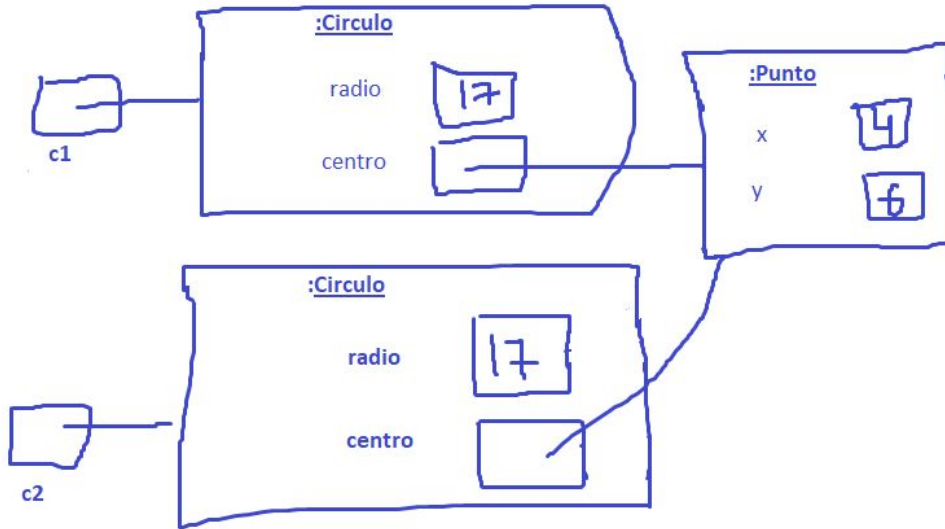
Circulo y Punto - Diagrama de objetos

Circulo c1 = new Circulo(17, new Punto(4, 6));



Circulo y Punto - Diagrama de objetos

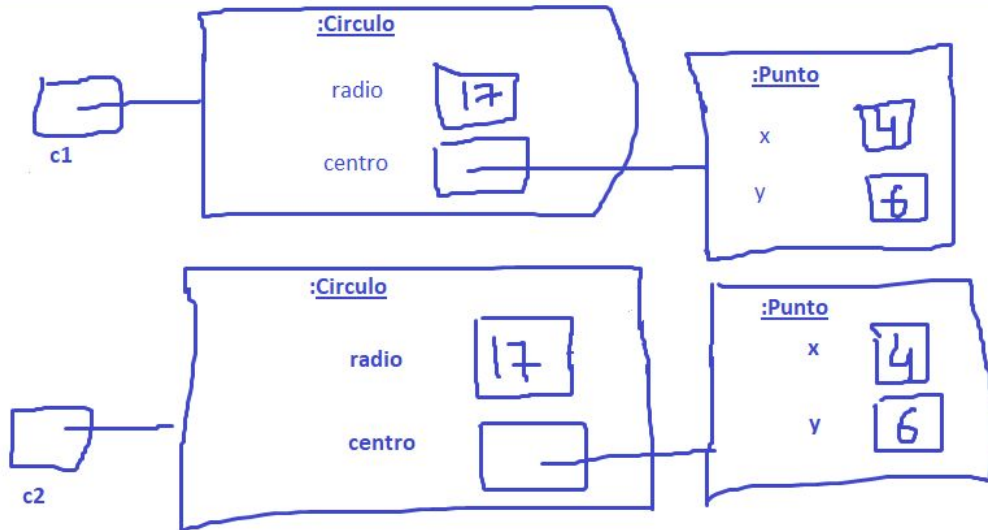
Circulo c2 = new Circulo(c1);



```
public Circulo(Circulo otro)
{
    this.radio = otro.getRadio();
    this.centro = otro.getCentro();
}
```

Circulo y Punto - Diagrama de objetos

Circulo c2 = new Circulo(c1);



```
public Circulo(Circulo otro)
{
    this.radio = otro.getRadio();
    this.centro =
        new Punto(otro.getCentro().getX(),
                  otro.getCentro().getY());
}
```


Dimension Paquete

(Entrega1 UT4 21-22)

Dimension y Paquete - Diagrama de objetos

En clase Dimension

```
/**
 * Devuelve una copia exacta al objeto actual
 */
public Dimension obtenerCopia() {
    return new Dimension(this.alto,
        this.ancho, this.largo);
}
```

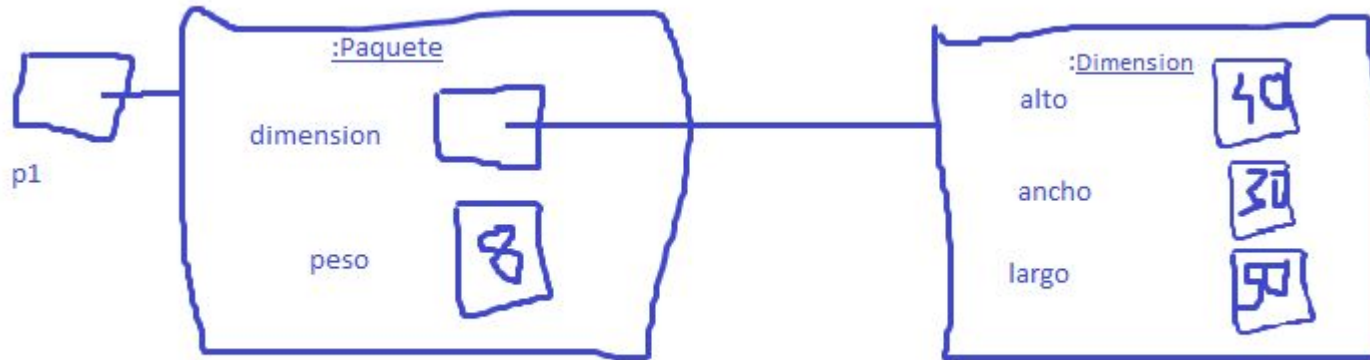
Dimension y Paquete - Diagrama de objetos

En clase Paquete

```
/**  
 * Devuelve una copia exacta al objeto actual  
 *  
 */  
public Paquete obtenerCopia() {  
    return new Paquete(this.dimension, this.peso);  
}
```

Dimension y Paquete - Diagrama de objetos

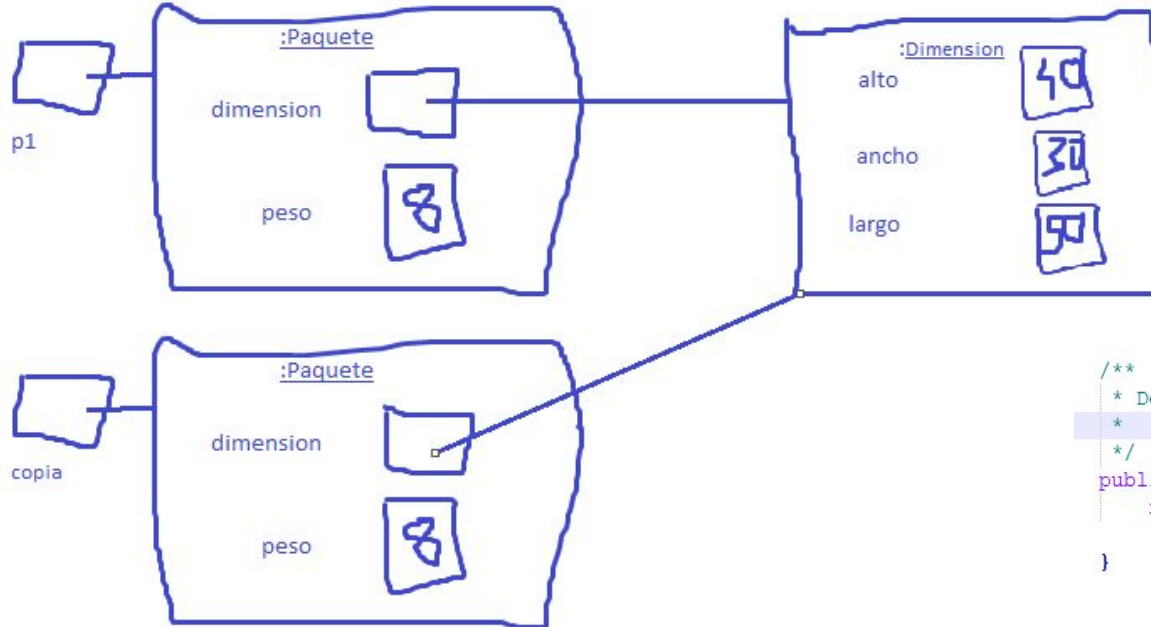
Paquete p1 = new Paquete();



Dimension y Paquete - Diagrama de objetos

Paquete p1 = new Paquete();

Paquete copia = p1.obtenerCopia();



```
/**  
 * Devuelve una copia exacta al objeto actual  
 */  
public Paquete obtenerCopia() {  
    return new Paquete(this.dimension, this.peso);  
}
```

Dimension y Paquete - Diagrama de objetos

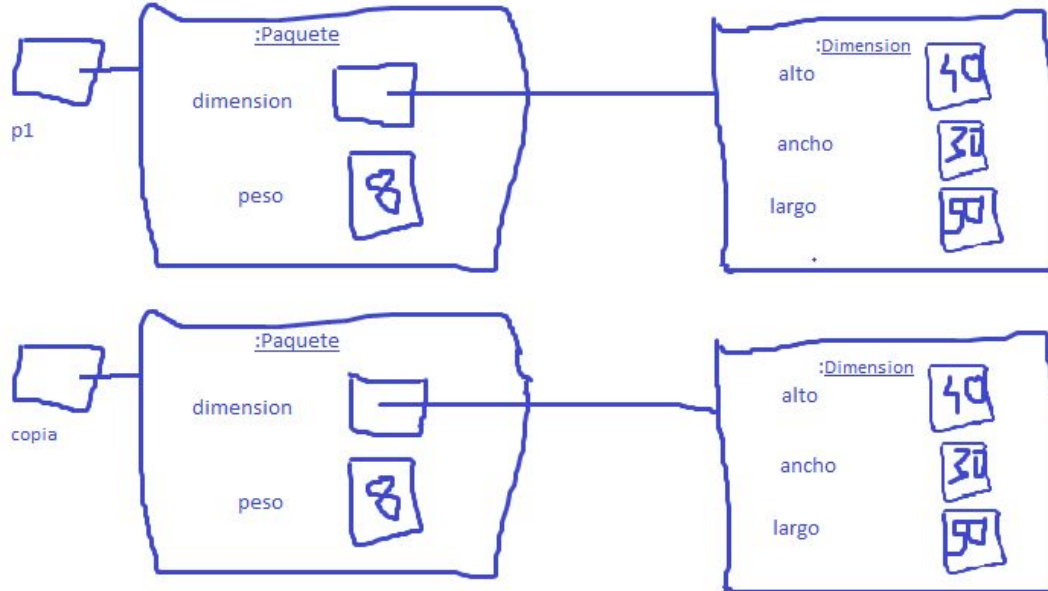
En clase Paquete

```
/**  
 * Devuelve una copia exacta al objeto actual  
 * Se obtienen copias también de los objetos que contenga  
 */  
public Paquete obtenerCopia() {  
    return new Paquete(this.dimension.obtenerCopia(), this.peso);  
}
```

Dimension y Paquete - Diagrama de objetos

Paquete p1 = new Paquete();

Paquete copia = p1.obtenerCopia();



```
/**  
 * Devuelve una copia exacta al objeto actual  
 * Se obtienen copias también de los objetos que contenga  
 */  
public Paquete obtenerCopia() {  
    return new Paquete(this.dimension.obtenerCopia(),  
                        this.peso);  
}
```