

COMPOSICIÓN - DELEGACIÓN

CESSI #ArgentinaPrograma #YoProgramo

Leonardo Blautzik - Federico Gasior - Lucas Videla

Agosto -Diciembre de 2021

Definición:

La **composición** consiste en crear una clase nueva agrupando objetos de clases que ya existen.

Una **composición** agrupa uno o más objetos para construir una clase, de manera que las instancias de esta nueva clase **contienen** uno o más objetos de otras clases.

Definición:

La **delegación** es una técnica en la que un objeto de cara al exterior expresa cierto comportamiento pero en realidad **delega** la **responsabilidad** de implementar dicho comportamiento a un objeto asociado (que contiene) en una relación inversa de responsabilidad.

Tabajemos con un ejemplo para ver de que se trata:

La corona circular

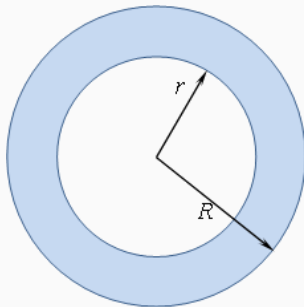


Figure 1: Corona circular

Se inicializa a partir de sus radios interior y exterior. Se debe poder obtener: radio interior, radio exterior, área y perímetro de la corona circular.

Primer implementación

```
public class CoronaCircular {  
  
    private double radioInterior;  
    private double radioExterior;  
  
    public CoronaCircular(double radioInterior, double radioExterior) {  
        if(radioInterior >= radioExterior)  
            throw new Error ("El radio interior debe ser menor que el exterior");  
        this.setRadioInterior(radioInterior);  
        this.setRadioExterior(radioExterior);  
    }  
}
```

```
public double getRadioInterior() {  
    return this.radioInterior;  
}  
  
public double getRadioExterior() {  
    return this.radioExterior;  
}  
  
public double getPerimetro() {  
    return (this.radioInterior+this.radioExterior)*2*Math.PI;  
}  
  
public double getSuperficie() {  
    return Math.PI*(Math.pow(radioExterior,2)-Math.pow(radioInterior,2));  
}
```

```
private void setRadioInterior(double ri) {  
    if(ri < 0 ) throw new Error("No se aceptan radios negativos");  
    this.radioInterior = ri;  
}  
private void setRadioExterior(double re) {  
    if(re < 0 ) throw new Error("No se aceptan radios negativos");  
    this.radioExterior = re;  
}  
}
```

Otra implementación

La class CoronaCircular contiene dos objetos Circulo

```
public class CoronaCircular {  
  
    private Circulo circuloChico;  
    private Circulo circuloGrande;  
  
    public CoronaCircular(double radioInterior, double radioExterior) {  
        if(radioInterior >= radioExterior)  
            throw new Error ("El radio interior debe ser menor que el exterior");  
        this.circuloChico = new Circulo(radioInterior);  
        this.circuloGrande = new Circulo(radioExterior);  
    }  
}
```



```
public double getRadioInterior() {  
    return this.circuloChico.getRadio();  
}  
public double getRadioExterior() {  
    return this.circuloGrande.getRadio();  
}  
public double getPerimetro() {  
    return this.circuloChico.getPerimetro()  
        + this.circuloGrande.getPerimetro();  
}  
public double getSuperficie() {  
    return this.circuloGrande.getSuperficie()  
        - this.circuloChico.getSuperficie();  
}
```

Hay que probar ambas implementaciones

Vamos a eclipse...

Tragamonedas



Figure 2: Tragamonedas

Un Tragamonedas está compuesto por 3 Tambores. Cuando el Tragamonedas se activa, giran los 3 Tambores. Cada Tambor se detiene en una posición comprendida entre 1 y 8. El Tragamonedas entrega un premio cada vez que, luego de ser activado los 3 Tambores se detienen en la misma posición. Implementar las clases Tragamonedas usando la clase Tambor:

Implementamos la class Tambor

```
class Tambor {  
    private int posicion;  
    /**  
     * post: inicialmente el Tambor esta en la posicion 1.  
     */  
    public Tambor() {  
        posicion = 1;  
    }  
    /**  
     * post: devuelve el numero de posicion en la que se encuentra el Tambor.  
     * Es un valor comprendido entre 1 y 8.  
     */  
    public int getPosicion() {  
        return posicion;  
    }  
}
```

```
/**
 * post: hace girar el tambor y luego se detiene en
 *       una posicion comprendida entre 1 y 8.
 */
public void girar() {
    /* provee un numero aleatorio en el rango [0.0, 1.0) */
    double aleatorio = Math.random();
    posicion = (int) (aleatorio * 8) + 1;
}
}
```

La class Tragamonedas usa la class Tambor

```
class Tragamonedas {  
    /**  
     * post: los 3 Tambores del Tragamonedas estan en la posicion 1.  
     */  
    public Tragamonedas() { }  
    /**  
     * post: activa el Tragamonedas haciendo girar sus 3 Tambores.  
     */  
    public void activar() { }  
    /**  
     * post: indica si el Tragamonedas entrega un premio  
     *         a partir de la posicion de sus 3 Tambores.  
     */  
    public boolean entregaPremio() { }  
}
```

Veamos la implementación

Vamos a eclipse a tirar unos fichines...