

Modificadores de acceso

Función y tipos

- Determinan la visibilidad de los miembros de una clase
- La siguiente tabla indica los tipos de modificadores y donde se aplican

	public	protected	(default)	private
clase	SI	NO	SI	NO
atributo	SI	SI	SI	SI
método	SI	SI	SI	SI
constructor	SI	SI	SI	SI

public

➤ Si una clase o miembro de la misma es public, puede utilizarse desde cualquier clase que esté en su mismo paquete o en cualquier otro:

```
package p1;  
public class Test{  
    public Test(int a){}  
    public void metodo(){}  
}
```

```
package p2;  
import p1.Test;  
class Otra{  
    void metodoEx(){  
        Test t=new Test(10); //ok  
        t.metodo(); //ok  
    }  
}
```

protected

➤ Si un miembro de la clase se declara como **protected**, será accesible desde cualquier clase de su mismo paquete y de sus subclases, independientemente de donde estas se encuentren:

```
package p1;  
public class Prueba{  
    protected int k=2;  
    protected void metodo(){}  
}
```

```
package p2;  
public class Test extends Prueba{  
    public void nuevoMetodo(){  
        this.k=10; //correcto  
        this.metodo(); //correcto  
    }  
}
```

(default)

➤ Es el ámbito que se aplica cuando no se indica ningún modificador. El elemento que lo lleve solo es accesible desde clases de su mismo paquete

```
package p1;  
public class Test{  
    Test(){  
        public Test(int a){}  
        void metodo(){  
    }  
    class Test2{  
        void tester(){  
            Test t=new Test(); //ok  
            t.metodo(); //ok  
        }  
    }  
}
```

```
package p2;  
import p1.Test;  
class Otra{  
    void metodoEx(){  
        Test t=new Test(10); //ok  
        t.metodo(); //error compilación  
        Test t2=new Test(); //error compilación,  
                                //constructor no public  
    }  
}
```

private

➤ El miembro solo es accesible desde el interior de la clase. Muy habitual en atributos para encapsulación.

```
public class Mesa{
    private int largo;
    private int ancho;
    private String color;
    :
}
class Otra{
    void metodo(){
        Mesa m=new Mesa();
        m.largo=2; //error compilación
    }
}
```