

# 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

	<b>public</b>	<b>protected</b>	<b>Sin especificar</b>	<b>private</b>
<b>clase</b>	SI	NO	SI	NO
<b>atributo</b>	SI	SI	SI	SI
<b>método</b>	SI	SI	SI	SI
<b>constructor</b>	SI	SI	SI	SI

- Protected se estudia en la herencia

# 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  
    }  
}
```

# (Sin especificar) Package-private

➤ 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(){} //default  
    public Test(int a){}  
    void metodo(){} //default  
}  
class Test2{ //default  
    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
    }
}
```