

Niveles de Acceso en Java

Los niveles de acceso nos introducen en el concepto de encapsulamiento, este oculta los detalles de la implementación de un objeto.

Es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.

Garantiza que los módulos pueden ser implementados y reimplementados independientemente, ya que ningún otro modulo puede depender de sus detalles de implementación. Los niveles de acceso permiten dar un nivel de seguridad mayor a nuestras aplicaciones restringiendo el acceso a diferentes atributos, métodos, constructores asegurándonos que el usuario deba seguir una "ruta" especificada por nosotros para acceder a la información [1].

Private

en Java es el más restrictivo de todos, básicamente cualquier elemento de una clase que sea privado puede ser accedido únicamente por la misma clase por nada más. Es decir, si por ejemplo, un atributo es privado solo puede ser accedido por los métodos o constructores de la misma clase. Ninguna otra clase sin importar la relación que tengan podrá tener acceso a ellos [1].

```
1 package map.ejemplo1;
2 public class Ejemplo1
3 {
4     private int atributo1; //Este atributo es privado
5     private int contador = 0; //Contador de registros
6
7     //Este atributo es privado entonces crear metodo que lo acc...
8     // para poder acceder al mismo a traves de los metodos
9
10    public void setAtributo1(int value)
11    {
12        contador++; //Incremento que tiene el registro de ediciones
13        atributo1 = value; //Asignamos el valor del atributo
14    }
15
16    public int getAtributo1()
17    {
18        return atributo1; //Retornamos el valor del atributo
19    }
20
21    //Met para el contador
22    public int getContador()
23    {
24        return contador;
25    }
26 }
```

Imagen 1 Código nivel de acceso private

En la imagen 1 tenemos un atributo privado y permitimos el acceso a él únicamente por medio de los métodos de get y set, notemos que estos métodos son públicos y por tanto cualquiera puede accederlos. Lo realmente interesante con los métodos get y set es que nos permiten realizar cualquier operación como por ejemplo llevar una cuenta de las veces que se estableció el valor para el atributo permitiéndonos mantener nuestro sistema sin problemas. También debemos notar que debido a que los métodos get y set son propios de la clase no tienen problemas con acceder al atributo directamente [1].

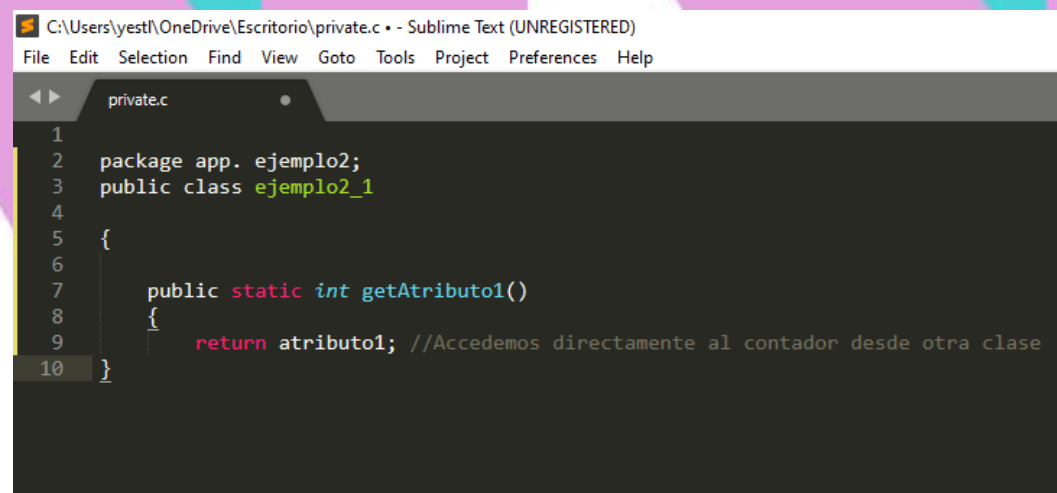
default

Java nos da la opción de no usar un modificador de acceso y al no hacerlo, el elemento tendrá un acceso conocido como default acceso por defecto que permite que tanto la propia clase como las clases del mismo paquete accedan a dichos componentes (de aquí la importancia de declararle siempre un paquete a nuestras clases) [1].



```
1 package app. ejemplo2;
2 public class ejemplo2
3 {
4     private static int atributo1; //Este atributo es privado
5     static int contador = 0; //Contador con acceso por defecto (default)
6     public static void setAtributo1(int valor)
7     {
8         contador++; //Contador que lleva el registro de ediciones
9         atributo1 = valor; //Establecemos el valor del atributo
10    }
11    public static int getAtributo1()
12    {
13        return atributo1; //Retornamos el valor del atributo
14    }
15 }
```

Imagen 2 Código nivel de acceso default

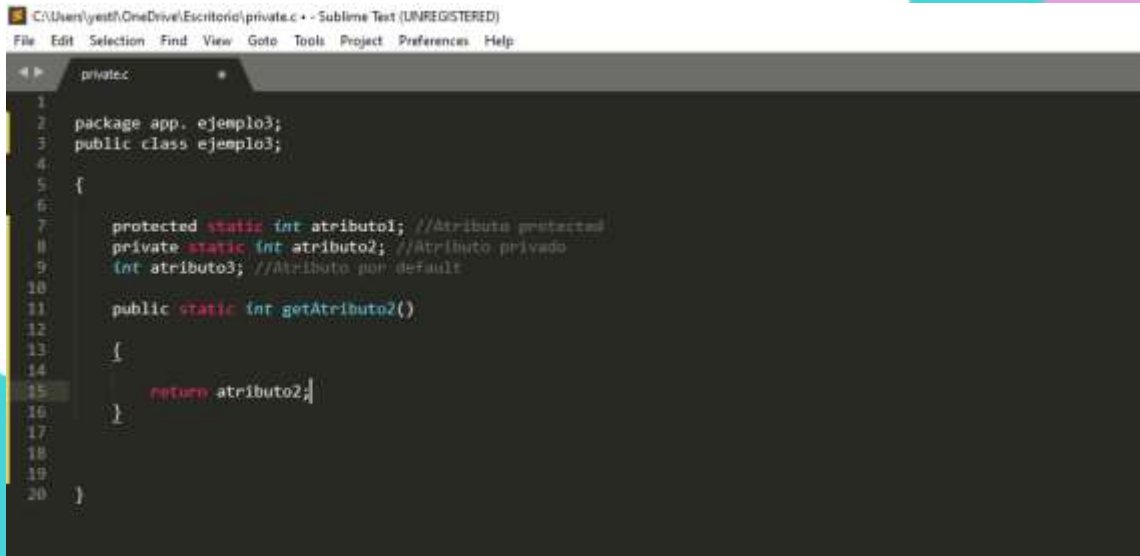


```
1 package app. ejemplo2;
2 public class ejemplo2_1
3 {
4     public static int getAtributo1()
5     {
6         return atributo1; //Accedemos directamente al contador desde otra clase
7     }
8 }
```

Imagen 3 Código nivel de acceso default 1

protected

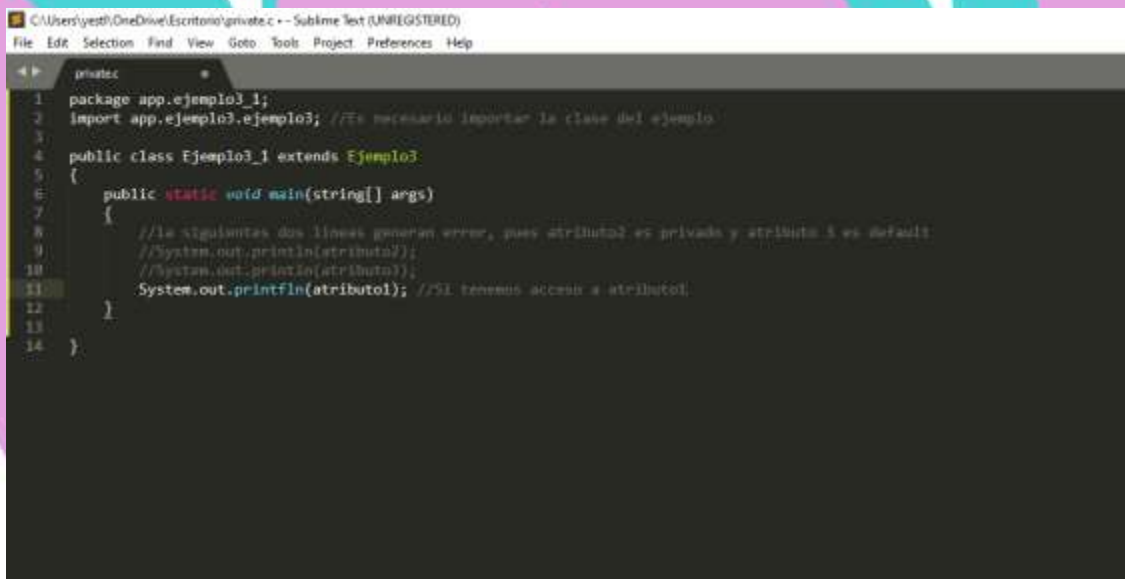
El modificador de acceso `protected` nos permite acceso a los componentes con dicho modificador desde la misma clase, clases del mismo paquete y clases que hereden de ella (incluso en diferentes paquetes) [1].



```
C:\Users\yesth\OneDrive\Escritorio\private.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

private.c
1 package app.ejemplo3;
2 public class ejemplo3;
3
4 {
5
6     protected static int atributo1; //Atributo protected
7     private static int atributo2; //Atributo privado
8     int atributo3; //Atributo por default
9
10    public static int getAtributo2()
11    {
12
13        return atributo2;
14    }
15
16
17
18
19
20 }
```

Imagen 4 Código de acceso `protected`



```
C:\Users\yesth\OneDrive\Escritorio\private.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

private.c
1 package app.ejemplo3_1;
2 import app.ejemplo3.ejemplo3; //Es necesario importar la clase del ejemplo
3
4 public class Ejemplo3_1 extends Ejemplo3
5 {
6     public static void main(String[] args)
7     {
8         //las siguientes dos líneas generan error, pues atributo2 es privado y atributo3 es default
9         //System.out.println(atributo2);
10        //System.out.println(atributo3);
11        System.out.println(atributo1); //Si tenemos acceso a atributo1.
12    }
13
14 }
```

Imagen 5 Código de acceso `protected 2`

Public

El modificador de acceso public es el más permisivo de todos, básicamente public es lo contrario a private en todos los aspectos (lógicamente), esto quiere decir que si un componente de una clase es public, tendremos acceso a él desde cualquier clase o instancia sin importar el paquete o procedencia de ésta [1].

```
C:\Users\yesth\OneDrive\Escritorio\private.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

private.c
1 package aap.ejemplo4;
2 public class Ejemplo4
3 {
4     public static int atributo1; //atributo publico
5     public static void metodo1()
6     {
7         System.out.println("Metodo publico");
8     }
9 }
```

Imagen 6 Niveles de acceso en java public

```
C:\Users\yesth\OneDrive\Escritorio\private.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

private.c
1 package paquete.externo;
2 import aap.ejemplo4.Ejemplo4; //importamos la clase del ejemplo4
3
4 public class ClaseExterna
5 {
6     public static void main(String[] args)
7     {
8         System.out.println(Ejemplo4.atributo1);
9         //Tendremos acceso directo por ser publico
10     }
11     Ejemplo4.metodo1(); //Metodo1 tambien es publico
12 }
13 }
```

Imagen 7 Niveles de acceso en java public

Tabla de funcionamiento

Modificador	La misma clase	Mismo paquete	Subclase	Otro paquete
private	Sí	No	No	No
default	Sí	Sí	No	No
protected	Sí	Sí	Sí/No	No
public	Sí	Sí	Sí	Sí

Tabla 1 Modificadores de acceso

Bibliografía

[1] J. D. M. González, «ProgramarYa,» 2012-2021. [En línea]. Available: <https://www.programarya.com/Cursos/Java/Modificadores-de-Acceso>. [Último acceso: 09 marzo 2021].

Imagen 1 Código nivel de acceso private	1
Imagen 2 Código nivel de acceso default	2
Imagen 3 Código nivel de acceso default 1	2
Imagen 4 Código de acceso protected	3
Imagen 5 Código de acceso protected 2	3
Imagen 6 Niveles de acceso en java public	4
Imagen 7 Niveles de acceso en java public	4
Tabla 1 Modificadores de acceso	5