

CURSO DE PROGRAMACIÓN CON JAVA

EJERCICIO

MODIFICADORES DE ACCESO EN JAVA



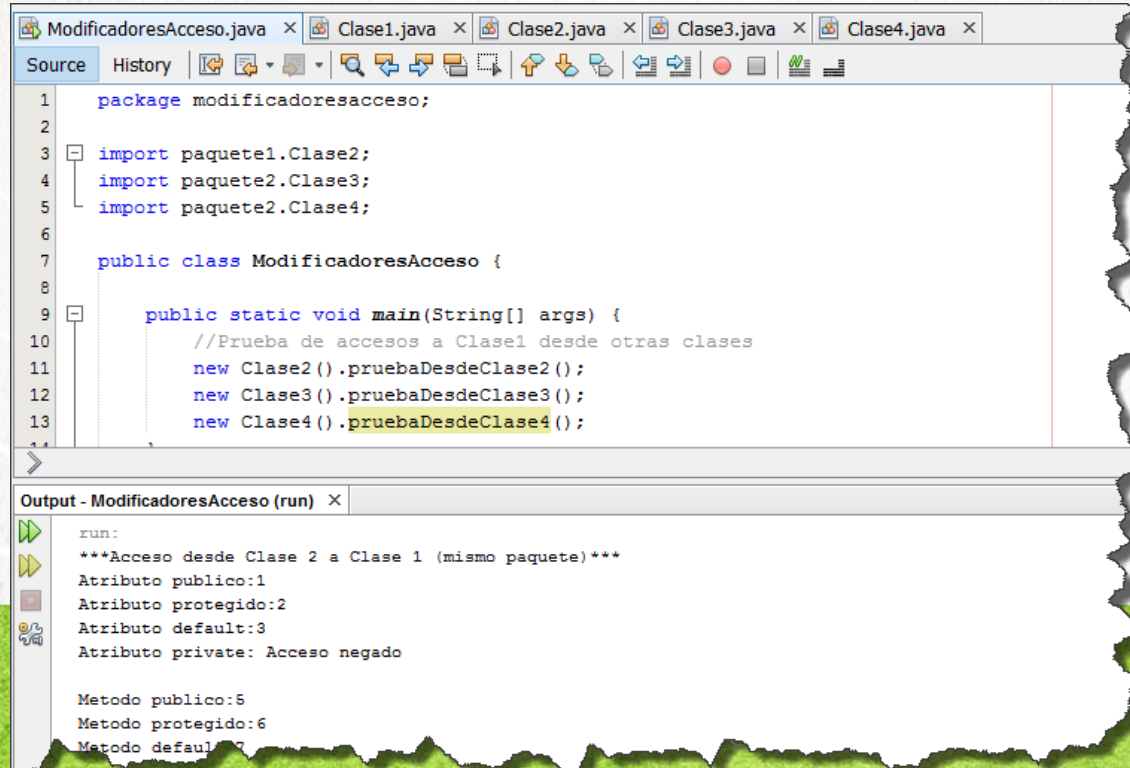
Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Crear un programa para practicar el manejo de modificadores de acceso en Java. Al finalizar deberemos observar lo siguiente:

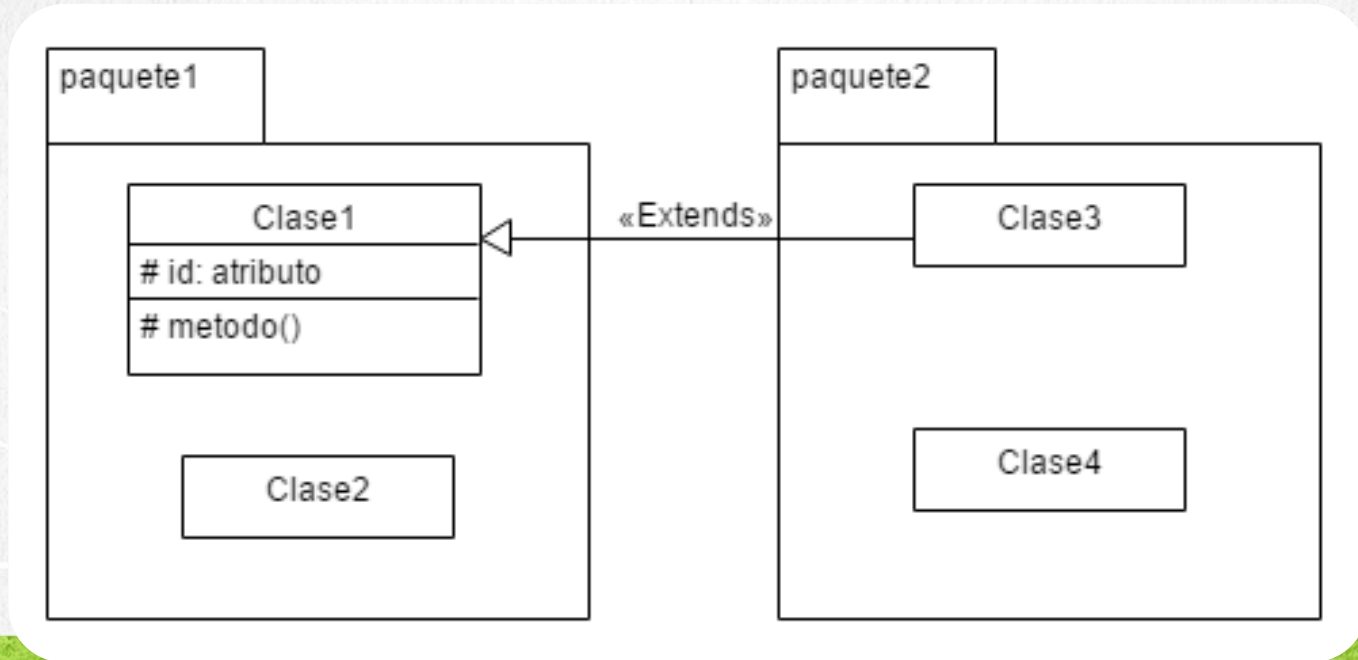
The image shows a screenshot of an IDE with several tabs open: 'ModificadoresAcceso.java', 'Clase1.java', 'Clase2.java', 'Clase3.java', and 'Clase4.java'. The 'Source' tab is active, displaying the code for 'ModificadoresAcceso.java'. The code includes package declarations, imports for 'Clase2', 'Clase3', and 'Clase4', and a 'main' method that creates instances of these classes and calls their 'pruebaDesdeClaseX()' methods. The 'Output - ModificadoresAcceso (run)' tab is also visible, showing the execution results. The output text is as follows:

```
run:
***Acceso desde Clase 2 a Clase 1 (mismo paquete)***
Atributo publico:1
Atributo protegido:2
Atributo default:3
Atributo private: Acceso negado

Metodo publico:5
Metodo protegido:6
Metodo default:7
```

OBJETIVO DEL EJERCICIO

El proyecto será basado en el siguiente diagrama de clases:



CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto ModificadoresAcceso:

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location: Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > **Finish** Cancel Help

PASO 2. CREACIÓN DE UNA CLASE

Vamos a crear una nueva clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: Clase1

Project: ModificadoresAcceso

Location: Source Packages

Package: paquete1

Created File: ramacionJava\Leccion09\ModificadoresAcceso\src\paquete1\Clase1.java

< Back Next > **Finish** Cancel Help

PASO 3. CREACIÓN DE UNA CLASE

Vamos a crear una nueva clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

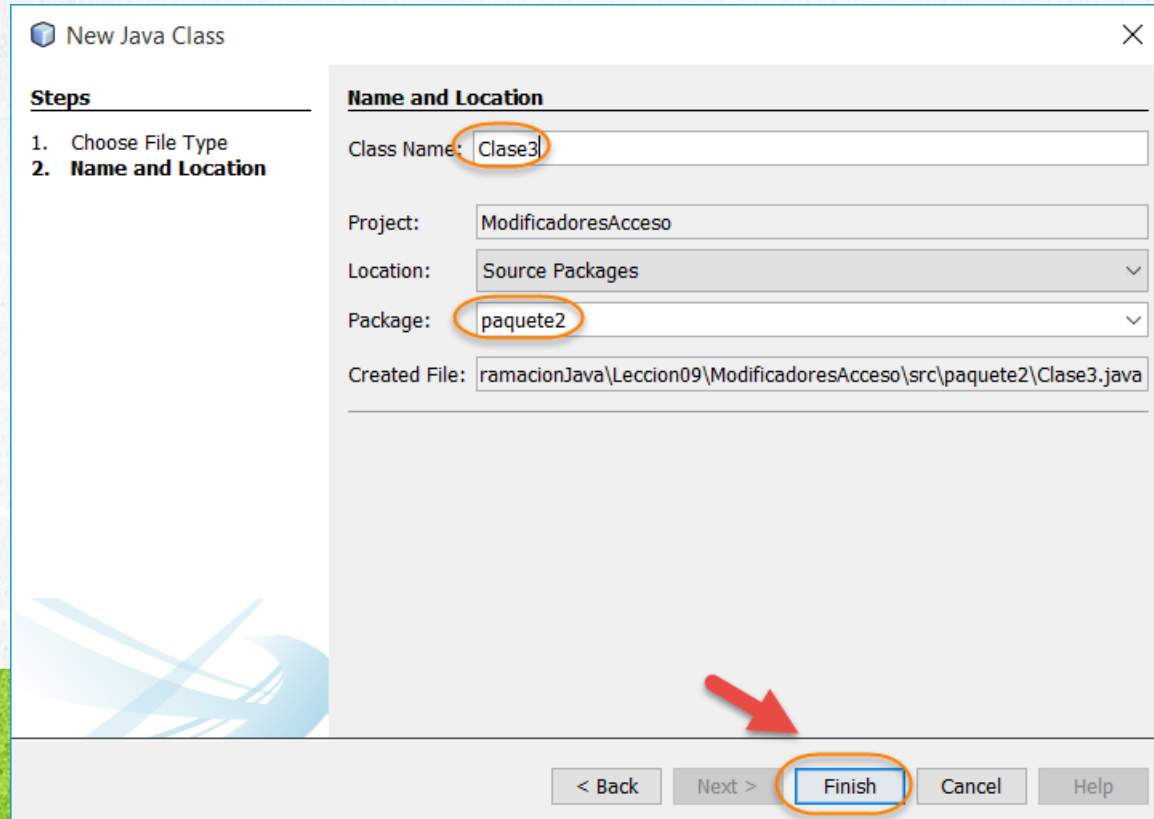
Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 4. CREACIÓN DE UNA CLASE

Vamos a crear una nueva clase:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: Clase3

Project: ModificadoresAcceso

Location: Source Packages

Package: paquete2

Created File: ramacionJava\Leccion09\ModificadoresAcceso\src\paquete2\Clase3.java

< Back Next > Finish Cancel Help

PASO 5. CREACIÓN DE UNA CLASE

Vamos a crear una nueva clase:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: Clase4

Project: ModificadoresAcceso

Location: Source Packages

Package: paquete2

Created File: ramacionJava\Leccion09\ModificadoresAcceso\src\paquete2\Clase4.java

< Back Next > **Finish** Cancel Help

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo Clase1.java:

```
package paquete1;

public class Clase1 {

    //Definicion de atributos
    public int atrPublico = 5;
    protected int atrProtegido = 6;
    int atrPaquete = 7;
    private int atrPrivado = 8;

    //Constructores
    public Clase1() {
    }

    public Clase1(int i) {
        System.out.println("Constructor publico 1");
    }

    protected Clase1(int i, int j) {
        System.out.println("Constructor protected 2");
    }

    Clase1(int i, int j, int k) {
        System.out.println("Constructor package 3");
    }

    private Clase1(int i, int j, int k, int l) {
        System.out.println("Constructor privado 4");
    }
}
```

```
//Definicion de métodos
public int metodoPublico() {
    return 9;
}

protected int metodoProtegido() {
    return 10;
}

int metodoPaquete() {
    return 11;
}

private int metodoPrivado() {
    return 12;
}
}
```

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo Clase2.java:

```
package paquetel;

public class Clase2 {

    public Clase2() {
        //Prueba constructores
        System.out.println("");
        //Constructor publico
        new Clase1(1);
        //Constructor protegido
        new Clase1(1, 2);
        //Constructor default o package
        new Clase1(1, 2, 3);
        //Constructor private
        //new Clase1(1, 2, 3,4);
        System.out.println("Constructor private: Acceso negado");
    }

    public void pruebaDesdeClase2() {
        //Acceso a Clase 1 desde Clase2
        Clase1 c1 = new Clase1();
        System.out.println("");
        System.out.println("Atributo publico:" + c1.ATR_PUBLICO);
        System.out.println("Atributo protegido:" + c1.ATR_PROTEGIDO);
        System.out.println("Atributo default:" + c1.ATR_PAQUETE);
        System.out.println("Atributo private: Acceso negado");

        System.out.println("");
        System.out.println("Metodo publico:" + c1.METODO_PUBLICO());
        System.out.println("Metodo protegido:" + c1.METODO_PROTEGIDO());
        System.out.println("Metodo default:" + c1.METODO_PAQUETE());
        System.out.println("Metodo private: Acceso negado");
    }
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo Clase3.java:

```
package paquete2;
import paquete1.Clase1;
public class Clase3 extends Clase1{

    public Clase3(){
        //Constructor protegido, al ser una subclase puede acceder
        //aunque esté en otro paquete
        super(1,2);
        //Acceso package, y al estar fuera de paquete esta restringido
        //super(1,2,3);
        //Acceso privado, restringido
        //super(1,2,3,4);
    }

    public void pruebaDesdeClase3() {
        //Acceso a Clase 1 desde Clase2
        //Clase 3 extiende de Clase 1
        Clase1 c1 = new Clase1();
        System.out.println("");
        System.out.println("Atributo publico:" + c1.atrPublico + " o heredado:" + atrPublico);
        System.out.println("Atributo protegido (heredado):" + atrProtegido);
        System.out.println("Atributo default: No se puede acceder desde un paquete externo ");
        System.out.println("Atributo private: Acceso negado" );

        //Constructor publico
        new Clase1();
        //Los demás constructores no se pueden probar asi, sino desde el constructor de esta clase
        //Ya que esta es una subclase en otro paquete

        System.out.println("");
        System.out.println("Metodo publico:" + c1.metodoPublico());
        System.out.println("Metodo protegido (heredado):" + metodoProtegido());
        System.out.println("Metodo default: No se puede acceder desde un paquete externo");
        System.out.println("Metodo private: Acceso negado" );
    }
}
```

PASO 9. MODIFICAMOS EL CÓDIGO

Archivo Clase4.java:

```
package paquete2;
import paquete1.Clase1;
public class Clase4 {
    public Clase4(){
        //Constructor protegido, restringido al no ser una subclase
        //super(1,2);
        //Acceso package, y al estar fuera de paquete esta restringido
        //super(1,2,3);
        //Acceso privado, restringido
        //super(1,2,3,4);
    }

    public void pruebaDesdeClase4 () {
        //Acceso a Clase 1 desde Clase4
        //Clase4 NO es subclase y esta en otro paquete
        Clase1 c1 = new Clase1();
        System.out.println("");
        System.out.println("Atributo publico:" + c1.atrPublico);
        System.out.println("Atributo protegido: No se puede acceder desde un paquete externo al NO ser una subclase");
        System.out.println("Atributo default: No se puede acceder desde un paquete externo");
        System.out.println("Atributo private: Acceso negado");

        //Constructor publico
        new Clase1();
        //los demas constructores están restringidos
        System.out.println("");
        System.out.println("Metodo publico:" + c1.metodoPublico());
        System.out.println("Metodo protegido: No se puede acceder desde un paquete externo al NO ser una subclase");
        System.out.println("Metodo default: No se puede acceder desde un paquete externo");
        System.out.println("Metodo private: Acceso negado");
    }
}
```


PASO 10. MODIFICAMOS EL CÓDIGO

Archivo ModificadoresAcceso.java:

```
package modificadoresacceso;

import paquete1.Clase2;
import paquete2.Clase3;
import paquete2.Clase4;

public class ModificadoresAcceso {

    public static void main(String[] args) {
        //Prueba de accesos a Clase1 desde otras clases
        //Acceso a Clase 1 desde Clase2
        System.out.println("***Acceso desde Clase 2 a Clase 1 (mismo paquete)***");
        new Clase2().pruebaDesdeClase2();

        //Acceso a Clase 1 desde Clase2
        //Clase 3 extiende de Clase 1
        System.out.println("\n***Acceso desde Clase 3 a Clase 1 (diferente paquete, pero es subclase)***");
        new Clase3().pruebaDesdeClase3();

        //Acceso a Clase 1 desde Clase4
        //Clase4 NO es subclase y esta en otro paquete
        System.out.println("\n***Acceso desde Clase 4 a Clase 1 (diferente paquete, NO es subclase)***");
        new Clase4().pruebaDesdeClase4();
    }
}
```

PASO 11. EJECUTAMOS EL PROYECTO

Resultado de ModificadoresAcceso.java:

Acceso desde Clase 2 a Clase 1 (mismo paquete)

Constructor publico 1
Constructor protected 2
Constructor package 3
Constructor private: Acceso negado

Atributo publico:5
Atributo protegido:6
Atributo default:7
Atributo private: Acceso negado

Metodo publico:9
Metodo protegido:10
Metodo default:11
Metodo private: Acceso negado

Acceso desde Clase 3 a Clase 1 (diferente paquete, pero es subclase)
Constructor protected 2

Atributo publico:5 o heredado:5
Atributo protegido (heredado):6
Atributo default: No se puede acceder desde un paquete externo
Atributo private: Acceso negado

Metodo publico:9

Metodo protegido (heredado):10

Metodo default: No se puede acceder desde un paquete externo

Metodo private: Acceso negado

Acceso desde Clase 4 a Clase 1 (diferente paquete, NO es subclase)

Atributo publico:5

Atributo protegido: No se puede acceder desde un paquete externo al NO ser una subclase

Atributo default: No se puede acceder desde un paquete externo

Atributo private: Acceso negado

Metodo publico:9

Metodo protegido: No se puede acceder desde un paquete externo al NO ser una subclase

Metodo default: No se puede acceder desde un paquete externo

Metodo private: Acceso negado

MACIÓN CON JAVA

TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Si es necesario probar con más elementos comprobando cada uno de los modificadores de acceso estudiados.

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica en concepto de modificadores de acceso.
- Como ya habíamos estudiado, los más utilizados son public y private, sin embargo es importante conocer la aplicación de los restantes como son protected y package y así conocer las aplicaciones que pueda tener cada uno de estos modificadores.



CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx

CURSO ONLINE

PROGRAMACIÓN CON JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE PROGRAMACIÓN CON JAVA

www.globalmentoring.com.mx