**ACTIVIDAD 11**

**Enunciado**

Realiza un programa Java (**Actividad11**) que ejecute el programa que se muestra a continuación (**Ejemplo2**). El programa Actividad11 sólo requiere adaptar el valor que se le asigna a la variable *comando*. En ambos programas se usan las clases Java Runtime y Processpara la gestión de procesos. Incluye los dos programas Java en el paquete por defecto que ofrece el IDE Eclipse.

Compila tanto el programa del enunciado (Ejemplo2) como el nuevo programa (Actividad11) desde la línea de comandos, aportando evidencias de cómo lo has hecho.

Después de que la compilación esté libre de errores, ejecuta el programa Java de nombre Actividad11 desde la línea de comandos, aportando evidencias de tanto su ejecución como del resultado de misma.

import java.io.\*;

public class Ejemplo2 {

public static void main(String[] args){

Runtime r=Runtime.*getRuntime*();

//String comando="ls -al"; //Descomentar la línea para Linux

//String comando="CMD C/ DIR"; //Descomentar la línea para Windows

Process p=null;

try {

p = r.exec (comando);

InputStream is = p.getInputStream();

BufferedReader br = new BufferedReader (new InputStreamReader(is));

String linea;

while((linea = br.readLine()) != null) // lee una linea

System.*out*.println(linea);

br.close();

}

catch (Exception e) {

e.printStackTrace();

}

///// Comprobación: Valor= 0 bien, Valor = - 1 mal

int exitVal;

try {

exitVal=p.waitFor();

System.*out*.println ("Valor de Salida "+exitVal);

} catch (InterruptedException e){

e.printStackTrace();

}

}

}

**Solución:**

// Ejemplo2

// Joseba Martinez

**package** code;

**import** java.io.\*;

**public** **class** Ejemplo2 {

**public** **static** **void** main(String[] args){

Runtime r=Runtime.*getRuntime*();

String comando="ls -al"; //Descomentar la línea para Linux

//String comando="CMD C/ DIR"; //Descomentar la línea para Windows

Process p=**null**;

**try** {

p = r.exec (comando);

InputStream is = p.getInputStream();

BufferedReader br = **new** BufferedReader (**new** InputStreamReader(is));

String linea;

**while**((linea = br.readLine()) != **null**) // lee una linea

System.***out***.println(linea);

br.close();

}

**catch** (Exception e) {

e.printStackTrace();

}

///// Comprobación: Valor= 0 bien, Valor = - 1 mal

**int** exitVal;

**try** {

exitVal=p.waitFor();

System.***out***.println ("Valor de Salida "+exitVal);

} **catch** (InterruptedException e){

e.printStackTrace();

}

}

}

// Actividad11

// Joseba Martinez

**package** code;

**import** java.io.BufferedReader;

**import** java.io.InputStream;

**import** java.io.InputStreamReader;

**public** **class** Actividad11 {

**public** **static** **void** main(String[] args) {

Runtime r=Runtime.*getRuntime*();

// String comando="/Applications/Firefox.app/Contents/MacOS/firefox";

// String comando="Notepad"; // en windows

String comando="java code.Ejemplo2"; // en linux

Process p;

**try** {

p = r.exec(comando);

InputStream is = p.getInputStream();

BufferedReader br = **new** BufferedReader (**new** InputStreamReader(is));

String linea;

**while**((linea = br.readLine()) != **null**) // lee una linea

System.***out***.println(linea);

br.close();

}**catch** (Exception e){

System.***out***.println("error en:"+comando);

e.printStackTrace();

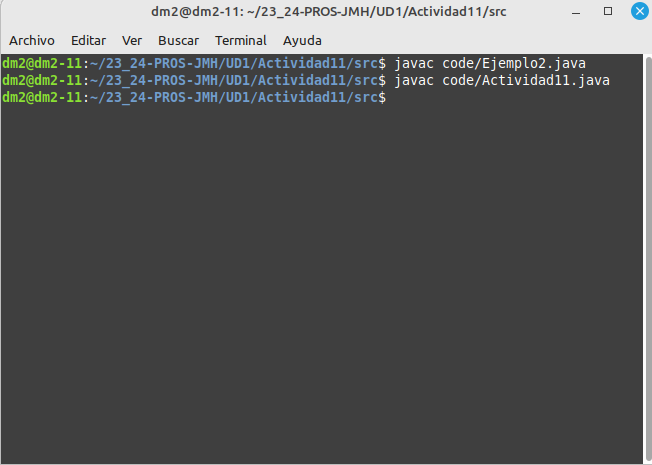
}

}

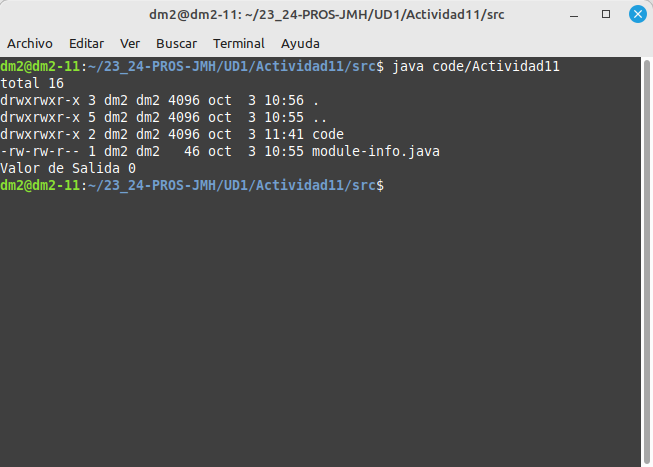
}

**Evidencias de la solución:**

* Compilación desde la línea de comandos de los programas fuente Ejemplo2.java y Actividad11.java:



* Ejecución del programa Actividad11.class desde la línea de comandos:

****

**Resolución de problemas:**

no se mostraban los printf de Ejemplo2, por lo que se ha trasladado parte del código a Actividad11 para solucionarlo.

**CRITERIOS DE CORRECCIÓN**

1. Nivel de ajuste a la nomenclatura de objetos definida en el enunciado de la actividad y nivel de ajuste al almacenamiento de la información requerida en las subcarpetas definidas para la actividad y unidad didáctica correspondientes (15%).
2. Correcto funcionamiento de los programas desarrollados en C (35%).
3. Ajuste a los resultados esperados (35%).
4. Claridad del código C (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida).
6. Avisos o errores de compilación (-15%).
7. Nivel de comentarios en el código fuente (-15%).