ENTORNOS DE DESARROLLO

DESARROLLO JAVA

ACTIVIDADES REFLEXIÓN (II)

Alberto Martínez Pérez

1º CFGS DESARROLLO DE APLICACIONES WEB (daw)

ÍNDICE

[1. ¿QUÉ ES UN PAQUETE? 3](#_Toc119039726)

[2. ¿CÓMO SE ORGANIZA UN PAQUETE EN EL SISTEMA DE DIRECTORIOS? 3](#_Toc119039727)

[3. ¿QUÉ ES UNA CLASE? PON UN EJEMPLO 3](#_Toc119039728)

[4. ¿QUÉ ES UN ATRIBUTO? PON UN EJEMPLO 3](#_Toc119039729)

[5. ¿CUÁNTOS MAIN PUEDE HABER EN UN PROYECTO? 3](#_Toc119039730)

[6. ¿CUÁNTOS FICHEROS PUEDE CONTENER UN PROYECTO? 3](#_Toc119039731)

[7. ¿TENGO QUE HACER UN IMPORT DE UNA CLASE QUE ESTÁ EN MI PAQUETE PARA PODER USARLA? 4](#_Toc119039732)

[8. ¿QUÉ ES LA HERENCIA? 4](#_Toc119039733)

[9. ¿CÓMO SE INDICA QUE UNA CLASE HEREDA DE OTRA? 4](#_Toc119039734)

[10. ¿QUÉ ES UN FICHERO .JAR? 4](#_Toc119039735)

[11. ¿PARA QUÉ SE CREA UN .JAR? 4](#_Toc119039736)

[12. ¿PARA QUÉ SIRVE EL FICHERO MANIFEST.MF? 4](#_Toc119039737)

# ¿Qué es un paquete?

Son el mecanismo que usa Java para facilitar la modularidad del código. De forma resumida se puede decir que un paquete es un conjunto de clases o interfaces que tiene un programa java (aunque puede que sólo contengan una clase, por ejemplo, por lo general el paquete principal sólo contendrá la clase con el método main).

# ¿Cómo se organiza un paquete en el sistema de directorios?

Dentro de un directorio raíz (que llevaría el nombre de nuestro programa) se pueden crear tantos paquetes como se desee, es decir, los paquetes son subdirectorios dentro del directorio raíz.

# ¿Qué es una clase? Pon un ejemplo

Es un conjunto de métodos y atributos que definen únicamente a esa clase. Uno de los métodos especiales que tienen es el constructor que se encarga de inicializar un objeto de esa clase, el constructor debe llevar el mismo nombre que la clase.

Por ejemplo, una clase Calculadora que, a través de diferentes métodos como suma, resta, multiplicación… puede realizar diferentes operaciones matemáticas con unos atributos como intNum1, intNum2...

# ¿Qué es un atributo? Pon un ejemplo

Los atributos son las características que definen a una clase.

Por ejemplo, una clase Persona va a tener un String nombre, un int edad, un double estatura… Esos datos serían los atributos de la clase Persona.

# ¿Cuántos main puede haber en un proyecto?

Sólo puede existir un método main dentro de un proyecto.

# ¿Cuántos ficheros puede contener un proyecto?

Tantos como sean necesarios, no hay un límite máximo (sí mínimo ya que debemos tener al menos un fichero con el método main el cual, en ese caso, contendrá todo el código del programa).

# ¿Tengo que hacer un import de una clase que está en mi paquete para poder usarla?

No, realizaremos imports de clases que estén en paquetes diferentes al de la clase que estemos usando en ese momento. Cuando las clases estén en el mismo paquete podremos llamar a los métodos propios de cada una (o realizar herencia) sin necesidad una importación de las clases.

# ¿Qué es la herencia?

Es un mecanismo de la programación orientada a objetos que permite definir una clase a partir de una ya existente previamente, por tanto, la herencia nos va a permitir compartir métodos y atributos desde unas clases padre a unas clases hija.

# ¿Cómo se indica que una clase hereda de otra?

Se indica en la creación de la clase mediante la palabra reservada o cláusula extends y con la siguiente sintaxis public class nombreClaseHija extends nombreClasePadre.

Por ejemplo, si tenemos una clase Alumno que comparte atributos con una clase Persona creada previamente podemos escribir la sintaxis public class Alumno extends Persona y con ello conseguir que haya una herencia de esos atributos.

# ¿Qué es un fichero .jar?

Es un archivo que contiene todos los archivos .class del proyecto Java que hayamos creado. Se puede decir que es el ejecutable de Java.

# ¿Para qué se crea un .jar?

Para comprimir todos los archivos .class en un único archivo. Esto facilita la portabilidad al ser un único archivo y, además, facilita su ejecución.

# ¿Para qué sirve el fichero MANIFEST.MF?

Es un archivo de metadatos que sirve para indicar cuál es la clase principal de un proyecto Java. Por lo general en su interior vamos a tener una única línea con la estructura Main-Class: principal.nombreClasePrincipal.