**Clase 9- Actividadades**

**Autor: Eduardo Zurbriggen**

Ejercicios: Actividades a Desarrollar

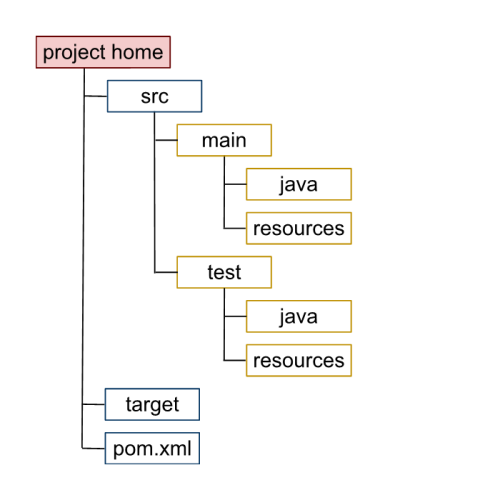
1. El comando “mvn install”, ¿para qué sirve?

Genera el artefacto, lanza los tests y lo instala en el repositorio local de Maven

2. ¿El archivo principal de configuración de un proyecto gestionado por Maven tiene extensión XML? ¿Cómo es su estructura?

La unidad básica de trabajo en Maven es el llamado Modelo de Objetos de Proyecto conocido simplemente como POM (de sus siglas en inglés: Project Object Model).Se trata de un archivo XML llamado pom.xml que se encuentra por defecto en la raíz de los proyectos y que contiene toda la información del proyecto: su configuración, sus dependencias, etc...

Por defecto, el directorio donde está el código fuente es src/main/java, donde se compila el proyecto es target y donde ubicamos los test unitarios es en src/main/test, etc... Al pom.xml global, con los valores predeterminados se le llama Súper POM.



3. Liste los comandos básicos de Maven para Elipse. Describa brevemente cada uno

mvn compile: Para compilar los ficheros .java en .class

mvn test: Se compilan las clases y se lanzan los tests de junit.mvn package: Se compilan las clases, se lanzan los tests y se genera un entregable de nuestro artefacto (por defecto un .jar)

mvn install: Genera el artefacto, lanza los tests y lo instala en el repositorio local de maven.

mvn deploy: Despliega el artefacto en un repositorio remoto.

4. Comandos básicos de git

git config

Establece valores de configuración para tu usuario, email, gpg key, algoritmo diff preferido, formatos de archivo y más

git init

Inicializa un repositorio git – crea el directorio .git inicial en un proyecto nuevo o existente

git add

Añade cambios de archivos en tu directorio de ensayo a tu index

git commit

Toma todos los cambios escritos en el index, crea un nuevo objeto de confirmación que apunta a él y establece la rama para que apunte a esa nueva confirmación

git status

Te muestra el estado de los archivos en el index en comparación con los del directorio de trabajo. Enumerará los archivos que no están rastreados (solo en su directorio de trabajo), modificados (rastreados pero aún no actualizados en tu index), y almacenados (añadidos a tu index y listos para comprometerse)

git branch

Para listar las ramas existentes, incluyendo las ramas remotas, si se proporciona ‘-a’. Crea una nueva rama si se proporciona un nombre.

git merge

Fusiona una o más ramas con otra rama activa y crea automáticamente un nuevo commit si no hay conflictos

git pull

Obtiene los archivos del repositorio remoto y los combina con el local.

git push

Envía todos los objetos modificados localmente al repositorio remoto

git log

Muestra una lista de confirmaciones en una rama que incluye los detalles correspondientes.

5.

Proyecto

Github

Realice modificaciones

No hago nada

Creo un proyecto o un repositorio en Github

Git status

Git add .

Git commit -m “msj”

Excepciones

1. ¿Funciona o se produce una excepción?

Se produce una excepción

2. Explique narrativamente la mecánica en caso que funcione o Explique donde y porque se la excepción en caso que no funcione.

cuando se llama la función reservar se produce una excepcion, ya que se debe enviarle un parámetro de tipo boolean

3. En cualquier caso: Agregue los bloques de control Try-Catch para atrapar las excepciones y muestre un mensaje adecuado por pantalla.

El bloque try-catch se debe agregar alrededor de la función reservar, ya que seria la línea en donde se puede producir un error. Mediante el bloque estaríamos manejando dicho error

4. Se imprime la línea que contiene el msj: La habitación ha sido reservada correctamente

Colecciones- Clase Hashmap

1.

Para obtener un valor usamos función get(Object key), con esta obtenemos el valor por medio de su key o llave.

Para actualizar o agregar un elemento en un HashMap en Java vamos a hacer uso de put. Con put vamos a obtener un valor por su key y sobreescribir encima su valor.

Map.put();

Para eliminar algún elemento usamos remove

Map.remove()

2.

a- la clave es un String y el valor un Producto objeto

b-no compila, a no ser que 137 sea un objeto de tipo Producto

c-verdadero

d-falso

Java literales

1. La barra invertida

2. La barra invertida se usa como carácter de escape, sus usos más habituales son para dar un salto de línea, agregar comillas simples o dobles dentro de un string cuando se quiere resaltar una palabra.

Aunque tiene muchos usos mas estos son los mas comunes

Java Metodos

1. Es un método

2. Proviene de java.time.\*;

Java – Archivos, bucles de control, Métodos

1. Se declara un string en donde se le asigna una ruta donde esta guardado un determinado archivo

Mediante un bucle for each, recorre linea a linea el archive txt.

Al usar el condicional if, le decimos al programa que si la línea que es leida, si empieza con un asterisco debe ser mostrada por pantalla

En caso que se produzca un error durante la ejecución del código se lanzara un objeto de la clase Exception

2. readAllLines pertenece a File

OIException de la clase Exception

Get de Path

3. d) b y c son las correctas

Java – Operadores lógicos y Bucles

1. False

True

True

2. Se imprime 7,5,3,1,0

Java – Arrays

1. El valor es 16

Junit – Unit Test

1. assertTrue/assertFalse (condición a testear): Comprueba que la condición es cierta o falsa.

assertEquals/assertNotEquals (valor esperado, valor obtenido). Es importante el orden de los valores esperado y obtenido.

assertNull/assertNotNull (object): Comprueba que el objeto obtenido es nulo o no.

assertSame/assertNotSame(object1, object2): Comprueba si dos objetos son iguales o no.

2. @Test: indica que el método que la contiene es un test.

@Before: ejecuta el método que la contiene justo antes de cada test.

@After: ejecuta el método que la contiene justo después de cada test.

3. Las respuestas correctas son a y b.

Java – POO

1. No tiene sentido

2. No, una clase solo puede heredar de una sola clase padre

3. La respuesta correcta es la opción d

4. El main es un método public (se puede acceder desde fuera de la clase) y static (se puede ejecutar sin una instancia de clase). En él se ejecuta el programa.

5. Público: atributos o métodos pueden ser utilizados por instancias de cualquier clase.

Protected: atributos o métodos pueden ser utilizados sólo por instancias de la misma clase o sus subclases.

Privado: atributos o métodos pueden ser utilizados sólo por instancias de la misma clase

6. Vamos a la opción Source, y seleccionamos Generate Getters and Setters.

Sino le damos click derecho en el código, click en source y click en Generate Getters and Setters.

Luego seleccionamos los atributos para los cuales queremos generar los métodos.

7. Par crear una instancia de clase u objeto debo usar la palabra new. Por ejemplo

Auto auto1= new Auto();

De esta manera estoy creando una instancia de la clase auto, por lo que se crea el objeto auto1.

8. Usando la palabra extends

Public class mesitaDeLuz extends Mueble() {}

9. Para heredar una interfaz tenemos que usar la palabla implement en la declaración.

Definimos la interfaz Gps(){}

Por ejemplo: class Coche extends Vehiculo implements Gps(){}

10. Los métodos se usan por medio de la notación del punto

Auto.puertas();

11. No se pueden usar, para poder hacer uso de la clase abstracta debe ser heredada

12. No se pueden utilizar ya que todos sus métodos son declarados como abstractos.

Eclipse

1. File-new-Java Proyect. En la ventana colocamos el nombre de nuestro proyecto y le damos a finalizar

2. File-new-Class. En la Ventana colocamos el nombre de la clase y tildamos la casilla para que nos coloque automáticamente el main y luego le damos a finalizar

3. File-new-pakage. En la Ventana colocamos el nombre de nuestro pague y le damos a finalizar

4. Debemos importar la clase en nuestro proyecto. Para esto usamos la palabra import seguido de la ruta