

Test final!!

rocksatetesteban@gmail.com Cambiar cuenta

\* Indica que la pregunta es obligatoria

Java

Cuántas veces se ejecutara el siguiente for? \*

2 puntos

```
for (int i = 2; i <= 128; i *= 2) {  
    System.out.println("i " + i);  
}
```

☐ Mostrará error

☐ 8

☐ 128

☐ 7

Los atributos static...

2 puntos

☐ Solo pueden ser usados dentro de metodos/funciones del tipo static

☐ Solo pueden ser usados dentro de metodos/funciones del tipo NO static

☐ Son si o si son del tipo de alguna Clase (como por ejemplo Integer)

☐ Ninguna de las anteriores es correcta

¿Qué es el encapsulamiento? \*

1 punto

☐ Hace referencia a crear métodos para obtener y modificar información de un objeto, manteniendo oculto lo que hace un objeto por dentro.

☐ Hace referencia a los métodos constructores con y sin parámetros

☐ Hace referencia a separar el código en "cápsulas" para mejor lectura del mismo.

¿Cuál de estas clases NO es parte del Collections Framework? \*

2 puntos

☐ Collection

☐ Matriz

☐ Iterator

☐ LinkedList

¿Cuál de los siguientes métodos NO es parte de la clase String? \*

1 punto

☐ .stringLength()

☐ concat()

☐ .isEmpty()

☐ .toUpperCase()

¿Qué significa el acrónimo UML? \*

2 puntos

☐ Ninguna de las anteriores

☐ Lenguaje de Modelo Único

☐ Lenguaje de Modelado Unificado

☐ Lenguaje de Mapas sobre Uniones

¿Cuál de las siguientes no es una Relación de clases? \*

2 puntos

☐ Asociación

☐ Navegacion

☐ Agregación

☐ Composición

¿Qué es un constructor? \*

1 punto

☐ Un método void

☐ Un método que nos define qué atributos tiene la clase

☐ Es un método que nos permite instanciar un objeto de la clase donde se encuentra

¿Qué hace el siguiente Comparator? \*

2 puntos

```
public static Comparator<Estudiante> compararEstudiantes= new Comparator<Estudiante>() {  
    @Override  
    public int compare(Estudiante e1, Estudiante e2) {  
        return e2.getNombre().compareTo(e1.getNombre());  
    }  
};
```

☐ Compara los estudiantes según su nombre de manera descendente

☐ Ninguna de las anteriores

☐ Compara los estudiantes según su nombre de manera ascendente

☐ Compara los estudiantes según su edad de manera ascendente

¿Qué mostraría por consola? \*

2 puntos

```
try {  
    double resul = 10 / 0;  
    System.out.println("Exito");  
} catch (ArithmeticException e) {  
    System.out.println("Error aritmetico");  
} catch (Exception e) {  
    System.out.println("Error Exception");  
} finally {  
    System.out.println("Terminamos");  
}
```

☐ "Error aritmetico", "Error Exception", "Terminamos"

☐ "Exito", "Terminamos"

☐ "Error Exception", "Terminamos"

☐ "Error aritmetico", "Terminamos"

¿La clase padre de todas las excepciones y errores, es la clase? \*

2 puntos

☐ Throwable

☐ Ninguna de las anteriores

☐ Error

☐ Exception

¿Las clases en UML se representan con un? \*

2 puntos

☐ Triangulo

☐ Circulo

☐ Ninguna de las anteriores

☐ Rectángulo

¿Cómo se llama el archivo que usa Maven para construir sus dependencias? \*

2 puntos

☐ Dependencies-properties.xml

☐ POM.xml

☐ Application-properties.yml

☐ Application-context.xml

Para decir que una clase hereda a otra se usa la palabra:

2 puntos

☐ Ninguna de las anteriores

☐ Extends

☐ Implements

☐ Inherits

¿Cuál es la diferencia entre clases y objetos? \*

1 punto

☐ Las clases son "molde" y los objetos son modelados a partir de esos moldes, también llamados "instancias"

☐ Los objetos son abstracciones del mundo real, las clases son de lunes a jueves

☐ Las clases y los objetos son los mismo

Elegir que anotación se aplica a la siguiente relación, un Perro pertenece a un Dueño:

2 puntos

☐ @OneToOne

☐ @ManyToOne

☐ @OneToMany

☐ @ManyToMany

¿Cómo accedo a los atributos de un objeto? \*

1 punto

☐ Con el símbolo ! luego del nombre del objeto

☐ Con el símbolo . luego del nombre del objeto

☐ Con el símbolo \_ luego del nombre del objeto

¿Cuál de estas herencias no es soportada en Java? \*

2 puntos

☐ Herencia Multinivel

☐ Herencia Simple

☐ Herencia Jerarquica

☐ Herencia Múltiple