

Tiempo restante 0:59:46

Pregunta 1

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Un método siempre debe capturar todas las excepciones que se lanzan en su bloque try.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

SIGUIENTE PÁGINA

Tiempo restante 0:54:40

Pregunta 2

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Cuando un bloque `try{}` puede lanzar excepciones de distinto tipo, la forma más correcta de atraparlas (a las excepciones que se decide atrapar) es declarar un bloque `catch()` con parámetro de tipo `Exception`, o sea:

```
catch(Exception e){ .....}
```

La respuesta correcta vale 1 punto (relativo al resto de las preguntas)

La respuesta equivocada resta 1/2 punto (relativo al resto de las preguntas)

- ☐ a. VERDADERO
- ☐ b. FALSO

Pregunta 3

Sin responder
aún

Puntúa como
3,00

🚩 Marcar
pregunta

Considere el método siguiente y diseñe la excepción

```
Socio
+ String nombre;
+ int legajo;
+ Categoria categoria;
public int modificarCategoriaSocio(int legajo, Categoria nuevaCategoria) throws NoExisteSocioException

{
    Socio aux = buscarSocio(legajo);
    if aux == null
        throw new NoExisteSocioException(...);
    // código para realizar la modificación
}
```

Supongamos que en un cierto ámbito, la solución al problema que causa la excepción sea agregar un nuevo socio.

- Realice la clase NoExisteSocioException, completa.
- Determine qué información debería recibir el constructor de la Excepción.
- Realice un código try/catch en el ámbito supuesto, donde se invocaría el método `modificarCategoriaSocio(...)`. Determine toda la información necesaria para poder solucionar el problema.
- Es necesario modificar el método `modificarCategoriaSocio(...)` ?
- Justifique sus respuestas.

Tiempo restante 0:43:03

Pregunta 4

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

El método clone() retorna la referencia al objeto que se quiere clonar

- ☐ a. VERDADERO
- ☐ b. FALSO

tiempo restante 0:40:08

Pregunta 5

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Un atributo privado no puede ser modificado en un método que sea público.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Tiempo restante 0:38:27

Pregunta 6

Sin responder aún

Puntuación como 3,00

Marcar pregunta

Explique en pocas palabras en qué casos es preferible utilizar una clase abstracta y en qué casos es preferible utilizar una interface.

De un ejemplo para cada caso.

Esta pregunta vale 3 puntos (relativo al resto de las preguntas)

Rich text editor toolbar with icons for undo, redo, bold, italic, bulleted list, numbered list, link, unlink, and image.

Pregunta 7

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Un Objeto referenciado por una variable/atributo "final" no puede cambiar su estado.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Tiempo restante 0:29:44

Pregunta **8**
Sin responder aún
Puntúa como 1,00
🚩 Marcar pregunta

Una variable de tipo Interface puede referenciar a cualquier objeto que la implemente.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. FALSO
- ☐ b. VERDADERO

tiempo restante: 0:00:00

Pregunta 9

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Las postcondiciones sobre un método determinan lo siguiente: el estado en que deben quedar los atributos de la clase, el retorno del método, el estado en que deben quedar los parámetros del método en caso de ser modificado su estado.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Tiempo restante 0:27:44

Pregunta 10

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Un método concreto no puede invocar un método abstracto.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Pregunta 10

Finalizado

Puntúa 0,00
sobre 3,00

🚩 Marcar
pregunta

Se tienen las clases Paciente y Enfermedad:

1. Indique qué debería considerar para que la clase Paciente se pueda clonar y desarrolle el método clone() para hacer una clonación correcta.
2. Suponga un método main() que tiene un arreglo de Pacientes. Se desea clonar el arreglo, indique qué debería contemplar para que esto sea posible y desarrolle el método clone() para el arreglo.

```
Paciente
+ String nombre;
+ int edad;
+ Enfermedad enfermedad;
```

```
Enfermedad
+int tipo
+String descripción
+boolean contagiosa
```

Valor del ejercicio: 3 puntos

Pregunta 11

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Qué patrón conviene utilizar para mejorar este diseño? La intención sería eliminar el uso de *instancias*

```
abstract class Trabajador {
    trabajar() {
        System.out.println("No hago nada!!");
    }
}

class Artesano extends Trabajador {
    trabajar() {
        System.out.println("Hice un duendecito!");
    }
}

class Zapatero extends Trabajador {
    trabajar() {
        System.out.println("Hice un mocasín!");
    }
}

class Artesano extends Trabajador {
    trabajar(Material material) {
        if (material instanceof Cuero) {
            return new Llaverero((Cuero) material);
        }
        else if (material instanceof Goma) {
            return new Juego((Goma) material);
        }
        else {
            throw new NoConozcoElMaterialException(material);
        }
    }
}
```

```
class Zapatero extends Trabajador {
    trabajar(Material material) {
        if (material instanceof Cuero) {
            return new ZapatoCuero((Cuero) material);
        }
        else if (material instanceof Goma) {
            return new Sandalia((Goma) material);
        }
        else {
            throw new NoConozcoElMaterialException(material);
        }
    }
}
```

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

Seleccione una:

- ☐ a. Decorator
- ☐ b. Double Dispatching
- ☐ c. Template
- ☐ d. Factory
- ☐ e. Singleton

Tiempo restante 0:22:11

Pregunta 12

Sin responder
aún

Puntúa como
3,00

🚩 Marcar
pregunta

En pocas palabras responda cómo podría identificar en un código el uso (correctamente implementado) del patrón Decorator dentro de un grupo de Clases relacionadas? Qué debería tener?

Valor del ejercicio: 3 puntos

↵	A ▼	B	<i>I</i>	≡	≡	≡	≡	🔗	🔄	🖼️
---	-----	---	----------	---	---	---	---	---	---	----

tiempo restante 0:10:51

Pregunta 14

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Las excepciones son el espacio de la solución a un problema ocurrido

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Tiempo restante 0:16:31

Pregunta 14

in responder
ún

untúa como
,00

> Marcar
pregunta

Las excepciones son el espacio de la solución a un problema ocurrido

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Tiempo restante 0:15:08

Pregunta 15

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Para implementar una interfaz, la clase debe proporcionar la implementación de todos los métodos en forma concreta (con cuerpo).

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. FALSO
- ☐ b. VERDADERO

Pregunta 16

Sin responder aún

Puntúa como 1,00

🚩 Marcar pregunta

Una clase Hija, para ejecutar cada método de la clase Padre está siempre obligada a anteponer la palabra "super".

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Pregunta 17

Sin responder
aún

Puntúa como
1,00

🚩 Marcar
pregunta

Sabiendo que la clase concreta Alumno se extiende de la clase abstracta Persona, analiza las siguientes sentencias y marca todas las opciones correctas:

Valor total de respuesta correcta: 1

Valor de cada respuesta incorrecta: -1/2

Seleccione una o más de una:

- ☐ a. `Alumno a_b= new Alumno();`
- ☐ b. `Persona p_b= new Alumno();`
- ☐ c. `Alumno a_a= new Persona();`
- ☐ d. Ninguna de las anteriores es correcta
- ☐ e. `Persona p_a= new Persona();`

Pregunta 18

Sin responder
aún

Puntúa como
3,00

🚩 Marcar
pregunta

Determine la salida correcta del siguiente programa.

```
public class A
{
    static int x[] = { 1, 2 };
    int y[];
    void f(int z[])
    {
        z[1] += 2;
    }
    void g()
    {
        A a = new A();
        a.x[0]++;
    }
}

public class B
{
    public static void main (String args[ ])
    {
        A b = new A ( );
        b.y = b.x;
        b.f (b.y);
        b.g ( );
        System.out.println (b.x[0] + " " + b.x[1]); //(1)
    }
}
```

Valor de respuesta correcta: 3 (relativo al resto de las preguntas)

Valor de respuesta incorrecta: resta el 50% del valor

- ☐ a. 2 2
- ☐ b. 2 4
- ☐ c. 1 4
- ☐ d. 1 2

Pregunta 19

Sin responder
aún

Puntúa como
3,00

🚩 Marcar
pregunta

¿Cómo es posible crear objetos de una clase cuyos constructores son todos privados?

Valor de respuesta correcta: 3 (relativo al resto de las preguntas)

Valor de respuesta incorrecta: resta la mitad del valor

- ☐ a. Definiendo una superclase con constructores públicos.
- ☐ b. No es posible.
- ☐ c. Definiendo un método estático público en la clase que cree un objeto de la clase y lo devuelva.
- ☐ d. Definiendo una subclase y declarando públicos los constructores heredados.