


Pregunta **1**
Correcta
Se puntúa 1,00
sobre 1,00
[🚩 Marcar
pregunta](#)

Una clase debe diseñarse de tal forma que esté abierta para su extensión

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. V 
- ☐ b. F

Respuesta correcta

La respuesta correcta es:

V

Pregunta **2**

Correcta

Se puntúa 1,00
sobre 1,00

🚩 Marcar
pregunta

Es posible propagar una excepción que sea de clase hija de la excepción declarada en la cláusula throws

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

Respuesta correcta

La respuesta correcta es:
VERDADERO

Pregunta **3**

Correcta

Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Las Interfaces pueden extenderse.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. VERDADERO ✓
- ☐ b. FALSO

Respuesta correcta

Las interfaces pueden extenderse para crear nuevas interfaces.

La respuesta correcta es:

VERDADERO

Pregunta **4**

Correcta


Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Las excepciones no pueden extenderse.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. V
- ☒ b. F 

Respuesta correcta

La respuesta correcta es:

F

Pregunta **5**

Correcta

Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Una excepción debe contener información necesaria para poder solucionar el problema (que la causó) dentro de la zona de recuperación.

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

Respuesta correcta

La respuesta correcta es:
VERDADERO

Pregunta **6**

Correcta


Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Una interface puede declarar métodos que lancen excepciones.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. V 
- ☐ b. F

Respuesta correcta

La respuesta correcta es:

V

Pregunta **7**

Finalizado

Sin calificar

✓ Marcar
pregunta

Suponiendo un conjunto de clases que conforman el patrón factory:

ClaseBase se extiende a

- ExtendidaTipo1
- ExtendidaTipo2
- ExtendidaTipo3

Y la clase BaseFactory con el siguiente método:

```
class BaseFactory
```

```
{
```

```
public ClaseBase getClaseBase(String tipo)
```

```
{
```

```
    ClaseBase respuesta=null;
```

```
    if(tipo.equalsIgnoreCase("tipo1")) respuesta= new ExtendidaTipo1();
```

```
    else if (tipo.equalsIgnoreCase("tipo2")) respuesta= new ExtendidaTipo2();
```

```
    else if (tipo.equalsIgnoreCase("tipo3")) respuesta= new ExtendidaTipo3();
```

```
    return respuesta;
```

```
}
```

```
}
```

Cómo podría ampliar la cantidad de clases que maneja la clase "Factory" si se agrega posteriormente la clase ExtendidaTipo4. Elija solamente la mejor opción que se le ocurra y escriba el código correspondiente.

Valor del ejercicio: 5 puntos

Hecho en hoja

Pregunta **8**

Correcta

Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Sea una jerarquía de clases: Clase Hijo extiende de Clase Padre, y se tiene la siguiente declaración:

```
Padre x = new Hijo();
```

Y la siguiente invocación.


```
x.metodo()
```

Determine verdadero o falso lo siguiente:

`metodo()` es cualquier método que pertenezca a Clase Hijo.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. F 
- ☐ b. V

Respuesta correcta

La respuesta correcta es:

F

Pregunta **9**

Correcta

Se puntúa 1,00
sobre 1,00

🚩 Marcar
pregunta

Un método de instancia puede invocar un método estático.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☒ b. V ✓

Respuesta correcta

La respuesta correcta es:

V


Pregunta **10**

Correcta

Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Si una clase padre es declarada ***cloneable*** sus clases hijas heredan esta condición

- ☐ a. FALSO
- ☒ b. VERDADERO 

Respuesta correcta

Es verdadero para que se cumple el principio de Liskov

La respuesta correcta es:

VERDADERO

Pregunta **11**

Incorrecta


Se puntúa -0,50
sobre 1,00

 [Marcar
pregunta](#)

Una interface es una herramienta de diseño que determina el comportamiento esperado por las clases que la implementen.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. FALSO 
- ☐ b. VERDADERO

Respuesta incorrecta.

La respuesta correcta es:
VERDADERO

Pregunta **12**

Correcta

Se puntúa 1,00
sobre 1,00

🚩 Marcar
pregunta

Señale la respuesta **CORRECTA**: En Java, si se tienen 2 métodos con el mismo nombre:

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. No pueden tener el mismo número de parámetros
- ☒ b. Se puede dar el caso de que tengan igual número de parámetros y de diferente tipo ✓
- ☐ c. Es posible que 2 métodos tengan los mismos parámetros y devuelvan el mismo tipo de resultado dentro de una misma clase
- ☐ d. Ninguna de las otras afirmaciones es cierta

Respuesta correcta

La respuesta correcta es:

Se puede dar el caso de que tengan igual número de parámetros y de diferente tipo

Pregunta **13**

Incorrecta


Se puntúa -0,50
sobre 1,00

 [Marcar
pregunta](#)

El patrón Singleton permite tener una única instancia de una clase a la que sólo puede acceder el primer método que la llama.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☒ b. V 

Respuesta incorrecta.

La respuesta correcta es:

F

Indique qué debería considerar para que la clase `Asist` se pueda clonar y devuelva el método `Clonar` para hacer una clonación correcta. Considere que la clase `Armaamento_Carros` no pueden ser clonados nunca y la clase `Armaamento_Misiles` sí se pueden clonar siempre.

Nota: Entre `Asist` y `Armaamento` hay una relación de COMPOSICIÓN. Entre `Asist` y `Disco` hay una relación de AGREGACIÓN.

sol:0

```
• String nombre;  
• float velocidad;  
• boolean misil;  
• Armaamento Armaamento;
```

Localidad

```
• int idAsistencia;  
• String nombre;  
• String descripcion;
```

Armaamento soporteMisil

```
• String descripcion;  
• float alcance;
```

Armaamento Misil

```
• double velocidadMaxima;  
• boolean ES;
```

Armaamento Disco

```
• int capacidad;
```

Valor de la pregunta: 8 puntos

Hacer un login

Le responde a esto es similar al ejercicio del Paciente con Informes abstracto:

```
public class Paciente implements Clinica {  
    String nombre;  
    int edad;  
    Informes informes();  
    public Paciente() {  
        name();  
    }  
    @Override  
    protected Object clone() throws CloneNotSupportedException {  
        {  
            Paciente p = new();  
            p = (Paciente) super.clone();  
            p.informes() = ((Informes) informes().clone());  
            return p;  
        }  
    }  
    public abstract class Informes implements Clinica {  
        String descripcion;  
        public Informes() {  
            name();  
        }  
        @Override  
        protected Object clone() throws CloneNotSupportedException {  
            return super.clone();  
        }  
    }  
    public class Informes_Fisica extends Informes implements Clinica {  
        double temperatura;  
        double presion;  
    }  
}
```

Algoritmo C++

```
    name();  
    }  
    @Override  
    protected Object clone() {  
        {  
            Informes_Fisica informes;  
            try {  
                if = ((Informes_Fisica)super.clone());  
            }  
            catch (CloneNotSupportedException e) {  
            }  
            return if;  
        }  
    }  
    public class Informes_Misiles extends Informes {  
        String informes;  
        public Informes_Misiles() {  
            name();  
        }  
        @Override  
        public Object clone() throws CloneNotSupportedException {  
            {  
                return new CloneNotSupportedException();  
            }  
        }  
    }  
}
```

Pregunta **15**

Correcta

Se puntúa 1,00
sobre 1,00

🚩 [Marcar
pregunta](#)

En el paradigma POO los objetos de una clase deberían poder ser reemplazados por instancias de una de sus subclases sin alterar el funcionamiento del programa.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☒ b. V ✓

Respuesta correcta

La respuesta correcta es:

V

Pregunta **16**

Correcta

Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Una Clase (no abstracta) puede extenderse en una Clase abstracta? O sea, la clase Padre no es abstracta y la clase hija es abstracta.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. V ✓
- ☐ b. F

Respuesta correcta

La respuesta correcta es:

V

Pregunta **17**

Correcta


Se puntúa 1,00
sobre 1,00

 [Marcar
pregunta](#)

Una clase final es aquella que no puede ser derivada, no puede tener subclases.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☒ a. V 
- ☐ b. F

Respuesta correcta

La respuesta correcta es:

V

Pregunta **18**

Sin contestar

Puntúa como
1,00

🚩 [Marcar
pregunta](#)

Un método (diferente de void) que propaga una excepción siempre retorna un valor antes de hacer la propagación.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. F
- ☐ b. V

Respuesta incorrecta.

La respuesta correcta es:

F

Pregunta **19**
Correcta
Se puntúa 1,00
sobre 1,00
[🚩 Marcar
pregunta](#)

Una clase abstracta debe tener solamente métodos abstractos.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- ☐ a. V
- ☒ b. F ✓

Respuesta correcta

La respuesta correcta es:

F

Pregunta **20**

Finalizado

Sin calificar

 Marcar pregunta

Considere el bosquejo del siguiente método

```
/**
 * retorna el índice de grasa corporal aproximado para personas mayores de "edad_limite_inferior" (parámetro).
 * precondiciones ....
 */
public double
calcular_indice_en_adultos(int edad_persona, int edad_limite_inferior, double peso, int altura) throws ....
{
    ...
}
```

Indique y justifique las precondiciones y/o excepciones que utilizaría. No es obligatorio utilizar ambas.

Establezca en los comentarios del contrato, el ámbito adecuado para invocar este método (esto está relacionado con las precondiciones).

hecho en hoja.