Comenzado el	jueves, 28 de abril de 2022, 11:00
Estado	Finalizado
Finalizado en	jueves, 28 de abril de 2022, 12:30
Tiempo empleado	1 hora 29 minutos
Calificación	Sin calificar aún

Pregunta **1**

Correcta

Se puntúa 1,00 sobre 1,00

Una Clase Abstracta puede tener constructores

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

Una variable de tipo interface permite acceder solamente a los métodos (del objeto referenciado) que están declarados en la interface.

```
Valor de respuesta correcta: 1
Valor de respuesta incorrecta: -1/2
```

a. V

b. F ✔

Respuesta correcta

Esta pregunta estuvo mal realizada, por lo tanto a todos se les considerará como válida.

Las respuestas correctas son:
V,

F

Pregunta **3**

Finalizado

Puntúa como 3,00

Indique qué debería considerar para que la clase Avión se pueda clonar y desarrolle el método Clone() para hacer una clonación correcta. Considerar que la clase Armamento_Canion no pueden ser clonadas nunca y la clase Armamento_Misil se pueden clonar siempre.

```
Avion
    +String Nombre;
    +Float Velocidad;
    +Armamentos Armamento;

Armamento <abstracta>
   +String Descripción;
   +Float Alcance;

Armamento_Misil
   +double Potencia_explosivo;
   +boolean ILS;

Armamento_Canion
   +int Cargador;
```

Valor de la pregunta: 3 puntos

Supongo que las clases Armamento_Canion y Armamento_Misil son clases concretas hijas de Armamento <abstracta> y éstas implementan los métodos que no hayan sido implementados en Armamento.

Como no puedo hacer un getClass(), reescribiría el método clone() de Object:

```
public Object clone()
{
   try
    {
        Avion nObj = (Avion)super.clone();
        ...
       return nObj
```

```
        }
    catch(CloneNotSupportedException e)
        {
            trow new InternalError(e.toString);
        }

}
```

Si el avión posee un armamento de tipo Armamento_Canion éste no se va a clonar y va a lanzar una excepción.

La respuesta a esto es similar al ejercicio del Paciente con Enfermedad abstracta:

```
public class Paciente implements Cloneable
{
    String nombre;
    int edad;
    Enfermedad enfermedad;
    public Paciente()
    {
        super();
    }

    @Override
    protected Object clone() throws CloneNotSupportedException
    {
        Paciente p = null;
        p = (Paciente) super.clone();
        p.enfermedad = (Enfermedad) enfermedad.clone();
        return p;
    }
}
public abstract class Enfermedad implements Cloneable
{
    String descripcion;

    public Enfermedad()
    {
        super();
    }

    @Override
    protected Object clone() throws CloneNotSupportedException
    {
        return super.clone();
    }
```

```
}
public class Enfermedad_Fisica extends Enfermedad implements Cloneable
{
    double temperatura;
    double presion;

    public Enfermedad_Fisica()
    {
        super();
    }

    @Override
    protected Object clone()
    {
        Enfermedad_Fisica ef=null;
        try
        {
            ef = (Enfermedad_Fisica)super.clone();
        }
        catch (CloneNotSupportedException e)
        {
        }
        return ef;
    }
}
public class Enfermedad_Psiquica extends Enfermedad
{
    String sintomas;

    public Enfermedad_Psiquica()
    {
        super();
    }
```

```
    @Override
    public Object clone() throws CloneNotSupportedException
    {
        throw new CloneNotSupportedException();
    }
}
```

Pregunta **4**

Correcta

Se puntúa 1,00 sobre 1,00

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. FALSO

b. VERDADERO ✔

Respuesta correcta

La respuesta correcta es:
VERDADERO

Pregunta **5**

Correcta

Se puntúa 1,00 sobre 1,00

Es posible atrapar una excepción que sea de tipo hija de una excepción declarada en un bloque catch(). Por eso es importante ordenar los bloques catch() según el tipo de excepción que atrape.

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**

Incorrecta

Se puntúa -0,50 sobre 1,00

Las Interfaces pueden extenderse.

```
Valor de respuesta correcta: 1
Valor de respuesta incorrecta: -1/2
```

- a. FALSO ✖
- b. VERDADERO

Respuesta incorrecta.

Las interfaces pueden extenderse para crear nuevas interfaces.

La respuesta correcta es:
VERDADERO

Pregunta **7**

Finalizado

Puntúa como 3,00

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).

Precondiciones:

* edad_persona > 0

* peso > 0

* altura > 0

A la hora de implementar el método, supongo que edad_limite_inferior es un valor mayor a cero.

Lanzaría una excepción a la hora de realizar el cálculo del índice si edad_limite_inferior es menor o igual a cero.

Pregunta **8**

Correcta

Se puntúa 1,00 sobre 1,00

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. FALSO

b. VERDADERO ✔

Respuesta correcta

La respuesta correcta es:
VERDADERO

Pregunta **9**

Correcta

Se puntúa 1,00 sobre 1,00

Los atributos privados no son accesibles desde el objeto instancia, pero sí son accesibles desde la misma clase y las clases hijas.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- a. F ✔
- b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **10**

Incorrecta

Se puntúa -0,50 sobre 1,00

Se dice que para que una clase sea abstracta debe contener al menos un método abstracto.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- a. F
- b. V ✖

Respuesta incorrecta.

La respuesta correcta es:
F

Pregunta **11**

Finalizado

Puntúa como 3,00

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

Primero las clases Paciente y Enfermedad deben implementar Cloneable.

Como dichas clases tienen una relación de composición, hay que hacer una clonación en cascada o profunda.

en Paciente:
* reescribo el método clon() para que sea Public

```
public Object clone() throws ClonNotSupportedException
{
   Paciente pacienteC = null;
   pacienteC = (Paciente)super.clone();
   pacienteC.enfermedad = (Enfermedad).this.enfermedad.clon();

   return paceinteC;
}
```

en Enfermedad:
*reescribo el método clon() para que sea Public

```
public Object clone() throws ClonNotSupportedException
{
   return super.clone();
}
```

en el main:

```
...
   Enfermedad enfermedad = new Enfermedad(.....);
   Paciente paciente1 = new Paciente(..... enfermedad);
   Pacinente paciente2 = null;

   try
   {
      paciente2 = (Paciente)paciente1.clone();
   }
   catch (CloneNotSupportedException e)
   {
      e.printStackTrace();
   }
....
```

Pregunta **12**

Correcta

Se puntúa 1,00 sobre 1,00

Cuál es el patrón que tiene como objetivo principal otorgar nuevo comportamiento en forma dinámica a objetos mediante la composición.

```
Valor de respuesta correcta: 1
Valor de respuesta incorrecta: -1/2
```

Seleccione una:

- a. Singleton
- b. Double dispatching
- c. Template
- d. Decorator ✔
- e. Factory

Respuesta correcta

Decorator

La respuesta correcta es: Decorator

Pregunta **13**

Correcta

Se puntúa 1,00 sobre 1,00

Java tiene pasaje de parámetros por valor para los parámetros de tipo primitivo y por referencia para los parámetros de tipo Object.

```
Valor de respuesta correcta: 1
Valor de respuesta incorrecta: -1/2
```

- a. F ✔
- b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **14**

Correcta

Se puntúa 1,00 sobre 1,00

Una Interface puede tener atributos de instancia pero sólo si son públicos.

```
Valor de respuesta correcta: 1
Valor de respuesta incorrecta: -1/2
```

- a. F ✔
- b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **15**

Correcta

Se puntúa 1,00 sobre 1,00

Una clase hija hereda los atributos, métodos y constructores de la clase padre

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- a. F ✔
- b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **16**

Incorrecta

Se puntúa -0,50 sobre 1,00

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

b. No pueden tener el mismo número de parámetros

c. Ninguna de las otras afirmaciones es cierta ✖

d. Es posible que 2 métodos tengan los mismos parámetros y devuelvan el mismo tipo de resultado dentro de una misma clase

Respuesta incorrecta.

La respuesta correcta es:
Se puede dar el caso de que tengan igual número de parámetros y de diferente tipo

Pregunta **17**

Correcta

Se puntúa 1,00 sobre 1,00

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

a. VERDADERO ✔

b. FALSO

Respuesta correcta

Es verdadero para que se cumple el principio de Liskov

La respuesta correcta es:
VERDADERO

Pregunta **18**

Finalizado

Puntúa como 3,00

Suponiendo un conjunto de clases que conforman el patrón factory:

1) cómo podría ampliar la cantidad de clases que maneja la clase "Factory" (la clase que construye a los objetos)? Elija solamente la mejor opción que se le ocurra.

2) en este caso, la única alternativa sería reescribir el código de la clase "Factory"?

Valor del ejercicio: 3 puntos

Una forma sería que ésta nueva clase se extienda de la clase "Factory" y luego realizar algo cómo:

```
class nuevaClase extends classFactory
{
   public variable getVariable(.....)

   {
      if(super().metodofactory(variable) != null)
          {
          //analizo si variable es válido
          }


   }


}
```

Pregunta **19**

Correcta

Se puntúa 1,00 sobre 1,00

Un método static puede acceder a un atributo de instancia pero solo si es public.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

a. F ✔

b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **20**

Correcta

Se puntúa 1,00 sobre 1,00

Si hay una relación de Agregación entre clases, la clonación debe ser Superficial

a. FALSO

b. VERDADERO ✔

Respuesta correcta

Verdadero

La respuesta correcta es:
VERDADERO

Pregunta **21**

Sin contestar

Puntúa como 1,00

Un atributo (no estático) "final" de tipo primitivo no puede inicializarse en un constructor.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

a. F

b. V

Respuesta incorrecta.

La respuesta correcta es:
F

Pregunta **22**

Correcta

Se puntúa 1,00 sobre 1,00

Cuando un método presenta una cláusula throws no puede tener bloques catch() en su interior

- a. F ✔
- b. V

Respuesta correcta

La respuesta correcta es:
F

Pregunta **23**

Incorrecta

Se puntúa -0,50 sobre 1,00

Las interfaces no modelan comportamiento.

Valor de respuesta correcta: 1

Valor de respuesta incorrecta: -1/2

- a. V ✖
- b. F

Respuesta incorrecta.

La respuesta correcta es:
F

◄ Encuesta sobre conocimientos previos

Ir a...