

Estimada comunidad universitaria:  
El próximo sábado 15 de julio, a partir de las 9hs y hasta las 14hs, Webasignatura se encontrará en mantenimiento.  
Durante ese periodo de tiempo Webasignatura no será accesible.  
Ante cualquier consulta, por favor comunicarse con Soporte a Usuarios.

**Comenzado el** jueves, 8 de junio de 2023, 18:40

**Estado** Finalizado

**Finalizado en** jueves, 8 de junio de 2023, 18:49

**Tiempo empleado** 9 minutos 3 segundos

**Puntos** 13,00/15,00

**Calificación** 8,67 de 10,00 (86,67%)

Pregunta **1**  
Correcta  
Se puntúa 1,00 sobre 1,00

- ¿Cuáles de los siguientes son síntomas que podemos estar ante un antipatron Spaghetti Code?
- i. Las clases son nombreadas como procesos del sistema y sus métodos son enormes
  - ii. El acceso a las variables miembro de una clase es mediante funciones de acceso
  - iii. Los métodos que utilizan pocos parámetros y se utilizan muchas clases y variables globales
  - iv. Bloques grandes de condicionales anidados

Seleccione la respuesta correcta:

- ☐ a. i, ii, iii
- ☐ b. i, ii, iii, iv
- ☒ c. i, iii, iv
- ☐ d. i, ii, iv



Respuesta correcta  
La respuesta correcta es: i, iii, iv



Pregunta **2**

Correcta

Se puntúa 1,00 sobre  
1,00

El patrón Singleton permite

- ☐ a. Especificar el tipo de Objetos que se crearán
- ☒ b. Limitar la Creacion de objetos pertenecientes a una clase
- ☐ c. Permite al usuario añadir nuevas funcionalidades a un objeto existente sin alterar su estructura
- ☐ d. Crear nuevos objetos clonando una instancia creada previamente



Respuesta correcta

La respuesta correcta es:

Limitar la Creacion de objetos pertenecientes a una clase

Pregunta 3

Correcta

Se puntúa 1,00 sobre 1,00

El siguiente fragmento de código:

...

```
class Coche {  
    String marca;  
    Coche(String marca){ this.marca = marca; }  
    String getMarcaCoche(){ return marca; }  
}  
public static void main(String[] args) {  
    Coche[] arrayCoches = {  
        new Coche("Renault"),  
        new Coche("Audi")  
    };  
    imprimirPrecioMedioCoche(arrayCoches);  
}  
  
public static void imprimirPrecioMedioCoche(Coche[] arrayCoches){  
    for (Coche coche : arrayCoches) {  
        if(coche.marca.equals("Renault")) System.out.println(18000);  
        if(coche.marca.equals("Audi")) System.out.println(25000);  
    }  
}
```

¿Cuál principio de los SOLID no estaría cumpliendo?

- ☐ a. SRP, porque *imprimirPrecioMedioCoche* tiene más de una responsabilidad
- ☐ b. Todas las anteriores
- ☒ c. OCP, porque en caso de añadir una nueva marca, tendríamos que modificar el método *imprimirPrecioMedioCoche*
- ☐ d. Bajo Acoplamiento, porque el método *imprimirPrecioMedioCoche* está muy acoplado



Respuesta correcta

La respuesta correcta es:

OCP, porque en caso de añadir una nueva marca, tendríamos que modificar el método *imprimirPrecioMedioCoche*

Pregunta 4

Correcta

Se puntúa 1,00 sobre 1,00

El siguiente fragmento de código:

...

```
abstract class Coche {  
    // ...  
    abstract int precioMedioCoche();  
}  
class Renault extends Coche {  
    @Override  
    int precioMedioCoche() { return 18000; }  
}  
class Audi extends Coche {  
    @Override  
    int precioMedioCoche() { return 25000; }  
}  
class Mercedes extends Coche {  
    @Override  
    int precioMedioCoche() { return 27000; }  
}  
Coche[] arrayCoches = {  
    new Renault(),  
    new Audi(),  
    new Mercedes(),  
    new Ford()  
};  
public static void imprimirNumAsientos(Coche[] arrayCoches){  
    for (Coche coche : arrayCoches) {  
        if(coche instanceof Renault)  
            System.out.println(numAsientosRenault(coche));  
        if(coche instanceof Audi)  
            System.out.println(numAsientosAudi(coche));  
        if(coche instanceof Mercedes)  
            System.out.println(numAsientosMercedes(coche));  
        if(coche instanceof Ford)  
            System.out.println(numAsientosFord(coche));  
    }  
}
```

```
imprimirNumAsientos(arrayCoches);
```

...

¿Cuál principio de los principios SOLID no se estaría cumpliendo?

- ☐ a. Todas las anteriores
- ☒ b. LSP, el programa debe conocer cada tipo de Coche y al llamar a su método *imprimirNumAsientos* asociado
- ☐ c. SRP, porque *imprimirPrecioMedioCoche* tiene más de una responsabilidad
- ☐ d. ISP, porque Mercedes, Audi y Renault, se pueden resumir en una interfaz



Respuesta correcta

La respuesta correcta es:

LSP, el programa debe conocer cada tipo de Coche y al llamar a su método *imprimirNumAsientos* asociado

Pregunta **5**

Correcta

Se puntúa 1,00 sobre 1,00

*“Los módulos de alto nivel no deberían depender de módulos de bajo nivel ambos deben depender de abstracciones”*

Con qué término se puede asociar esta oración:

- ☐ a. Alta cohesión, bajo acoplamiento
- ☐ b. Principio de Sustitución de Liskov (LSP)
- ☐ c. Principio de Responsabilidad Única (SRP)
- ☒ d. Principio de Inversión de Dependencias (DIP)



Respuesta correcta

La respuesta correcta es: Principio de Inversión de Dependencias (DIP)

## Pregunta 6

Correcta

Se puntúa 1,00 sobre 1,00

*“... los clientes no deberían verse forzados a depender de interfaces que no usan ...”*

Con qué término se puede asociar esta oración:

- ☐ a. Principio abierto/cerrado (SRP)
- ☐ b. Principio de Inversión de Dependencias (DIP)
- ☐ c. Principio de Sustitución de Liskov (LSP)
- ☒ d. Principio de Segregación de Interfaz (ISP)



Respuesta correcta

La respuesta correcta es: Principio de Segregación de Interfaz (ISP)

## Pregunta 7

Incorrecta

Se puntúa 0,00 sobre 1,00

Cuál de las siguientes es una típica causa para el antipatrón Spaghetti Code:

- ☐ a. Desarrolladores trabajando aisladamente en la solución
- ☐ b. Falta de uso de cualquier tipo de arquitectura
- ☐ c. Componentes reusables que una vez creados, no están lo suficientemente documentados para su uso
- ☒ d. Todas las anteriores



Respuesta incorrecta.

La respuesta correcta es: Desarrolladores trabajando aisladamente en la solución

## Pregunta 8

Correcta

Se puntúa 1,00 sobre 1,00

*"... representa un desvío en la forma de reutilizar el código haciendo que el mantenimiento sea una pesadilla ..."*

A que antipatrón se refiere este fragmento:

- ☐ a. Spaghetti Code
- ☐ b. The Blob
- ☐ c. Golden Hammer
- ☒ d. Cut-and-Paste Programming



Respuesta correcta

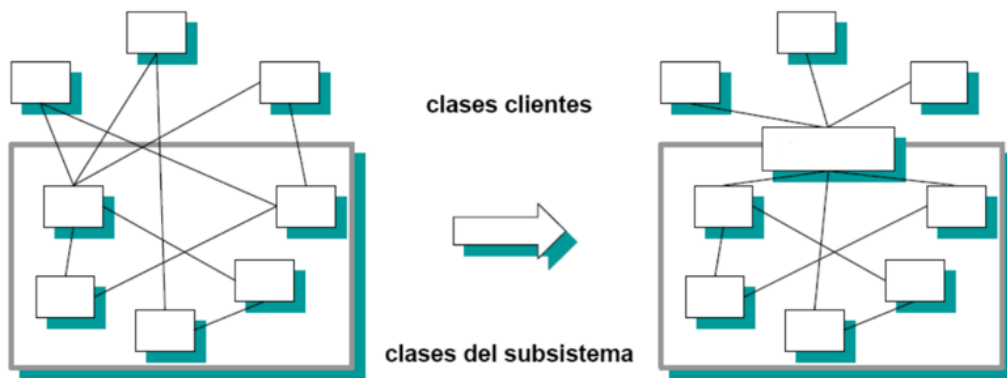
La respuesta correcta es: Cut-and-Paste Programming

## Pregunta 9

Correcta

Se puntúa 1,00 sobre 1,00

Dada la siguiente imagen



¿A qué patrón hace referencia?

- ☐ a. Singleton
- ☐ b. Factory Method
- ☒ c. Facade
- ☐ d. Observer



Respuesta correcta

La respuesta correcta es:  
Facade

## Pregunta 10

Incorrecta

Se puntúa 0,00 sobre 1,00

Que opción es falsa respecto al patrón observer

- ☐ a. El problema que soluciona es que tenemos un sistema en el cual existe un alto acoplamiento. Un objeto de una capa se comunica con varios de otra capa.
- ☐ b. Un objeto tiene que ser capaz de notificar a otros objetos sin saber nada mas de esos objetos.
- ☐ c. Se da en sistemas interactivos de monitorización de datos.
- ☒ d. Define una dependencia uno a muchos entre objetos.



Respuesta incorrecta.

La respuesta correcta es:

El problema que soluciona es que tenemos un sistema en el cual existe un alto acoplamiento. Un objeto de una capa se comunica con varios de otra capa.

## Pregunta 11

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuál de los siguientes anti-patrones se refiere a aquellos componentes que se encuentran implementados en el sistema, que no tienen uso actual y actualmente se encuentran allí por enfoques anteriores de otros desarrolladores?

- ☐ a. Golden Hammer
- ☒ b. Ninguna de las anteriores
- ☐ c. Spaghetti Code
- ☐ d. The Blob



Respuesta correcta

La respuesta correcta es:

Ninguna de las anteriores



## Pregunta 12

Correcta

Se puntúa 1,00 sobre 1,00

En el antipatrón Golden Hammer tenemos las siguientes características:

- ☒ a. Todas las anteriores ✓
- ☐ b. Un ejemplo, para este patrón puede ser el constante uso del mismo lenguaje de programación para cada proyecto sea o no conveniente.
- ☐ c. Se trata del uso obsesivo de una herramienta y de la terquedad de los desarrolladores para usar un paradigma de solución.
- ☐ d. : Es un vicio relacionado con aferrarse a un paradigma, para solucionar todos los problemas que se nos presenten al desarrollar sistemas.

Respuesta correcta

La respuesta correcta es:  
Todas las anteriores

## Pregunta 13

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuáles de los siguientes son síntomas que podemos estar ante un antipatrón The Blob?

- i. Clases con más de 60 atributos y operaciones.
- ii. Un diseño legado que no ha sido reprogramado correctamente en una arquitectura orientada a objetos
- iii. Nuevos desarrollos dependen demasiado de una herramienta
- iv. Clase de tipo "Blob" puede ser resultar costoso de cargar en memoria

Seleccione la respuesta correcta:

- ☒ a. i,ii,iv ✓
- ☐ b. Todos los síntomas son correctos
- ☐ c. i,ii,iii
- ☐ d. ii,iii,iv

Respuesta correcta

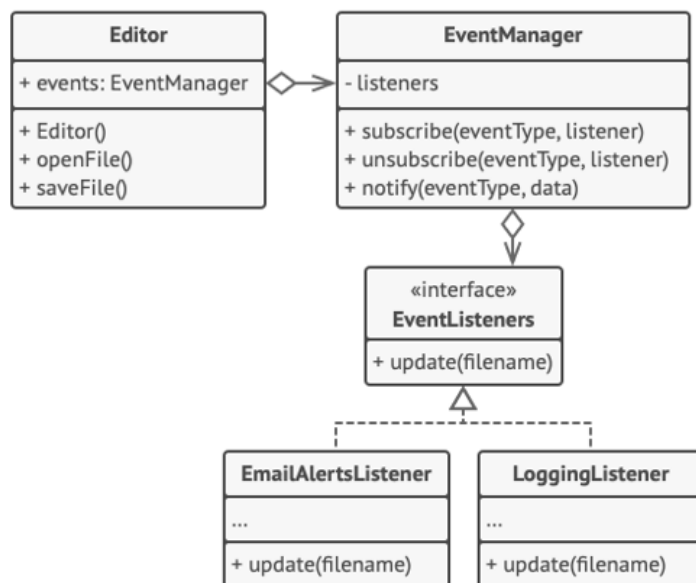
La respuesta correcta es: i,ii,iv

Pregunta **14**

Correcta

Se puntúa 1,00 sobre  
1,00

Dada la siguiente imagen:



- ☐ a. Visitor
- ☒ b. Observer
- ☐ c. Decorator
- ☐ d. Mediator



BLACKBOX AI

Respuesta correcta

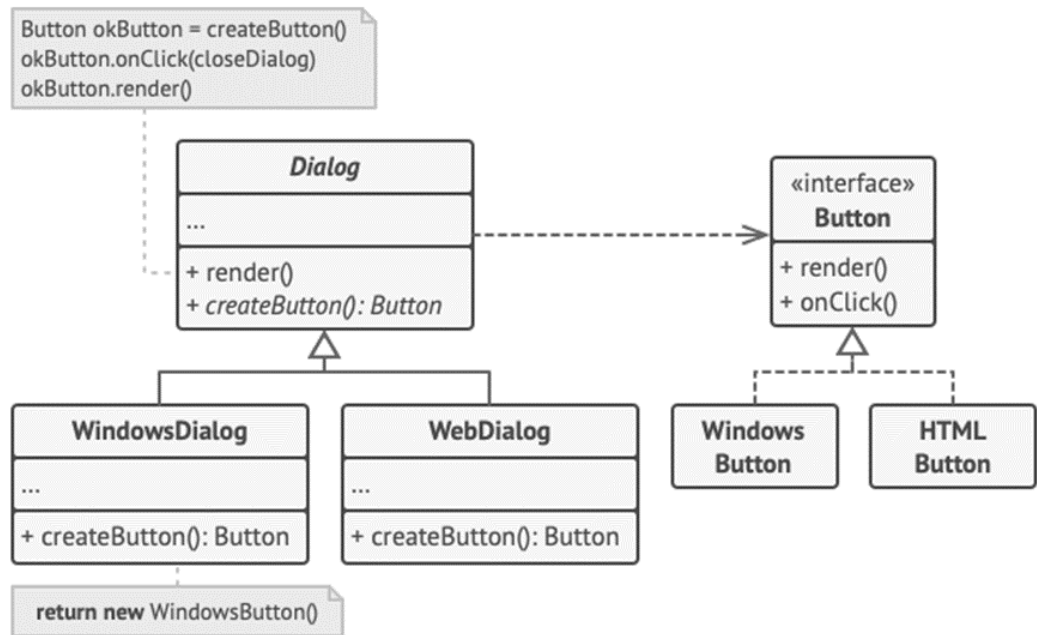
La respuesta correcta es:  
Observer

Pregunta **15**

Correcta

Se puntúa 1,00 sobre  
1,00

Dada la siguiente imagen:



- ☐ a. Bridge
- ☐ b. Builder
- ☐ c. Chain of Responsibility
- ☒ d. Factory method



Respuesta correcta

La respuesta correcta es:

Factory method

Actividad previa

Ir a...

Próxima actividad