



Has completado este test el *06/06/2022, 17:05*
Tu calificación es 70.00%

INCORRECTO

Ordenados son uno de los principios base a la hora de desarrollar soluciones software

✗ *Simple, (Keep)*

✗ *Keep (It)*

✗ *It (Simple,)*

✓ *Stupid*

CORRECTO

El principio Kiss fue acuñado por primera vez en los años 70s por los desarrolladores de software Gene Simmons y Paul Stanley

verdadero

✓ *falso*

INCORRECTO

El principio Kiss aplicado al software nos invita a:

Desarrollar componentes que fallen y puedan ser arreglados constantemente

✓ *Algoritmos intuitivos y fáciles de leer por otros desarrolladores*

Diseño de sistemas que no toleren fallos una vez se encuentran en producción

La complejidad del software debe ser probada en cada caso de prueba

✓ *Soluciones sencillas para resolver los problemas actuales pero adaptables a los cambios*

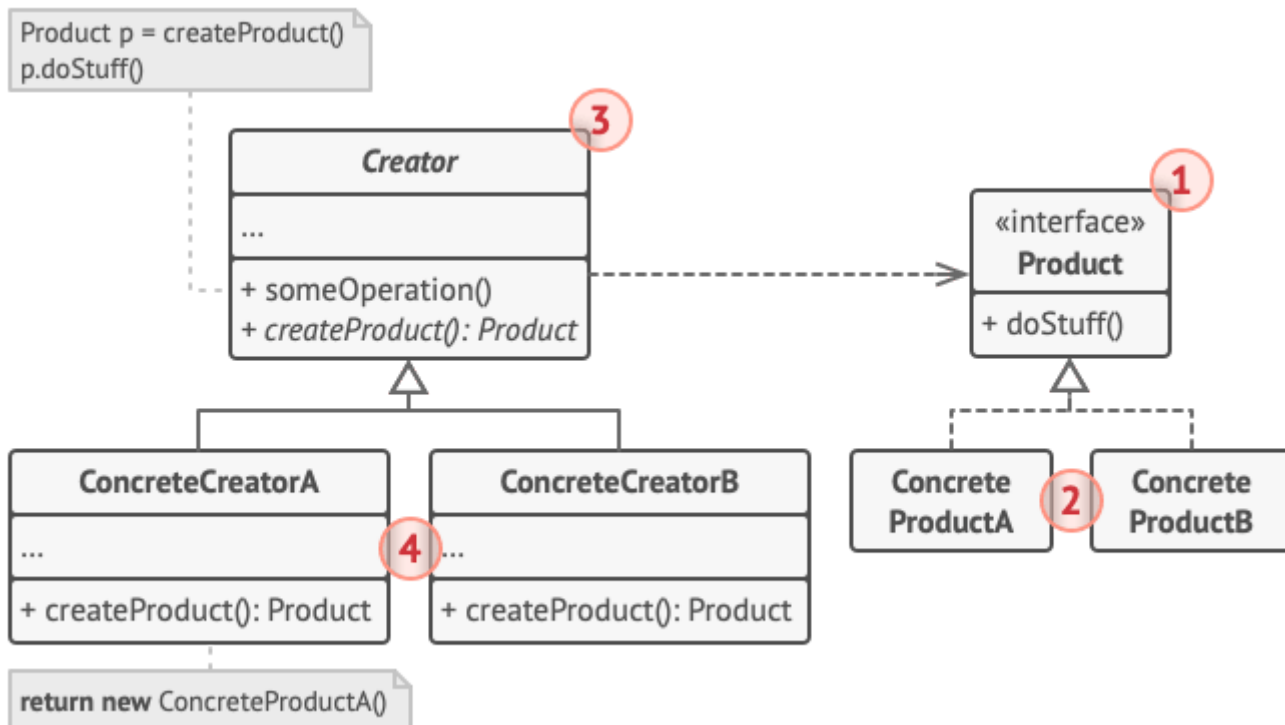
CORRECTO

Relacione los conceptos del paradigma orientado a objetos con sus correspondientes descripciones

✓ Extensibilidad	✓ El software debe permitir agregar nuevas funcionalidades sin dañar el sistema actual
✓ Responsabilidad	✓ Una clase debe realizar únicamente las tareas para la cual fue creada
✓ Reutilización	✓ Desarrollo de nuevas funcionalidades a partir de componentes previamente creados
✓ Paradigma	✓ Conjunto de reglas, normas o recomendaciones que se siguen en una disciplina o área determinada del conocimiento
✓ Encapsulamiento	✓ Ocultar el estado y comportamiento de objetos bajo una serie de reglas que permita la interacción controlada con otros objetos

CORRECTO

El patrón de diseño Factory Method...



...restringe la creación de instancias de una clase a un único objeto

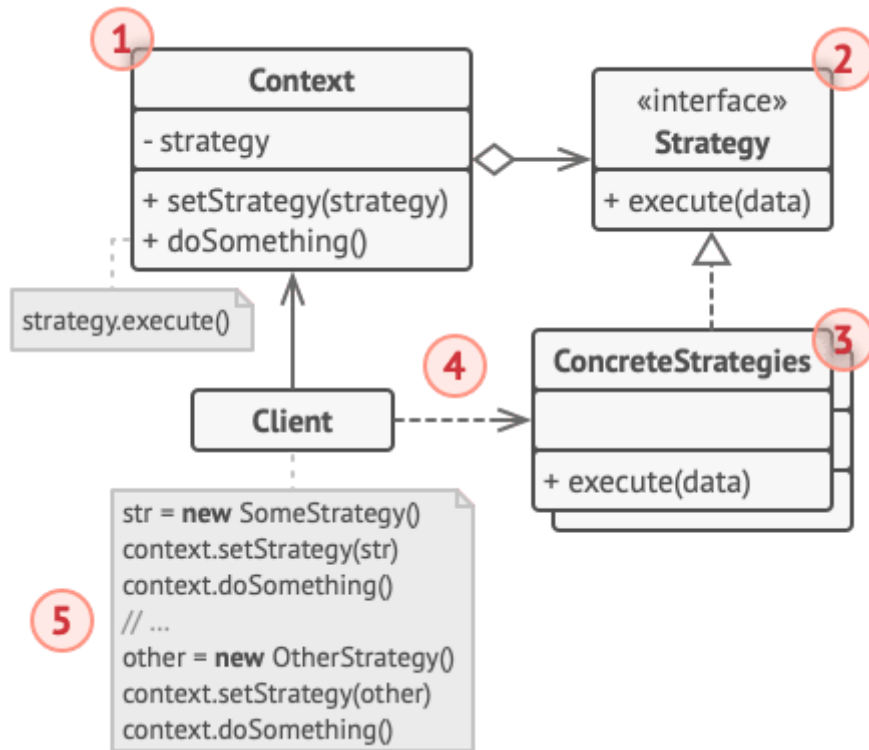
...permite crear interfaces comunes a partir de objeto determinado

✓ **...permite crear objetos de un subtipo determinado a partir de una interfaz común**

...proporciona una interfaz simplificada, un marco o cualquier otro conjunto complejo de clases

CORRECTO

El patrón de diseño **Strategy...**



...define una interfaz para crea conjuntos o familias de objetos relacionados sin especificar el nombre de la clase

...define una superclase con comportamientos comunes que serán implementados por clases específicas

...define una clase para crea conjuntos o familias de objetos relacionados sin especificar el nombre de la clase

- ✓ **...define una interfaz con comportamientos comunes que serán implementados por clases concretas (intercambiables)**

CORRECTO

Relaciones los principios SOLID con su correspondiente definición

✓ Dependency Inversion	✓ Los componentes de alto nivel no deben depender de los componentes de bajo nivel
✓ Liskov Substitution	✓ Una subclase puede ser sustituible por su superclase sin presentarse fallos o comportamientos inesperados
✓ Open/Close	✓ Los componentes software deben estar abiertos para su extensión pero cerrados para su modificación
✓ Interface Segregation	✓ Los clientes de una interfaz no deben verse forzados a usar comportamientos que no usan

✓ **Singles Responsibility**

✓ **Una clase o componente debe ser responsable de una sola cosa dentro de las funcionalidades de un sistema**

INCORRECTO

Cuando hablamos de cohesión y acoplamiento nos referimos a:

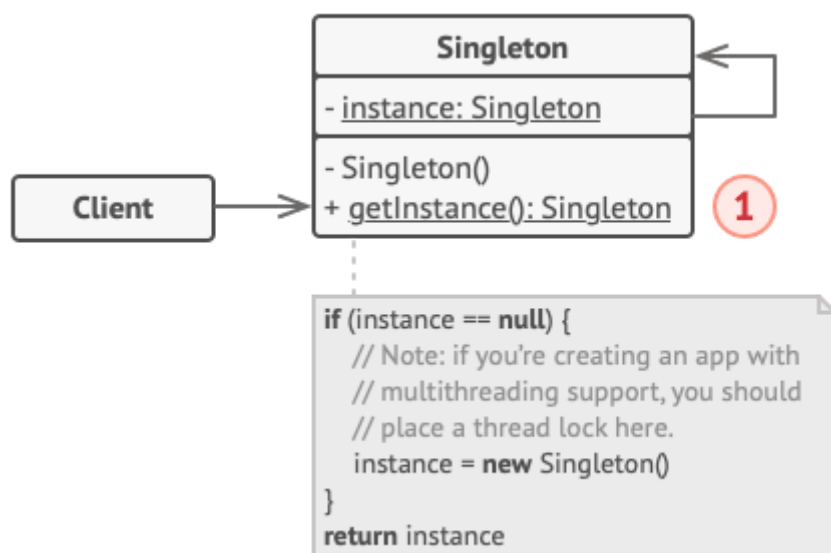
Cohesión: grado de interdependencia que tienen dos unidades de software entre sí

Acoplamiento: grado de interdependencia que tienen dos unidades de software entre sí

- ✓ **Cohesión: grado en que elementos diferentes de un sistema permanecen unidos para alcanzar un mejor resultado**
- ✗ **Acoplamiento: grado en que elementos diferentes de un sistema permanecen unidos para alcanzar un mejor resultado**

CORRECTO

El patrón de diseño Singleton...



...restringe la creación de objetos a un numero determinado de instancias de clase

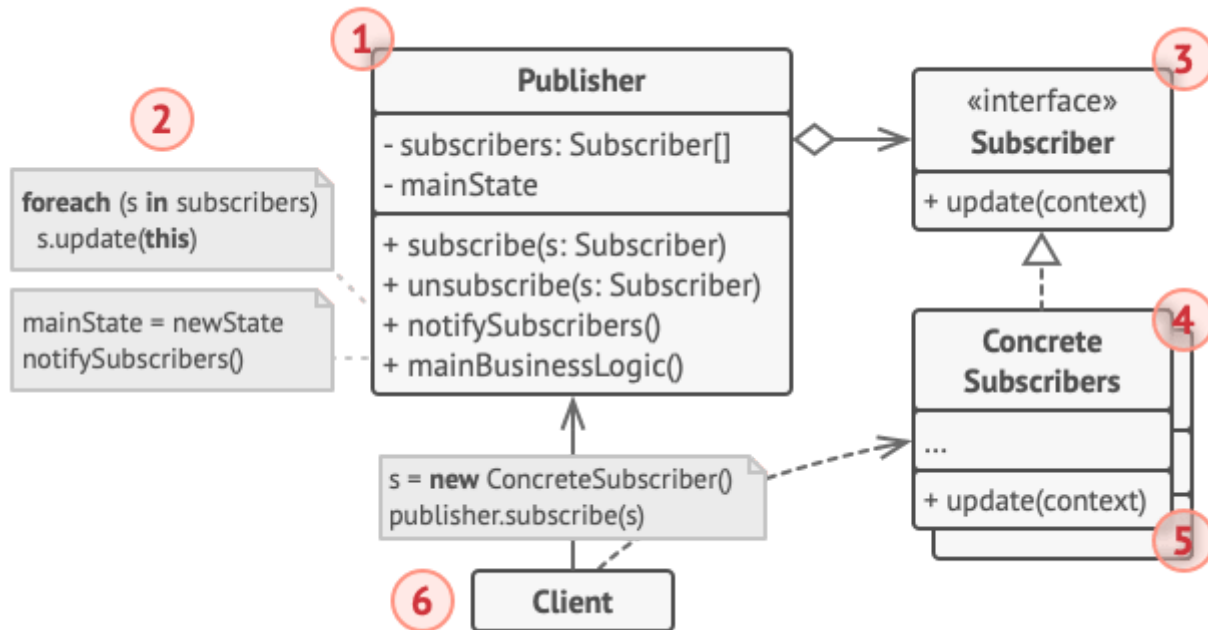
- ✓ **...restringe la creación de instancias de una clase a un único objeto**

...permite producir diferentes tipos y representaciones de un objeto utilizando el diferentes códigos de construcción

...permite producir diferentes tipos y representaciones de una interfaz utilizando el mismo código de construcción

CORRECTO

El patrón de diseño Observer...



- ✓ **...define un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que está siendo observado**

...define el comportamiento de una clase como variable con su estado y, por lo tanto, está representado por el objeto de contexto

...define un mecanismo de inscripción para notificar a varias interfaces sobre cualquier evento que le suceda al objeto que está siendo observado

...define el comportamiento de una objeto como variable con su estado y, por lo tanto, está representado por la interfaz de contexto