

Parcial II
Programación Orientada a Objetos
29/11/2021

Parte I

1. Desarrolle los siguientes conceptos: SRP, OCP, LSP, ISP, DIP.
2. Observe los siguientes casos e indique si cumplen o no con SOLID y qué debería hacerse para que cumpla con éste.

Caso 1	<pre>interface IPost { void CreatePost(); } interface IPostNew { void CreatePost(); void ReadPost(); }</pre>
Caso 2 Deseo agregar el atributo color	<pre>public class Car { private String brand; private String model; //Constructors, getters & setters }</pre>
Caso 3	<pre>public class Engine { // stuff } public class Car { private Engine engine; public Car(Engine xEngine) { this.engine = xEngine; } }</pre>

Parte II

En una tienda se desea avisar cuando el stock de ciertos productos es repuesto según las necesidades de cada cliente frecuente.

Se sabe que existe una lista para clientes frecuentes de alimentos procesados y otra para alimentos orgánicos.

Cuando un alimento es repuesto por un reponedor (agregando un elemento a una lista), dependiendo de qué tipo sea el alimento, sus consumidores frecuentes serán notificados, a través de una clase notificadora que debe tener una única instancia.

Los clientes deberían ser capaces de remover elementos de la lista; esta acción puede no ser notificada, pero si es necesaria la implementación de dicha funcionalidad, donde una excepción debe ser lanzada si se intenta extraer un elemento cualquiera de la lista de productos vacía.

Imagine que, una vez terminado el programa, debe agregar un nuevo tipo de alimento “elaborado”. Haga los cambios necesarios para que ésto sea posible y también pueda notificar a clientes frecuentes de dicha lista.

Justifique decisiones de diseño.

No es necesario agregar de forma dinámica los datos, se evaluará el correcto uso de patrones de diseño, principios SOLID y control de excepciones.