Taller Principios SRP – OCP – LCP

(Single Responsibility Principle / Open Closed Principle / Liskov Principle)

Objetivos:

- Identificar incumplimiento a los principios SRP OCP LCP.
- Corregir código que incumpla los principios SRP OCP LCP.
- Utilizar Github como herramienta colaborativa para cargar los proyectos.
- Lengua de programación JAVA

Instrucciones

- En taller se debe realizar individualmente.
- Usted trabajará con el proyecto en un repositorio de Github. El proyecto está compuesto por 6 paquetes (2 por cada principio, "Wrong" y "Good").
- Descargue el proyecto desde el repositorio llamado: https://github.com/neortiz1
- Los paquetes que contienen 'mal' en su nombre, contiene clases e interfaces que incumplen un principio. Las tres primeras letras del nombre del paquete le indican el principio que se está incumpliendo.
- Analice el código y reestructure las clases (Refactorizar), de modo que el principio SOLID se esté cumpliendo. Agregue su propuesta de la solución en los sub-paquetes 'Good' respectivos.

Conteste

Explique brevemente como cada principio se incumplió en el código analizado.

- SRP: La clase Vehicle tomaba todos los métodos, tenía más funciones de las que debería y que deben ir en otra clase.
- OCP: La clase EventHandler estaba cerrado a extensiones y abierto a modificaciones, debería tener una clase abtracta o interface que cree métodos que permitan la modificación de los atributos de cada tipo de vehivulo y cada uno tener una clase que acceda a estos métodos para asignarles un valor único a sus atributos.
- LSP: Las subclases no sustituían a la clase padre. Debería haber una clase para las aves que vuelan que heredan la clase padre Bird en la cual solo existe el método eat() mientras que en está nueva clase se añade un método mas que es fly() de esta forma la clase pato puede heredar los métodos de la clase de aves que vuelan para tener ambos métodos y la clase avestruz hereda solo el método de la clase Bird.

Entregable

https://github.com/elascano/ESPE2210-OOPSW7996