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: En este apartado, pudimos observar que hay dos secciones, el de máxima gasolina, y gasolina restante, por lo que, para aplicar el srp se crea una nueva clase fuel pump para administrar la gasolina en una parte distinta del vehículo.
* OCP: Event Handler cambia su como de manejo, se debe abrir sus paquetes por extensión, siendo así lo que se debe modificar.
* LSP: Aquí podemos que la Avestruz no puede volar, por lo que sería erróneo mantener con los métodos de volar y comer, lo que sí hace es comer, siendo ello lo que se debe modificar, creando la clase Ostrich, para seguirlo manejando sin perder el sentido del programa.

# Entregable

* Enlace al repositorio de GitHub donde colocaron su solución a cada principio y este archivo Word.

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

ws – Barrionuevo – unit 2 – ws26