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 vehículo contiene el método reFuel que no corresponde la responsabilidad de un vehículo, por ende no cumple con el SRP.
* OCP: Cuando queramos agregar un nuevo modo de manejo para el vehículo tendremos que cambiar todo el código de la clase eventHandler, esto rompe el OCP, lo correcto seria tener diferentes clases correspondientes a cada modo de manejo que hereden métodos de una clase padre llamada modo de manejo.
* LSP: La clase padre pájaro contiene un método volar y esta lo heredan un pato, que si puede volar, y una avestruz que no tiene la capacidad de volar, por lo tanto esta mal aplicado el LSP ya que la clase avestruz no puede comportarse como su clase padre

# Entregable

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