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 la clase Vehicle se encuentran dos responsabilidades la que esta sobrando ahí, es “reFuel” ya que eso debería ser una responsabilidad aparte es por ellos que se debe hacer una clase aparte “reFuel” para realizarlo esa responsabilidad y así cumplir la parte de este principio el cual es tener una única responsabilidad
* OCP: Aquí no podemos modificar nada del código que ya se encuentra hecho lo que podemos hacer es añadir una clase main para imprimir los valores de power y suspesionheingth de acuerdo a cada modo de manejo de esa manera cumplimos el principio no modificamos nada solo añadimos un método para imprimir los datos de cada modo de conducción.
* LSP: Aquí faltaba imprimir si la avestruz volaba, pero ella no vuela entonces solo modificamos el código de la clase de la avestruz la cual es una subclase de la superclase bird es decir cumplimos herencia, entonces si podemos modificar esa clase de ese modo ya podemos mandar al main a que nos imprimi el msj de que no puede volar sin alterar el programa de ese modo cumplimos este principio

# Entregable

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