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: El vehículo también se encargaba de regargar la gasolina. Pero de esto se debería pertenecer a otra clase como una gasolinera.
* OCP: Existía una clase con un método para cambiar a otros tipos en un mismo método.
* Pero lo que se debería haber hecho es hacerle a esta clase abstracta de manera que herede estos métodos a sus clases hijas y estas puedan reutilizar el código.
* LSP: La clase Ostrich no puede ser usada como un Bird ya que no puede heredar todos los metodos de esta clase Padre, esto se debe a que las Ostrichs no vuelan.

Por lo cual se deberia crear otra clase hija que a su vez sea padre de Ostrich y pueda heredar todos sus metodos. Esta es la clase FlyingBird.

# Entregable

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