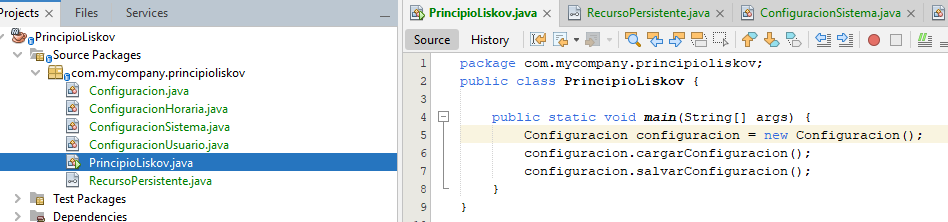
1. Crea un programa principal que ejecute los métodos de la clase Configuración.



2. Cumple la clase Configuración en Principio OCP. Justifica la respuesta.

RTA: La clase Configuración no cumple completamente el Principio OCP. Si se añade una nueva clase que implementa la interfaz RecursoPersistente, por ejemplo, una clase para cargar la configuración de red, entonces se tendría que modificar la clase Configuración para agregar la nueva clase y llamar a su método correspondiente. Esto viola el principio de "cerrado para modificación, abierto para extensión".

3. Cumple la clase Configuración el Principio de Liskov. Justifica la respuesta.

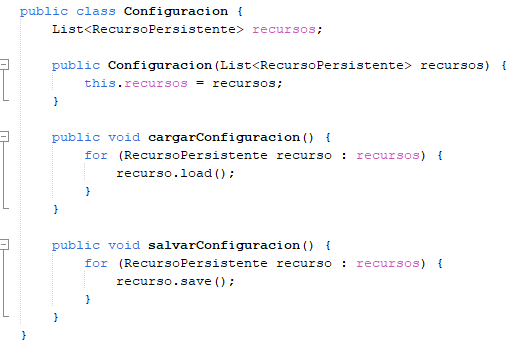
RTA: La clase Configuración no tiene una relación de herencia, por lo que no se aplica el Principio de Liskov.

4. Refactoriza la aplicación para que cumpla el principio de Liskov. La solución a

este ejercicio lo puedes encontrar en: https://lassala.net/2010/11/04/a-good-

example-of-liskov-substitution-principle/

RTA: La solución para cumplir el Principio de Liskov sería utilizar el principio de inversión de dependencias y la inyección de dependencias. En lugar de que la clase Configuración cree las instancias de las configuraciones y llame a sus métodos directamente, se puede hacer que la clase Configuración reciba las instancias de las configuraciones a través de su constructor o de un método de configuración. De esta forma, se pueden agregar nuevas configuraciones sin modificar la clase Configuración



5. Explica de forma general (independientemente del ejemplo) cual es el

problema y la solución propuesta.

RTA: El problema que se aborda con el Principio de Liskov es que una clase derivada debe ser sustituible por su clase base sin afectar el correcto funcionamiento del programa. Esto significa que si una clase A es una subclase de B, entonces cualquier instancia de A debe poder ser utilizada en cualquier lugar donde se espera una instancia de B, sin que esto cause errores o comportamientos inesperados. La solución propuesta para cumplir este principio es asegurarse de que las clases derivadas respeten el contrato de la clase base, y evitar agregar restricciones innecesarias en las clases derivadas que limiten su capacidad de ser utilizadas como sus clases base.