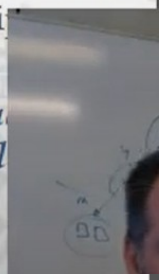
Principio de sustitución de Liskov

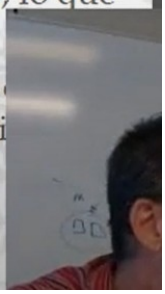
5.5.1. Principio de Sustitución de Liskov

- Definido por **Robert Martin** (*Liskov's Sustitution Principle - **LSP***) como uno de los principios SOLID
  - Está inspirado en los trabajos de **Barbara Liskov**: *"Lo que se quiere aquí es algo como la siguiente propiedad de sustitución: si para cada objeto o_1 de un tipo S, hay un objeto o_2 de tipo T tal que para todo progama P definido en términos de T, el comportamiento de P no cambia cuando o_1 es sustituido por o_2, entonces S es un subtipo de T"*
  - Se cumple sólo cuando los tipos de derivados son totalmente sustituibles por sus tipos base de forma que las funciones que utilizan estos tipos base pueden ser reutilizados con impunidad y los ti derivados se puede cambiar con impunidad.
  - ***El Principio de Sustitución de Liskov dice que las funciones qu punteros o referencias a una clase base debe ser capaz de usar l objetos de las clases derivadas sin conocerlas.***



5.5.1. Principio de Sustitución de Liskov

- Por tanto, la relación de herencia se refiere al comportamiento. No al comportamiento privado intrínseco sino al comportamiento público extrínseco del que dependen los clientes
  - ***El Principio de Sustitución de Liskov dice que se cumple cuando se redefine un método en una derivada reemplazando su precondición por una más débil y su postcondicion por una más fuerte***
    - La precondición de un subtipo es creada combinando con el operador OR las precondiciones del tipo base y del subtipo, lo que resulta una precondición menos restrictiva.
    - La postcondición de un subtipo es creada combinando con operador AND las postcondiciones del tipo base y del subti que resulta una postcondición más restrictiva.



5.5.1. Principio de Sustitución de Liskov

- Definido por **Robert Martin** (*Liskov's Sustitution Principle - **LSP***) como uno de los principios SOLID
  - Está inspirado en los trabajos de **Barbara Liskov**: *"Lo que se quiere aquí es algo como la siguiente propiedad de sustitución: si para cada objeto o_1 de un tipo S, hay un objeto o_2 de tipo T tal que para todo progama P definido en términos de T, el comportamiento de P no cambia cuando o_1 es sustituido por o_2, entonces S es un subtipo de T"*
  - Se cumple sólo cuando los tipos de derivados son totalmente sustituibles por sus tipos base de forma que las funciones que utilizan estos tipos base pueden ser reutilizados con impunidad y los ti derivados se puede cambiar con impunidad.
  - ***El Principio de Sustitución de Liskov dice que las funciones qu punteros o referencias a una clase base debe ser capaz de usar l objetos de las clases derivadas sin conocerlas.***

20. v190. Principio de Sustitución de Liskov

- Errores:
  - **5.5.1. Principio de Sustitución de Liskov**
    - No se cumple que la precondición sea menos restrictiva en el constructor donde la clase derivada no admite valores fuera del rando de [0,2]
- Solución:
  - **5.5.2. Relación de Herencia vs Composición**
    - La clases *TicTacToeCoordinate* no hereda de *Coordinate* sino que se compone de ella y delega su comportamiento