



Modelado y programación 2022-I

Práctica 1

López Miranda Ángel Mauricio

Anaya Solis Daniela



Para compilar la práctica basta con ubicarse en la carpeta src y poner `javac Simulacion.java` y después `java Simulacion`

Integrantes: Daniela Anaya Solis López Miranda Angel Mauricio.

Desarrollo de la práctica: Nos juntamos todos los días desde el día 5 de octubre de 7pm a 10pm para la discusión de la practica y la correcta elaboración del UML ademas de la elaboración del código

Dificultades que presentamos: Presentamos muchas dificultades con el UML ya que no teníamos una idea clara de donde poner el patrón Strategy ya que pensamos que el hecho de tener un supertipo Servicio era mas que suficiente para poder implementarlo mediante la sobreescritura de un método abstracto cobrar, por lo que decidimos preguntar a Arturo y nos sugirió un cambio en el UML pero esto no nos permitía quedarnos con casi nada de código y tuvimos que empezar de cero otra practica, como nos estaba generando ansiedad la herencia decidimos volver a discutir el UML con lo que llegamos a la conclusión de que era mejor tener otra interfaz que hiciera de supertipo para los planes en específico.

Parte teórica: Ventaja Strategy: Permite encapsular comportamiento para reutilizarlo y no repetir código. Desventaja Strategy: La aplicación debe estar al tanto de todas las posibles estrategias para seleccionar la adecuada para cada situación. Ventaja de Observer: La principal ventaja es que proporciona un diseño acoplado entre los objetos que interactúan, significa que los objetos no conocen mucho del otro. El observado sólo conoce la interfaz que implementan los observadores, para agregar o quitar observadores no hay que modificar al observado y ambas clases pueden utilizarse de manera independiente (observado y observador). Desventaja Observer: Aumenta el número de clases necesarias para implementar el problema a solucionar y disminuye la comprensibilidad de este.