Facultad de Ciencias, UNAM
Modelado y Programación
Práctica 2

Hernández Morales José Ángel
No. de cuenta: 315137903
Rubí Rojas Tania Michelle
No. de cuenta: 315121719

07 de agosto de 2020

1. Menciona los principios de diseño esenciales de los patrones Decorator y Adapter. Menciona una desventaja de cada patrón.

   Solución:

   a) Patrón Decorator

   Permite agregar nuevas funcionalidades a las clases sin modificar su estructura. El concepto de este patrón es agregar de forma dinámica nuevo comportamiento o funcionalidades a la clase principal, es decir, vamos a *decorar* los objetos para darles más funcionalidad de la que tienen en un principio.

   Una desventaja es la mantenibilidad del código ya que este patrón crea muchos decoradores similares que a veces son difíciles de mantener y distinguir.

   b) Patrón Adapter

   Este patrón sirve cuando tenemos interfaces diferentes o incompatibles entre sí y necesitamos que el cliente pueda usar ambas del mismo modo. Adapter convierte las interfaces existentes en una nueva interfaz para lograr la compatibilidad y la reutilización de las clases no relacionadas en una aplicación.

   Una desventaja es que a veces se requieren muchas adaptaciones a lo largo de una cadena de adaptadores para alcanzar el tipo que se requiere. Otra podría ser que este patrón aumenta innecesariamente el tamaño del código, ya que la herencia de clases se utiliza menos y mucho código se duplica innecesariamente entre las clases.

2. Funcionalidad del programa

   a) Primero, debemos posicionarnos dentro de la carpeta **src**. Aquí se encuentran todas las clases de nuestra práctica.

   b) Compilamos todas nuestras clases.

   ```
   $ javac *.java
   ```

   c) Ejecutamos nuestro programa principal.

   ```
   $ java Main
   ```

Referencias

[1] https://experto.dev/patron-de-diseno-decorator-en-java/

[2] https://www.codeproject.com/tips/468951/decorator-design-pattern-in-java

[3] https://www.genbeta.com/desarrollo/patrones-de-diseno-decorator

[4] https://reactiveprogramming.io/blog/es/patrones-de-diseno/adapter

[5] https://experto.dev/patron-de-diseno-adapter-en-java/

[6] https://javarevealed.wordpress.com/tag/adapter-pattern/

[7] https://javarevealed.wordpress.com/tag/adapter-pattern/