

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias

Equipo NullPointerException:

Adrián Aguilera Moreno - 421005200

Diego Angel Rosas Franco - 318165330

Marco Antonio Rivera Silva - 318183583



Modelado y programación

Práctica 2

Menciona los principios de diseño esenciales del patrón State, Template e Iterator:

Template

Principios de diseño esenciales:

- **Template define los pasos de un algoritmo en un método (un esqueleto):**
En Java, dentro de este método tenemos otros métodos que estarían representando los pasos del algoritmo.
- **Template define algunos pasos de este algoritmo y otros los deja a criterio de las subclases que implementan este algoritmo:**
En Java, podemos ver a estos pasos como métodos ya definidos y que las subclases no necesitarán volver a definir, y métodos abstractos, que son aquellos pasos que deben definir las subclases según les corresponda.
- **Algunos de estos pasos (métodos) ya definidos pueden ser hooks:**
Es decir, las subclases deben decidir si la implementación que tienen coincide con su fin o deben redefinirla.
- Evita que subclases reescriban código (pasos) de un determinado algoritmo.
- Permite a las subclases adaptar el algoritmo sin modificar la estructura del mismo.

Desventajas:

- **Agregar un paso implicaría tener que implementar ese paso (si se requiere) en todas las subclases.** Es decir, si en nuestro algoritmo agregamos uno o más pasos extras que las subclases deben implementar, este cambio puede ser muy exhaustivo si es que se tienen demasiadas subclases.
- **Si se usa mal el patrón perderíamos flexibilidad en el código.**
Ya que al tener una estructura definida, y querer un cambio radical en el algoritmo, tendríamos que implementar uno nuevo.

State

Principios de diseño esenciales:

- Permite a un objeto cambiar su comportamiento (indistintamente) de manera instantánea en tiempo de ejecución, y el objeto cambia cuando su estado interno se ve modificado de alguna manera.
- El patrón de diseño “state” se usa cuando una clase (una entidad) tiene distintas fases (estados) y en cada una de estas su comportamiento es distinto.
- Se encapsulan las acciones a realizar dando como resultado un estado.

Desventajas:

- Si se tienen demasiados estados y hay varias conexiones entre ellos, puede resultar difícil el saber como se relacionan teniendo solamente el código.
- El impacto en espacio, ya que por cada estado tendríamos una clase y si son demasiados estados, sería mucho espacio.

Iterator

Principios de diseño esenciales:

- Nos permite acceder a los elementos de un objeto que los contiene de manera secuencial sin exponer su representación interna mediante Iteradores.
- Ayuda a generalizar la tarea de recorrer los elementos de una estructura con un Iterador.

Desventajas:

- **El iterador afecta el dinamismo de la estructura:**

Por ejemplo, podría surgir un cambio en la estructura, ya sea la inserción o eliminación de un elemento, causando que el Iterador tenga que actualizarse y posicionarse donde se había quedado o en una determinada posición.

Instrucciones de instalación, compilación y ejecución.

Se dará por hecho que el usuario sabe moverse en terminal.

Requerimientos previos:

- Se debe contar con Java en su computadora. De preferencia la versión más reciente.

Ejecución del proyecto:

- Si está leyendo esto significa que desempaquetó con éxito el proyecto.
- Abra su terminal y diríjase a la ruta donde desempaquetó el proyecto.
- Una vez estando en la ruta `Practica02_NullPointerException`, diríjase a `Practica02_NullPointerException/src/fciencias/modelado/`
- Ejecute: `"javac Practica02.java"`, esto generará los `.class` del proyecto.
- Ejecute: `"java Practica02"`, esto ejecutará el proyecto mostrándole el menú solicitado para la practica.

