

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 4

Menciona los principios de diseño esenciales de los patrones Factory, Abstract Factory y Builder. Menciona una desventaja de cada patrón.

Factory

Principios de diseño :

- Esta abierto al cambio pero no a la modificación, es decir que te permiten añadir clases nuevas sin que el código tenga que cambiar de ninguna manera.
- Si una de las clases, que no sea la principal y que a su vez forme parte del patrón, no funciona de manera correcta, esta no afecta a las demás, pues es esta quién depende de alguna otra y no al réves.
- Ordena la herencia en las clases.

Desventajas:

- No es el patrón más óptimo para complejidad en espacio.
- Se necesitan muchas clases para concretar el patrón, y cada vez que se quiera extender, entonces se agregan más clases.
- Cada vez que se necesita anexar más partes de código le agregamos complejidad al diseño, esto no permite que sea tan fácil.

Abstract Factory

Principios de diseño :

- Encapsula la responsabilidades de los objetos.
- Hace intercambiables compartimientos que ya dependen de alguna implementación del patrón `abstract factory`.
- Los comportamientos en una familia dada son compatibles entre sí.
- Se puede mover el código de creación de productos a un solo lugar, haciendo que el código sea más fácil de mantener.
- Es abierto al cambio y cerrado a la modificación

Desventajas:

- Cada vez que necesitemos anexar más comportamientos o clases, se necesitará una nueva interfaz (en caso de no poder depender de alguna ya existente) e implementar cada método en esta.
- Crece en complejidad en espacio.
- Hace más complejo el diseño del código.

Builder

Principios de diseño :

- El código puede ser reutilizable, o ser extendido y modificado de manera significativa, esto simplifica el proceso de extensión cuando el comportamiento a anexar tiene similitudes con los ya existentes.
- Puedes construir objetos paso a paso.
- A menudo se usa para construir estructuras compuestas.

Desventajas:

- El diseño del código se vuelve complejo.
- Aumenta la complejidad en espacio.

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 `Practica04_NullPointerException`, diríjase a `Practica04_NullPointerException/src/fciencias/modelado/`
- Ejecute: `"javac Práctica04.java"`, esto generará los `.class` del proyecto.
- Ejecute: `"java Práctica04"`, esto ejecutará el proyecto mostrándole el menú solicitado para la practica.

0.0.1 Lista de Estadísticas de las Partes de la Nave

Tipo Coponente	Precio	Peso	Ataque	Defensa	Velocidad
Laser Simple	79,834.99\$	999.52 kg	7	2	-
Misiles de Plasma	298,777.99\$	1,987.52 kg	23	2	-
Laser Destructor de Planetas	945,785.99\$	10,000.52 kg	50	2	-
Blindaje Simple	50,000.50\$	4,900.90 kg	-	5	-
Blindaje Reforzado	145,000.00\$	9,975.87 kg	-	15	-
Blindaje Fortaleza	385,000.00\$	49750.80 kg	-	50	-
Cabina Un Piloto	19,878.99\$	1,986.10 kg	-	2	-
Cabina Tripulación Pequeña	69,876.00\$	4,899.52 kg	-	5	-
Cabina Ejercito	199,789.99\$	34,567.00 kg	-	10	-
Propulsión Viaje Intercontinental	105,000.00\$	997.90 kg	-	1	100
Propulsión Viaje Interplanetario	294,000.00\$	2,985.90 kg	-	3	500
Propulsión Viaje Intergalactico	875,000.00\$	9,950.49 kg	-	5	1000

Diagrama UML: