

INFORME PRACTICA DE DISEÑO – XESTION DE MAQUINAS DE VENDING

1. Explicación dos principios de deseño (SOLID) usados

Neste exercicio utilizamos diversos principios SOLID, empezando polo primeiro de todos, o principio de responsabilidade única, pois creamos unha clase VendingMachine que se encarga de crear os comportamentos da máquina, para o tipo de cambio temos dúas clases distintas, cambioSimple e cambioDeposito que teñen a responsabilidade de devolver o cambio correcto.

Por outro lado tamén facemos uso do principio aberto cerrado pois podemos cambiar as clases de tipo cambio ou engadir novas no futuro como indica no enunciado sen necesidade de cambiar código na clase principal, ademais estas clases cambio sobrescriben o método da interface polo tanto tampouco sería necesario modificar código nela.

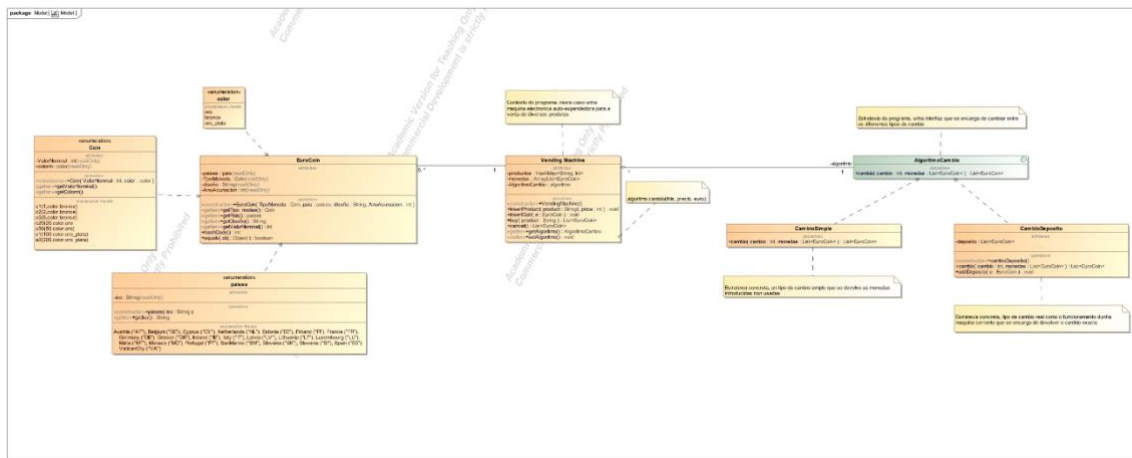
O principio de substitución de Liskov non é utilizado para este exercicio, sen embargo si facemos uso do principio de inversión da dependencia pois a utilización dunha interface que fai que tanto a clase principal como as dúas clases de cambio dependas del. Tamén podemos comentar que na clase VendingMachine traballamos con variables List do cal obtemos vantaxes a hora de traballar coa interface.

O principio de segregación de interfaces para este caso non se utiliza pois so necesitamos dunha interfaz sinxela, no caso de ser un exercicio mais complexo poderíamos vernos na situación de usar este principio.

2. Explicación do patrón de deseño utilizado

Para este exercicio decidimos utilizar o patrón estratexia xa que se encarga de definir unha familia de algoritmos, encapsulalos e facelos intercambiables. O motivo da utilización deste patrón é que temos unha maquina auto-expendedora de diversos produtos (contexto do patrón), e queremos que traballe con distintos sistemas de cambios, cambio simple e cambio deposito (estratexias concretas), ademais plantéase o uso de outros tipos de cambio para o futuro. Poren decidimos usar este patrón para separar a parte do código que varia do contendo mediante unha interfaz (estratexia) que é un algoritmo que permite intercambiar o tipo de cambio según o necesitemos.

3. Diagrama de clase



(ver archivo .pdf)

4. Diagrama dinámicos

Para este ejercicio decidimos utilizar dous diagramas de secuencias similares para representar o comportamento do método buy e a súa comunicación coa interface e os dous tipos de cambio. A primeira parte de ámbolos dous diagramas é igual, a diferenza está no algoritmo, pois dependendo do tipo de cambio no que esteamos interesados cambia a representación do diagrama. E importante ter en conta que cando non se pode comprar determinado produto mostrase unha mensaxe de alerta e VendingMachine permanece no mesmo estado.

