Universidade do Minho

Licenciatura em Engenharia Informática

Ano letivo 2022/2023

POO – Trabalho prático

Grupo 38

Uma imagem com pessoa, homem, interior, pousar

Descrição gerada automaticamenteUma imagem com pessoa, parede, interior, homem

Descrição gerada automaticamenteUma imagem com pessoa, interior

Descrição gerada automaticamenteAntónio Silva(A100533), José Matos(A100612), Pedro Silva(A100745)

**Índice**

1. **Introdução**
2. **Classes**
   1. Artigo
   2. Sapatilhas
   3. TShirt
   4. Mala
   5. Utils (Artigos)
   6. Utilizador
   7. Encomenda
   8. Utils (Encomenda)
   9. Transportadora
   10. Utils(Transportadora)
   11. SaveLoad
   12. Clock
   13. UI
   14. Info
   15. InfoUtils
   16. Manage
   17. ManageUtils
   18. Stats
   19. StatsUtils
   20. ControlCenter
   21. TimeTravel
   22. LocalDateTimeTypeAdapter
   23. AutoRun
   24. ErrorCode
   25. ErrorHandler
   26. Vintage
   27. Utils (Vintage)
   28. Main
3. **Diagrama de classes**
4. **Aplicação desenvolvida e bibliotecas utilizadas**
5. **Conclusão**

**Capítulo 1**

**Introdução e principais objetivos**

**Capítulo 2**

**Classes**

**2.1.** Artigo

Quando nos foi proposto o projeto, foi recomendado que fosse possível uma futura implementação de outros tipos de artigos e nós para facilitarmos essa implementação criamos a classe abstrata Artigo que teria todas as características comuns a todos os artigos.

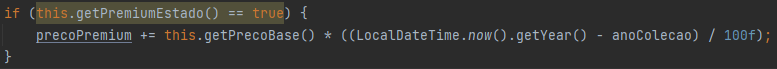
Inicialmente temos três constantes inteiras(MALA, SAPATILHAS e TSHIRT) para identificar os tipos de artigos que temos atualmente. As outras características comuns são:

* private int tipo
* private float estadoUtilizacao
* private int numDonos
* private String descricao
* private String marca
* private int codigo
* private float precoBase
* private int codigoVendedor
* private Transportadora transportadora
* private boolean premiumEstado

Futuramente iremos falar do que são as Transportadoras e os artigos que não são supostos ter premium irão sempre devolver false não interferindo de qualquer forma nos seus cálculos e estatísticas.

Criamos um construtor de artigo de forma a que as subclasses que sejam extensões do artigo poupem trabalho na sua criação, informamos que todas têm de ter uma função de calcularPreco e calcularCorrecao dependo da transportadora, por exemplo. Tem a função de calcularCorrecaoPremium já implementada, porque esta será igual para todos que utilizem premium.

A forma que arranjamos de calcular a evolução do preço de um artigo premium foi:



Dá para compreender pela função que quanto mais antigo for o Artigo, maior será o preço do Artigo premium.

Finalmente, tem os getters e setters de todos os atributos.

**2.2.** Sapatilhas

O primeiro artigo que implementamos foram as sapatilhas. Como mencionamos anteriormente, todos os artigos vão ser uma extensão da classe Artigo, sendo que cada subclasse tem as suas características específicas. No caso das sapatilhas, seria ter atacadores ou atilhos, o tamanho delas, a cor e o ano da coleção.

Estes atributos estão guardados do seguinte modo:

* public static final int ATACADORES = 0;
* public static final int ATILHOS = 1;
* private int tamanho;
* private int atacadores;
* private String cor;
* private int anoColecao;

As contantes ATACADORES e ATILHOS servem para facilitar a compreensão do tipo de atacadores.

Além disso têm o seu próprio construtor, utilizando a funcionalidade super com os atributos da classe Artigo que são usados nas sapatilhas, a sua própria função de calcularPreco que utiliza o precoBase, a calcularCorrecaoPremium e a sua própria calcularCorrecao.

No caso das sapatilhas, a calcularCorrecao,além de se adicionar o preço da Transportadora, caso não sejam novas, ou seja, estado de Utilização diferente de 1, sendo que estes podem ser 0.25,0.5,0.75 ou 1, vão sofrer uma redução de preço quão pior for o seu estado(Pouco Uso 25%, Muito Uso 50% e Estragado 75%) e caso sejam novas e de tamanho maior de 45 recebem ainda mais uma dedução de 25% também.

No nosso caso, o estado nunca terá nada a ver com o número de donos pois achamos que isso é uma correlação pouco baseada, porque dependendo do tipo de dono, o artigo pode não diminuir no seu estado de utilização, por exemplo, algo pode ter 3 donos e continuar praticamente novo e também ter 3 donos e ficar completamente estragado.

Por fim, tem os getters e os setters e na sua toString utiliza funções de utils que serão todas explicadas no seu devido tempo.

**2.3.** TShirt

Sendo a TShirt outra extensão da classe Artigo, também só terá de implementar as suas características específicas. No caso da TShirt são o tamanho e o seu padrão.

Para facilitar a identificação de cada usamos as seguintes constantes:

Uma imagem com texto

Descrição gerada automaticamente

private int padrao;

private String tamanho;

Depois, tal como nas sapatilhas, criamos o construtor utilizando novamente o mesmo método super.

O calcularPreco é implementado, tal como o calcularCorrecao.

O calcularCorrecao no caso das TShirts funciona mais focado no seu padrão. Além de adicionar o valor da transportadora, caso o padrão seja Liso não há mais nenhum desconto, caso contrário, se a TShirt não for nova, independente do padrão, exceto o Liso, teria um desconto fixo de 50%.

Por fim, está tal e qual as Sapatilhas, getters, setters e toString

**2.4.**  Mala