

Trabalho de Fundamentos Kotlin

Professores: Emmerson Santa Rita da Silva
Disciplina: PDM

Enunciado

A figura abaixo representa parte do diagrama de classes de um supermercado que funciona na Web – o SPW S/A. O SPW é um supermercado online que trabalha com produtos de vários gêneros. Este diagrama apresenta apenas três gêneros: Leite, Presunto e Achocolatado.

cd DC - Supermercado

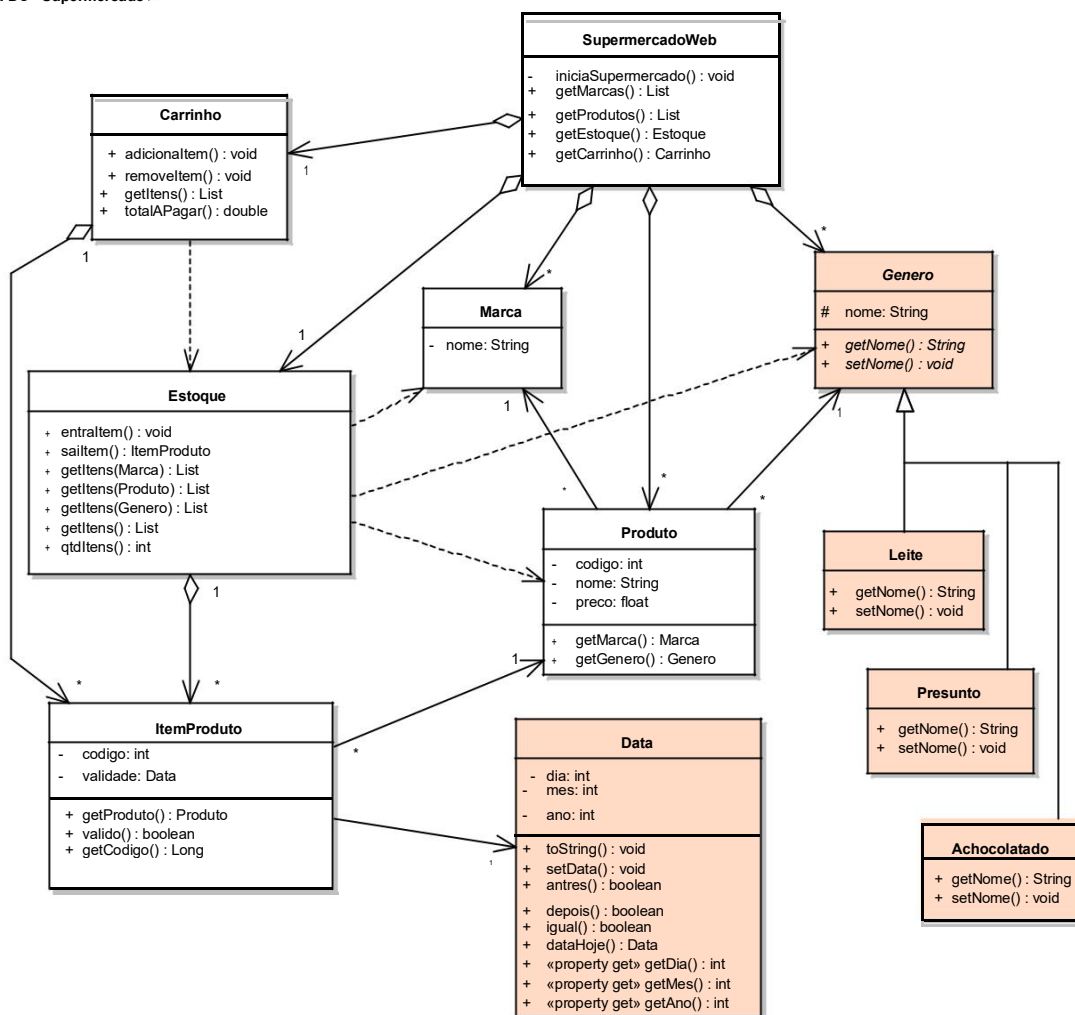


Figura – Parte do diagrama de classes do supermercado SPW S/A.

Há diversos produtos de diversos gêneros e marcas. Veja alguns exemplos:

PRODUTO	GÊNERO	MARCA
Ninho	Leite	Nestlé
Molico	Leite	Nestlé

Farinha Láctea	Leite	Nestlé
Leite em Pó Itambé	Leite	Itambé
Leite em Caixa	Leite	Duleite
Leite em Pó	Leite	Duleite
Leite em Caixa	Leite	Parmalat
Nescau	Achocolatado	Nestlé
Ovomaltine	Achocolatado	ABF
Toddy	Achocolatado	Pepsico do Brasil
Muky	Achocolatado	Bretzke
Chocolate em Pó	Achocolatado	Nestlé

Descrição das classes

Marca. Indica a marca de um produto à venda no supermercado. Possui um nome. O construtor com parâmetros inicializa o nome da marca.

Produto. Um produto indica algo à venda no supermercado, possui um código, nome e preço. Um produto está associado a uma marca, a muitos itens daquele produto e a um gênero. O método `getMarca()` retorna o objeto marca que corresponde à marca de um produto. O construtor com parâmetros inicializa todos os atributos e a associação com `Marca`.

Gênero. Um gênero é uma forma de classificar produtos que possuem características gerais comuns, por exemplo, produtos lácteos (leite), achocolatados, biscoitos, etc. A classe abstrata `Genero` possui como atributo `nome` (para o nome do gênero), e os métodos abstratos `get` e `set` implementados nas subclasses de `Genero`.

Leite, Presunto e Achocolatado. São subclasses concretas de `Genero`. O construtor de cada uma inicializa o atributo `nome` com o nome específico do gênero.

ItemProduto. Um produto indica algo à venda no supermercado, porém, não é um produto o que você leva para casa. O que você vê na prateleira, compra e leva para casa é um item de produto. Por exemplo, o supermercado pode vender 50 latas de Leite Ninho, 30 pacotes de Ovomaltine, 10 latas de Nescau e 10 pacotes de Presunto Delícia. Em outras palavras, o supermercado vende 50 itens do produto Leite Ninho, 30 itens do produto Ovomaltine, e assim por diante. Inicialmente, todos os itens de produto entram no estoque do SPW. A classe `ItemProduto` possui um código (o código de barras). Um item está associado a uma data de validade e a um único produto. O método `valido()` retorna *true* se o item de produto for válido, isto é, se estiver dentro da data de validade, ou retorna *false* se o item já está vencido. Um item é válido se sua data de validade vem depois da data do dia corrente.

Data. A classe `Data` armazena uma data no formato DD/MM/AAAA. O construtor da classe recebe o dia, o mês e o ano e inicializa os atributos correspondentes. Esta classe não verifica a validade de datas (por exemplo, 45/11/2010 é uma data inválida). O método `toString()` retorna uma string com a data no formato DD/MM/AAAA. O método `setData()` recebe o dia, o mês e o ano e atualiza os atributos correspondentes. O método `antes()` recebe uma data como parâmetro e retorna *true* se uma data vem antes do seu parâmetro data, caso contrário retorna *false*. O método `depois()` recebe uma data como parâmetro e retorna *true* se uma data vem depois do seu parâmetro data, caso contrário retorna *false*. O método `igual()` recebe uma data como parâmetro e retorna *true* se uma data é igual ao seu parâmetro data, caso contrário retorna *false*. O método estático `dataHoje()` retorna a data corrente. Os demais métodos são métodos *get* para os atributos `dia`, `mes` e `ano`.

Estoque. O estoque armazena todos os itens à venda no supermercado. O método `entraItem()` recebe um item como parâmetro e o adiciona à lista de itens do estoque. O método `saiItem()` recebe um item de produto como parâmetro, o remove do estoque, e o retorna. O método sobrecarregado `getItens()` retorna a lista de itens do estoque. Se o método receber como parâmetro uma marca, então deverá retornar os itens daquela marca no estoque; se o método receber como parâmetro um produto, retornará os itens daquele produto no estoque; e se método receber como parâmetro um objeto gênero (Leite, Presunto, etc), então retornará os itens daquele gênero. Observe que esta última versão do método `getItens()` deve ser polimórfica, isto é, o método não sabe que tipo específico de gênero de produto irá receber. O polimorfismo tratará de decidir se o que foi enviado como parâmetro é um objeto leite, ou presunto, etc. O método `qtdItens()` retorna a quantidade corrente de itens no estoque. A classe não possui construtor com parâmetros nem métodos *set*.

Carrinho. Para fazer compras no SPW, um cliente adiciona o item de produto desejado a um carrinho de compras virtual. Quando isto ocorre, o item adicionado ao carrinho sai do estoque. Quando um cliente remove um item do carrinho (porque desistiu de levar aquele item), o item removido entra novamente no estoque. Um carrinho está associado a vários itens de produto (que vêm do estoque).

O método `adicionaItem()` recebe um item de produto e o estoque como parâmetros. Este método faz o item de produto recebido sair do estoque e, em seguida, adiciona o item retirado do estoque ao carrinho. O método `removeItem()` recebe um item de produto e o estoque como parâmetros. Este método remove o item de produto do carrinho, e faz a entrada do item no estoque.

O método `totalAPagar()` retorna a soma dos preços dos itens do carrinho, isto é, o total que o cliente deverá pagar pelos itens presentes no seu carrinho. Se o carrinho estiver vazio, este método retorna 0,0.

O método `getItens()` retorna os itens do carrinho.

A classe não possui construtor com parâmetros nem métodos *set*.

SupermercadoWeb. É classe que representa o supermercado e todos os seus elementos: estoque, carrinho, uma lista de gêneros, uma lista de produtos e uma lista de marcas. O método `iniciaSupermercado()` é chamado pelo construtor padrão da classe e instancia estoque, gêneros, marcas, produtos, carrinho, itens e faz os itens entrarem no estoque. Neste método instancie 10 itens do gênero Leite de três marcas distintas, 10 itens do gênero Achocolatado de três marcas distintas, e 10 itens do gênero Presunto de três marcas distintas. Escolha datas de validade e preços sortidos.

O que deve ser feito

Codifique em Kotlin as classes acima conforme indica o diagrama de classes e esta especificação, empregando *encapsulamento*, *herança*, *associações*, *sobrecarga de métodos* (e qualquer outro assunto ministrado nas aulas de Kotlin, se necessário), e responda as questões abaixo relativas à avaliação.

Questões– Parte I

1ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada `AP2Questao1` com um método *main* que mostre na tela todos os itens estocados no supermercado SPW e o total de itens ao final, de acordo com o seguinte formato.

```
##### ESTOQUE #####
CÓD NOME          GENERO          MARCA          PREÇO          VALIDADE          VÁLIDO?
-----
11  Ninho          Leite          Nestlé          R$10,35  11/04/2010  não
43  Ovomaltine     Achocolatado  ABF            R$15,00  11/10/2010  sim
47  Delícia        Presunto      Perdigão       R$5,50   26/11/2011  sim
21  Tio Joca        Leite          Fazenda        R$11,00  01/12/2010  sim
31  Ovomaltine     Achocolatado  ABF            R$15,00  05/03/2011  sim
41  Nescau          Achocolatado  Nestlé         R$12,50  15/10/2009  não
12  Ninho          Leite          Nestlé         R$10,99  23/09/2009  não
...
TOTAL: 35 itens.
```

Observação: cada coluna não precisa estar exatamente ajustada como mostrado acima.

2ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada AP2Questao2 com um método *main* que mostre na tela todos os itens estocados no supermercado SPW agrupados por gênero, e o total de itens por gênero, de acordo com o seguinte formato.

```
##### ESTOQUE POR GÊNERO #####
CÓD NOME          GENERO          MARCA          PREÇO          VALIDADE          VÁLIDO?
-----
11  Ninho          Leite          Nestlé          R$10,35  11/04/2010  não
21  Tio Joca        Leite          Fazenda        R$11,00  01/12/2010  sim
12  Ninho          Leite          Nestlé         R$10,99  23/09/2009  não
TOTAL GÊNERO: 3 itens.
31  Ovomaltine     Achocolatado  ABF            R$15,00  05/03/2011  sim
41  Nescau          Achocolatado  Nestlé         R$12,50  15/10/2009  não
43  Ovomaltine     Achocolatado  ABF            R$15,00  11/10/2010  sim
TOTAL GÊNERO: 3 itens.
...
TOTAL: 35 itens.
```

Observação: cada coluna não precisa estar exatamente ajustada como mostrado acima, e os itens não precisam ser apresentados em ordem alfabética de gênero, marca ou nome.

3ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada AP2Questao3 com um método *main* que mostre na tela todos os itens válidos estocados no supermercado SPW agrupados por gênero e marca, e o total de itens mostrados, de acordo com o seguinte formato.

```
##### ESTOQUE POR GÊNERO E NOME #####
CÓD NOME          GENERO          MARCA          PREÇO          VALIDADE          VÁLIDO?
-----
23  Tio Joca        Leite          Fazenda        R$11,00  01/12/2010  sim
27  Tio Joca        Leite          Fazenda        R$11,00  01/12/2010  sim
13  Ninho          Leite          Nestlé         R$12,00  05/10/2010  sim
43  Ovomaltine     Achocolatado  ABF            R$15,00  11/10/2010  sim
49  Ovomaltine     Achocolatado  ABF            R$15,00  11/10/2010  sim
51  Ovomaltine     Achocolatado  ABF            R$15,00  11/10/2010  sim
41  Nescau          Achocolatado  Nestlé         R$12,50  15/10/2009  sim
40  Nescau          Achocolatado  Nestlé         R$12,50  15/10/2009  sim
47  Delícia        Presunto      Perdigão       R$5,50   26/11/2011  sim
48  Delícia        Presunto      Perdigão       R$5,50   26/11/2011  sim
48  Salsichon      Presunto      Sadia          R$7,50   23/07/2009  sim
...
TOTAL: 35 itens.
```

Observação: cada coluna não precisa estar exatamente ajustada como mostrado acima, e os itens não precisam ser apresentados em ordem alfabética de gênero, marca ou nome.

4ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada AP2Questao4 com um método *main* para (1) contar quantos itens válidos (não vencidos) há no estoque; (2) contar quantos itens vencidos há no estoque; (3) somar os dois valores e verificar se o resultado é igual ao total de itens no estoque. Mostre todos os resultados na tela.

5ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada AP2Questao5 com um método *main* para: (1) selecionar os itens do estoque cujo código for ímpar e adicioná-los ao carrinho de compras do SPW; (2) mostrar os itens do carrinho, sua quantidade e quanto o cliente irá pagar pelos itens no carrinho; (3) mostrar os itens restantes no estoque e o total de itens no estoque. Observações: você sabe que precisa usar um comando *for* para percorrer a lista de itens do estoque. Entretanto, à medida que percorrer lista, alguns itens são retirados da lista para entrarem no carrinho. Isto faz com que a lista seja modificada ao mesmo tempo em que é percorrida.

6ª questão (1,5 ponto). Codifique em Kotlin uma classe chamada AP2Questao6 com um método *main* para: (1) selecionar aleatoriamente alguns itens válidos (não vencidos) do estoque e adicioná-los a um carrinho de compras; (2) mostrar os itens do carrinho e sua quantidade de itens; (3) mostrar os itens restantes no estoque e o total de itens no estoque. Observe que tanto a quantidade de itens a serem adicionados no carrinho, quanto quais itens serão incluídos ou não no carrinho, será feita pelo programa de forma aleatória. Observação: valem aqui as mesmas observações da 5ª questão.

7ª questão (1,5 ponto). Codifique em Kotlin uma classe chamada AP2Questao7 com um método *main* que permita a um usuário entrar com o nome de uma marca, e realizar uma busca por todos os itens de produtos válidos do estoque que sejam da marca fornecida. Mostre os itens encontrados e sua quantidade.

8ª questão (2,0 pontos). O professor irá inspecionar o código do trabalho para verificar a legitimidade (se o trabalho não é uma cópia de outros) e a correção das implementações. Dois ou mais trabalhos com códigos iguais receberão a nota 0,0 (zero) na **AP2**.

.....

Questões– Parte II

1ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada AFQuestao1 com um método *main*. O programa deve permitir que um usuário entre com o nome de um gênero, e realize uma busca por todos itens do estoque que pertençam ao gênero fornecido. Mostre os itens encontrados e sua quantidade total seguindo o seguinte formato (supondo que o gênero escolhido seja achocolatado):

#####	ESTOQUE	#####				
CÓD	NOME	GENERO	MARCA	PREÇO	VALIDADE	VÁLIDO?
---	-----	-----	-----	-----	-----	-----
43	Ovomaltine	Achocolatado	ABF	R\$15,00	11/10/2010	sim
49	Ovomaltine	Achocolatado	ABF	R\$15,00	11/10/2010	sim
51	Ovomaltine	Achocolatado	ABF	R\$15,00	11/10/2010	sim
41	Nescau	Achocolatado	Nestlé	R\$12,50	15/10/2009	sim
40	Nescau	Achocolatado	Nestlé	R\$12,50	15/10/2009	sim

...
TOTAL: 5 itens.

Observação: nesta questão você não pode usar método `getItens()` sem parâmetro da classe `Estoque`. Dica: identifique o gênero que o cliente deseja, instancie um objeto do gênero escolhido e use um certo método da classe `Estoque` para obter os itens.

2ª questão (1,0 ponto). Codifique em Kotlin uma classe chamada `AFQuestao2` com um método *main*. O programa deve permitir que um usuário entre com uma data (dia, mês e ano), e realize uma busca por todos os itens de produtos do estoque cuja data de validade ultrapassou (é mais antiga) que a data fornecida. Insira todos os produtos encontrados em uma lista *x*. Ao final, mostre todos os itens da lista *x* e sua quantidade total.

3ª questão (1,5 ponto). Codifique em Kotlin uma classe chamada `AFQuestao3` com um método *main*. O programa deve realizar uma busca, identificar e mostrar os produtos mais baratos de cada gênero de produto do SPW. Por exemplo, considerando o gênero leite, e os produtos {"Leite Ninho", R\$11.00), ("Leite Itambé", R\$9.50), ("Leite Bafo de Vaca", R\$3.25), ("Leite Integral Du Bão", R\$8.45), ("Leite Pó D'ouro", R\$35.99)}, o produto mais barato é o "Leite Bafo de Vaca" de R\$3.25.

4ª questão (1,5 ponto). Codifique em Kotlin uma classe chamada `AFQuestao4` com um método *main*. O programa deve: (1) calcular quanto o SPW irá ganhar com a venda de todos os itens do estoque e mostrar o estoque com o total de itens e o lucro; (2) retirar todos os itens vencidos do estoque (considere que serão jogados no lixo) calculando o prejuízo com a retirada dos itens; (3) mostrar o estoque sem os itens vencidos, o total de itens e o prejuízo; (4) mostrar o lucro real considerando o prejuízo. Observação: valem aqui as mesmas observações da 5ª questão da AP2.

5ª questão (1,5 ponto). Codifique em Kotlin uma classe chamada `AFQuestao5` com um método *main*. O programa deve: (1) mostrar o estoque e o carrinho (vazio) do SPW; (2) preencher o carrinho com dez itens do estoque, aleatoriamente, e mostrar o estoque e o carrinho; (3) colocar todos os itens do carrinho de volta no estoque, e mostrar o estoque e o carrinho. Observação: valem aqui as mesmas observações da 5ª questão da AP2, no entanto, deverá haver também uma lista auxiliar do Carrinho para o item (3) desta questão.

6ª questão (1,5 ponto). Modifique a classe `Supermercado` para conter dois carrinhos de compras: `carrinhoA` e `carrinhoB`. Codifique em Kotlin uma classe chamada `AFQuestao6` com um método *main*. O programa deve: (1) preencher o `carrinhoA` com todos os itens válidos do estoque; (2) preencher o `carrinhoB` com todos os itens inválidos do estoque; (3) mostrar o estoque e os dois carrinhos. Observação: valem aqui as mesmas observações da 5ª questão da AP2.

7ª questão (2,0 pontos). O professor irá inspecionar o código do trabalho para verificar a legitimidade (se o trabalho não é uma cópia de outros) e a correção das implementações. Dois ou mais trabalhos com códigos iguais receberão a nota 0,0 (zero) na **AF**.

Importante

Você somente conseguirá responder as questões acima se codificar corretamente todas as classes do `Supermercado SPW` aplicando encapsulamento, associações, herança, etc. A resolução (codificação em Kotlin) de cada uma das questões das avaliações acima e seu resultado correto é uma prova de que você codificou o modelo do SPW corretamente.

O que deve ser entregue

Você deve entregar seu projeto do supermercado, completo, via google classroom. Certifique-se de que o projeto foi exportado corretamente; se o professor não conseguir importar (abrir) o projeto, não poderá corrigi-lo e a nota do trabalho será 0,0 (zero). Deve ser entregue o link do repositório Git contendo o exercício completo.

O diagrama de classes do supermercado SPW não está totalmente completo. Isto significa que em alguns casos você terá de adicionar novos métodos em algumas classes (por exemplo, métodos *get*, *se necessário*).

Para cada questão utilize os métodos das classes do SPW que mais facilmente possam ajudar você a resolver a questão.

Quando for codificar preste atenção nas navegações das associações entre as classes, para saber em quais classes você deve incluir as associações.

Boa diversão!

\\('t')/\\
\\u//
