

LISTA 3

1. Implemente, em Java, as classes **Tabuleiro** e **Peça**. A classe **Tabuleiro** guarda um número variável de peças e têm o método **desenhaPeças()** que é responsável por desenhar todas as peças guardadas. A classe **Peça** tem dois atributos inteiros que representam as suas coordenadas no tabuleiro, um atributo que representa a cor da peça (cadeia de caracteres) e dois métodos: o método **move()** e o método **devolveCor()**. Existem dois tipos de peças: **Cabo** e **Sargento**. Cada tipo de peça move-se de forma distinta: as peças do tipo **Cabo** avançam uma unidade em cada uma das coordenadas enquanto que as peças do tipo **Sargento** avançam 3 unidades em cada uma das coordenadas. Para simplificar o seu trabalho considere que o tabuleiro não tem limites.

Se necessário pode implementar outras classes e/ou métodos.

2. Implemente um programa para ler os dados de N produtos (**N** fornecido pelo utilizador). No final deve mostrar a etiqueta de preço de cada produto na mesma ordem em que foram digitados invocando o método **etiquetaProduto()**.

Todo produto possui nome e preço. Produtos importados possuem uma taxa de alfândega, e produtos usados possuem data de fabricação. Estes dados específicos devem ser acrescentados na etiqueta de preço. Para produtos importados, a taxa de alfândega deve ser acrescentada ao preço final do produto.

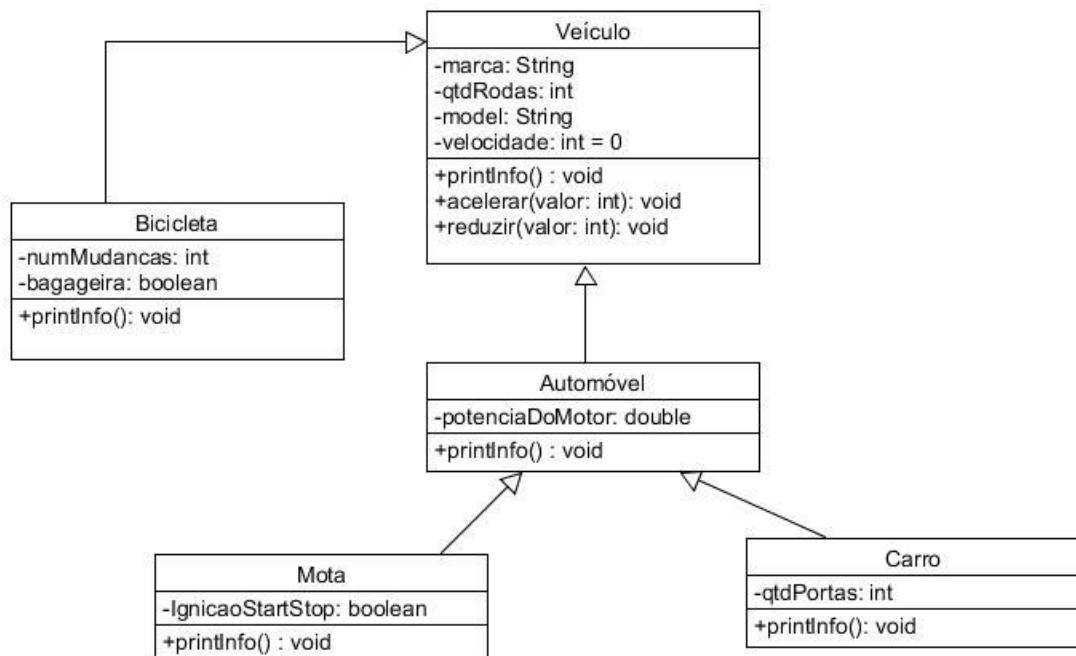
3. Concretize, em Java, as classes **Fábrica** e **Empregado**. A classe **Fábrica** guarda um número variável de empregados e tem o método **fabricar()** que é responsável por colocar todos os seus empregados a realizar trabalho. Cada empregado tem um nome e um ano de nascimento e tem o método **trabalha()** e os métodos que permitem aceder aos seus atributos. Existem dois tipos de empregado: **Operário** e **Gestor**. Cada tipo de empregado tem o seu próprio comportamento no que diz respeito ao método **trabalha**. Para simplificar, considere que este método escreve a cadeia de caracteres "Gestor X a gerir" no caso de um gestor e "Operário X a trabalhar" no caso de um

operário. em ambos os casos, "X" deve ser substituído pelo nome do empregado em causa..

Se necessário, pode concretizar outras classes e/ou métodos ou utilizar classes já existentes da biblioteca Java.

4. Modele e implemente uma classe que represente uma versão muito simples do conceito Gato. Um Gato tem como características o nome, a idade e o peso.
 - a) Implemente o método de comparação (equals), por forma a considerar que dois gatos são iguais se as suas características forem iguais.
 - b) Implemente o método de apresentação (toString), por forma a produzir uma cadeia de caracteres onde seja apresentado o nome, a idade e o peso do gato.
 - c) Implemente um programa (main) que ilustre a utilização dos métodos anteriores.

5. Dado o diagrama abaixo



- a) Crie o **construtor** para cada uma das classes e os métodos **get** e **set** para cada um dos atributos das classes.

- b) O método **acelerar** da classe Veículo deve somar o valor passado por parâmetro a velocidade actual do veículo.
 - c) método **reduzir** da classe Veículo deve subtrair o valor passado por parâmetro a velocidade actual do veículo.
 - d) O método **printInfo** de cada uma das classes deve mostrar no ecrã o conteúdo de cada um dos atributos da classe.
6. Modele e implemente a classe **Table** que contém um número fixo de inteiros (especificado na altura da criação das suas instâncias). Essa classe possui um método (**print**) que permite imprimir todo o conteúdo, aplicando uma transformação a cada valor. A transformação aplicada varia de acordo com o tipo do objecto **Transform**, passado como argumento de **print**. As potenciais transformações têm um método **transform** (aceita um inteiro e devolve outro inteiro) que permite modificar um valor inteiro de uma forma específica. Defina os conceitos **Table** e **ITransform**. Defina ainda duas transformações: uma que calcula o sucessor de cada inteiro (**Successor**); e outra que calcule o quadrado de cada um dos valores da tabela (**Square**). Pode realizar outros conceitos/métodos que julgue necessitar. Evite repetições de código.
7. Modele e implemente a classe **Table** que contém um número fixo de inteiros (especificado na altura da criação das suas instâncias). Essa classe possui um método (**print**) que permite imprimir todo o conteúdo, aplicando um formato a cada valor. O formato aplicado por **print** varia de acordo com o tipo de “impressora” (objecto do tipo **Printer**, passados como argumento de **print**). As potenciais impressoras têm um método **format** (aceita um inteiro e devolve uma cadeia de caracteres – tipo **String**) que permite apresentar um valor inteiro de uma forma específica para cada impressora. Defina os conceitos **Table** e **IPrinter**. Defina ainda duas impressoras: uma que formata cada inteiro entre parêntesis rectos (**BracketsPrinter**); e outra que repita cada valor duas vezes cada

um dos valores da tabela (**DuplicatePrinter**). Pode realizar outros conceitos/métodos que julgue necessário. Evite repetições de código.