**Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**## Observações**

- Antes de iniciar a sua prova coloque o seu nome e assine nas duas folhas;

- A **prova é individual** e com consulta ao seu próprio material; respostas iguais serão anuladas;

- Lembre que a interpretação do enunciado faz parte da avaliação destas questões;

- A prova deve ser desenvolvida durante o período desta aula, e entregue até no máximo as 21:15. Lembre-se que tens que postar as respostas no AVA3. Caso a prova não seja entregue neste período a nota será zero. Assim, mesmo que não tenha terminado a prova providencie a sua entrega antes do final deste prazo;

- Só é permitido o uso de comandos estudados em aula até o momento, envolvendo o conteúdo das unidades 1, 2, 3, 4, 5 e 6;

- O arquivo .java deverá ser entregue através do “AVA3”. Para isso siga as instruções abaixo:

- Abra o AVA3 na disciplina de Introdução à Programação;

- Clique na opção “Conteúdo” e, em seguida, selecione a opção “Prova2\_Respostas”;

- Adicione os arquivos .java da sua prova;

- Antes de sair da aula deves entregar a folha da prova assinada. Neste momento o professor vai confirmar com o aluno a entrega da prova no AVA3;

- A nota dessa prova será definida após a entrevista com o aluno para verificar se ele realmente fez a prova;

- **ATENÇÃO**: Não é permitido o uso do smartphone durante a aula, nem mesmo para fotografar a prova;

- Boa prova!

**PARTE TEÓRICA: 3,0**

1. (1,0) Analise o seguinte código e responda o que se pede:

**Texto

Descrição gerada automaticamente**

1. (0,5) O texto “Fim do Método” aparecerá no **máximo** quantas vezes e no **mínimo** quantas vezes? Em que momentos aparecerá?

Máximo: 1 vez - se o valor da variável “valor” não for igual a nenhum dos elementos do “vetor”, mas deve considerar somente os elementos do “vetor” até o índice “vetorPos”.

Mínimo: 0 vez – se o valor da variável “valor” for igual a um dos elementos do “vetor”.

1. (0,5) Quantas vezes o laço “for” irá executar? Justifique.

Vai executar a quantidades de vezes igual ao valor da variável “vetorPos”, mas pode “parar” antes caso “valor” seja igual a um valor do “vetor”.

**Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. (2,0) Analise o seguinte código e responda o que se pede:

**Texto

Descrição gerada automaticamente**

1. (0,5) O que está acontecendo no método acima?

O método “teste” inverte os valores do “vetor” em relação aos seus índices. Ou seja, o valores do “vetor” são { 10, 5, 3, 8, 9} ficam { 9, 8, 3, 5, 10}.

1. (0,5) Quantas vezes o laço “for” com índice “j” irá executar?

Justifique.

O laço “j” será executado 10 vezes.

1. (0,5) Qual o conteúdo da variável “vetor” na linha 7 ?

Justifique.

O valor da variável “vetor” é { 9, 8, 3, 5, 10}.

Pois o método “teste” inverte os valores do “vetor”, e como o “vetor” foi passado como “passagem de valor por referência”, o “vetor” tem seus valores trocados com a “volta” do método.

1. (0,5) Qual o conteúdo da variável “numero” na linha 7?

Justifique.

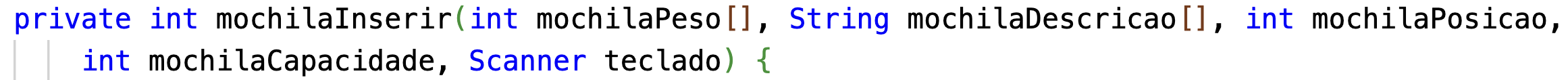
O valor da variável “numero” é 5. Pois mesmo com o seu valor aletardo no método “teste”, como foi passado com “passagem por valor”, ao voltar do método a variável não tem seu valor trocado.

**Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

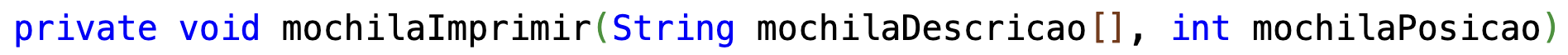
**PARTE PRÁTICA: 7,0**

Você é um aventureiro que possui uma mochila com capacidade limitada de carga informada pelo usuário. Você deseja levar a maior quantidade de itens possíveis na mochila, sem exceder a capacidade máxima de peso da mochila. Cada item possui um **peso** (int) e uma **descrição** (String). Faça uma solução em Java utilizando vetores unidimensionais, seguindo os seguintes métodos:

ATENÇÃO: os métodos DEVEM ser criados de acordo com as assinaturas abaixo, caso contrário a questão do método receberá nota zero.



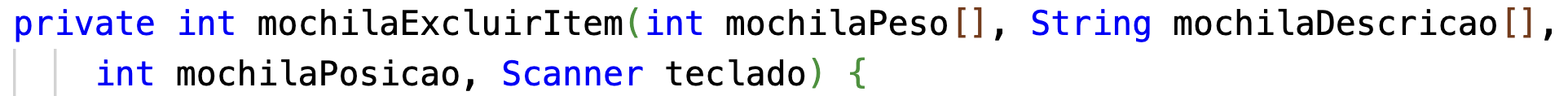
1. (2,0) Se deve incluir o **peso** e a **descrição** do item na mochila, caso a mochila ainda tenha alguma posição disponível e não tenha atingido o peso máximo. Se o peso do novo objeto ultrapassar o peso total da mochila ou não tiver mais vaga disponível na mochila, deve-se apresentar as respectivas mensagens: “Atingiu o peso máximo!” e “Mochila cheia!”.



1. (1,0) Se deve imprimir a mochila listando o valor de cada índice do vetor e as suas respectivas **descrições** na ordem INVERSA ao qual foi informada, ou seja, deve-se imprimir do último item ao primeiro (obs.: não imprimir **peso**);



1. (1,5) Se deve retornar o valor do item mais pesado da mochila para ser impresso dentro do “case” do menu após esse método ser chamado;



1. (1,5) Ao informar a descrição do item, deve-se excluí-lo da mochila. Se o item não for encontrado, deve-se informar que o item não existe. Ao excluir o item do vetor mochila, deve-se excluir o seu peso correspondente no vetor de pesos da mochila, bem como sua descrição no vetor de descrição. Também se deve “reorganizar” os itens (peso e descrição) na mochila para sempre preencherem as primeiras posições vagas da mochila considerando o índice do vetor;



1. (1,0) Para chamar esses métodos, deve-se criar um menu utilizando “switch..case” e “do..while” no construtor, de modo que o usuário possa chamar o método que desejar em qualquer ordem. Lembre-se de criar uma opção para sair do programa.

Lembre-se de fazer as validações (consistências) necessárias. Para armazenar os itens da mochila se deve criar um vetor do tipo “int” para o peso e um do tipo “String” para a descrição dos itens.