



Exercícios

Arrays e ArrayList

Quarta-feira, 12 de abril de 2023.

1. Resolva os exercícios abaixo do Capítulo 7 - *Arrays* - do livro Java Como Programar. 6ª ed. DEITEL, H. M., DEITEL, P. J. São Paulo: Pearson Prentice Hall, 2006.

Exercícios 7.10, 7.12, 7.14, 7.16, 7.19, 7.20 e 7.29.



Nota: Esses exercícios estão disponíveis também nas edições 8 e 10 do livro Java Como Programar, publicados, respectivamente, em 2010 e 2017.

2. Escreva um programa que receba a temperatura média de cada mês do ano e armazene essas temperaturas em um vetor; calcule e imprima a maior e a menor temperatura do ano e em que mês essas temperaturas acontecem.
3. Faça um programa que leia 10 números inteiros, armazene-os em um vetor, calcule e imprima:

- a) o seguinte somatório: $\sum_{i=1}^{10} \frac{1}{a_i}$, onde a_i é um elemento do vetor;
- b) quantos termos da série possuem no denominador um número par;
- c) quantos termos da série possuem no denominador um número ímpar.

4. O sistema de avaliação de uma disciplina obedece aos seguintes critérios:

- durante o semestre são aplicadas três avaliações no valor de 10 pontos cada uma;
- a nota final é obtida pela média aritmética dessas três avaliações;
- é considerado aprovado o discente que obtiver a nota final superior ou igual a 6 pontos e que tiver comparecido a um mínimo de 75 aulas.

Faça um programa que:

- a) Leia os seguintes dados de cada discente: o nome, as três notas das avaliações e sua frequência (número de aulas assistidas).
- b) Calcule e exiba:
 - b.1) para cada discente, o seu nome, a nota final e sua situação (Aprovado ou Reprovado);
 - b.2) a maior e a menor nota final da turma;
 - b.3) o número total de alunos aprovados e reprovados;

- b.4) a porcentagem de alunos aprovados e reprovados;
- b.5) a porcentagem de alunos reprovados por frequência.

5. Desenvolva um programa que receba:

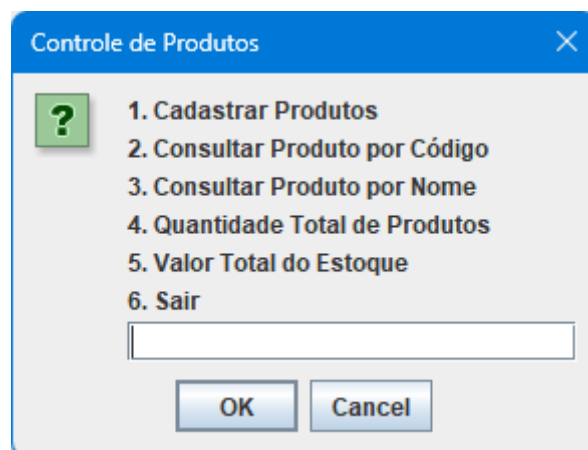
- 10 nomes de produtos vendidos por uma empresa e armazene-os em um vetor;
- a quantidade dos 10 produtos em cada um dos 5 armazéns dessa empresa e armazene-os em uma matriz de 10x5;
- o preço dos 10 produtos e armazene-os em outro vetor.

Calcule e exiba um relatório com as seguintes informações:

- a) a quantidade total de cada produto armazenado em cada armazém;
- b) a quantidade total de cada produto armazenado em todos os armazéns juntos;
- c) o valor total de cada produto em cada armazém;
- d) o valor total de cada produto em todos os armazéns juntos;
- e) o nome de cada produto e o número do armazém que possui a maior quantidade desse produto;
- f) o nome de cada produto e o número do armazém que possui a menor quantidade desse produto;
- g) o nome e o preço do produto mais barato;
- h) o nome e o preço do produto mais caro.

6. Reescreva sua solução do Exercício 5 substituindo os dois vetores e a matriz por um vetor de objetos da classe Produto. Essa classe deve ter os seguintes atributos: nome, quantidade, preço e estoque. Esse deve ser um vetor de 5 elementos do tipo int para representar a quantidade do produto em cada um dos cinco armazéns da empresa.

7. Desenvolva um programa Java GUI que leia o nome, a quantidade disponível em estoque e o preço unitário dos produtos de um supermercado. O programa deve exibir o *menu* abaixo.



Caso o usuário escolha a opção:

- 1. Será permitido que ele faça o cadastro dos produtos segundo os dados supracitados, não

permitindo o cadastramento de mais de um produto com o mesmo código, que deve ser gerado automaticamente pelo programa.

2. Ele deve digitar o código do produto e o programa exibirá os dados do mesmo, caso ele esteja cadastrado, se não, será impresso a seguinte mensagem: Produto não cadastrado com o código DDD. Onde DDD é o código fornecido.
3. Ele deve digitar o nome do produto e o programa exibirá os dados do mesmo, caso ele esteja cadastrado, se não, será impresso a seguinte mensagem: Produto não cadastrado com o nome XXXXX. Onde XXXXX é o nome fornecido.
4. Deve-se exibir a quantidade total de todos os produtos disponíveis em estoque.
5. O programa deve exibir o valor total em reais do estoque.
6. Finaliza o programa, que também deve ser encerrado quando os botões Cancelar e Fechar da caixa de diálogo do *menu* forem clicados.

Para desenvolver esse programa crie as seguintes classes:

Produto: representa as propriedades (atributos e métodos) de um produto.

Estoque: para gerenciar o estoque de produtos usando um vetor de objetos da classe Produto. Essa classe deve fornecer dois construtores: 1. o padrão, que cria um vetor para estocar no máximo 50 produtos; 2. um construtor sobrecarregado que cria um vetor com a capacidade de armazenamento (número máximo de produtos) especificada pelo usuário da classe. Essa classe deve implementar também os métodos abaixo.

```
/** Insere um produto no estoque. */  
public void inserir(Produto produto);  
  
/** Obtém um produto do estoque de acordo com o seu nome.  
 * Se o produto não for localizado no estoque retorna null. */  
public Produto obter(String nome);  
  
/** Obtém o número atual de produtos do estoque. */  
public int tamanho();  
  
/** Obtém a capacidade máxima de armazenamento do estoque. */  
public int capacidade();
```

ControleProdutos: implementa as funcionalidades do programa apresentadas acima.

8. Reescreva a classe Estoque do Exercício 7 substituindo o vetor de objetos por um `java.util.ArrayList` que armazena objetos da classe Produto. Observe que a refatoração¹ da classe Estoque deve ser feita sem gerar impactos para as classes usuárias. Em seguida, execute o programa

¹ Refatoração é uma técnica para melhorar a estrutura interna do código de um sistema de *software* sem alterar seu comportamento externo.

ControleProdutos criado no Exercício 7 usando a nova implementação da classe Estoque para testar o resultado da refatoração. O esperado é que a classe ControleProdutos não apresente nenhum erro se a refatoração for bem sucedida.

9. Escreva um método que recebe uma lista de argumentos de comprimento variável. Esse método, denominado `porExtenso`, deve receber uma lista de números inteiros e retornar um `java.util.List` de objetos `String` com cada posição da lista contendo uma *string* correspondente ao número escrito por extenso. Considerar apenas os números inteiros de zero a um milhão. A lista deve ser ordenada antes de ser retornada pelo método `porExtenso`. Use o método `sort` da classe `java.util.Arrays`. Para a ordenação desse método funcionar corretamente classifique os números inteiros e não as *strings*. Use a definição abaixo para implementar o método.

```
public static List<String> porExtenso(int... numeros);
```

10. Desenvolva um programa que receba vários números inteiros como argumentos no *prompt* de comandos do sistema operacional. Utilize o método `porExtenso` do Exercício 9 para implementar esse programa e exibir os números escritos por extenso.
11. Faça a refatoração da classe `Dicionario` criada no Exercício 4 da lista sobre Introdução a Classes e Objetos para alterar o tipo de dado da variável de instância de `StringBuilder` para `java.util.List`. Com essa modificação não é mais necessário o uso do caractere separador, pois cada palavra do dicionário deve ser armazenada em uma posição da lista de objetos do tipo `String`. Portanto, os construtores dessa classe serão padrão e um sobrecarregado para receber o tamanho (número de palavras) inicial do dicionário.
12. Use a classe `Dicionario`, refatorada no Exercício 11, para executar o programa de teste, chamado `DicionarioAurelio`, criado no Exercício 5 da lista sobre Introdução a Classes e Objetos.

Prof. Márlon Oliveira da Silva
marlon.silva@ifsudestemg.edu.br