

Aula: 13

Assunto: Switch-case, ternário, exercícios com vetores, matrizes e arraylists

Problemas Propostos:

Exercícios iniciais: valor 0,5 ponto

Resolva os exercícios desta seção para conquistar 0,5 ponto

1. (Prof. Fúlvio Prevot) Uma empresa possui 20 linhas com modems que podem ser conectados a um provedor de acesso. Escrever a classe Modems, que possui o atributo tempos, que é um vetor (array de double), usado para armazenar o tempo gasto por cada modem, e o atributo custo (double), que armazenará o custo, em reais, por minuto de transmissão. A classe também tem os seguintes métodos:

- a) Um método construtor que instancia o atributo tempos com tamanho 20 e que inicia o atributo custo com um valor passado por parâmetro (lembrar que o atributo custo não pode armazenar valor negativo);
- b) Um método que, usando o método showDialog(), lê e armazena, em cada posição do array, os tempos (double) gastos por cada modem (lembrar que o vetor, neste caso, não deve armazenar valores negativos);
- c) Um método, que retorna em uma String, os valores armazenados no vetor;
- d) Um método que calcula e retorna o tempo gasto total em minutos;
- e) Um método que calcula e retorna o custo total em reais;
- f) Um método que calcula e retorna o tempo médio gasto;
- g) Um método que retorna a posição (número inteiro) do vetor que contém o maior tempo gasto;
- h) Um método que retorna a diferença entre o maior e o menor tempo gasto.

Em seguida, escrever outra classe com o método main(), com a finalidade de testar os métodos criados na classe Modems. Para entrada e saída de dados, usar os métodos showDialog() e showMessageDialog() da classe JOptionPane.

OBS.: Não crie novos atributos na classe Modems.

Exercícios intermediários: valor 0,5 ponto

Resolva os exercícios desta seção para conquistar mais 0,5 ponto

2. (Prof. Fúlvio Prevot) Desenvolver um aplicativo, com no mínimo duas classes, em linguagem Java, para corrigir provas de múltipla escolha. Assumir que cada prova tem dez questões e que cada questão vale um ponto. O aplicativo deve ler o gabarito e o número de provas a serem corrigidas. A seguir, devem ser lidos os dados das provas a serem corrigidas. Ao final, o aplicativo deve calcular e mostrar:

- a) a nota de cada aluno;
- b) a percentagem de alunos aprovados, assumindo que a nota mínima para aprovação é 7 (sete);
- c) a média da turma.

OBS.: Não crie novos atributos.

Use o bloco try-catch onde for necessário. Faça o possível para desenvolver classes e métodos coesos e reduzir a quantidade de acoplamentos entre classes. Para entrada e saída de dados, usar os métodos `showInputDialog()` e `showMessageDialog()` da classe `JOptionPane`.

Exercícios complementares (para praticar)

Resolva os exercícios desta seção para aprimorar seus conhecimentos

3. (Prof. Fúlvio Prevot) Escrever a classe `Matriz2x2`, cujo atributo único é um array (matriz) de `double`. Escrever também, nesta classe:

- a) Um método construtor, sem parâmetros, que instancie o array (atributo), de modo que ele tenha duas linhas e duas colunas;
- b) Um método que, usando o método `showInputDialog()`, leia e armazene, em cada posição do array, números reais;
- c) Um método, que retorna em uma `String`, os valores armazenados no vetor;
- d) Um método que calcula e retorna o valor do determinante da matriz;
- e) Um método que determine e retorne a matriz transposta da matriz do atributo;
- f) Um método que retorne `true`, se a matriz do atributo é uma matriz simétrica (uma matriz simétrica é aquela que é igual à sua matriz transposta), caso contrário, retorna `false`;
- g) Um método que retorne `true`, se a matriz do atributo é uma matriz identidade (uma matriz identidade é aquela em que os elementos da diagonal principal são todos iguais a 1 e os demais elementos são iguais a zero), caso contrário, retorna `false`;

- h) Um método que recebe por parâmetro uma matriz com as mesmas dimensões que a matriz do atributo e que retorna outra matriz, que contém o resultado da soma matricial entre a matriz passada por parâmetro e a matriz do atributo;
- i) Um método que retorna uma matriz, que contém o resultado da multiplicação matricial (linha por coluna) entre a matriz do atributo e uma matriz (com as mesmas dimensões da matriz do atributo) passada por parâmetro para este método.

Em seguida, escrever outra classe com o método `main()`, com a finalidade de testar os métodos criados na classe `Matriz2x2`. Para entrada e saída de dados, usar os métodos `showInputDialog()` e `showMessageDialog()` da classe `JOptionPane`.

4. (Prof. Fúlvio Prevot) Uma empresa que possui 8 lojas trabalha com 11 tipos de produtos. Escrever a classe `Empresa`, com os seguintes atributos:

- a) um Array bidimensional (matriz) de double quantidades, sendo que cada linha representa uma das lojas, cada coluna um produto e cada célula contém a quantidade de cada produto por loja;
- b) um Array simples (vetor) de double preços, sendo que cada linha representa um produto e cada célula contém o preço unitário (em reais) de cada produto.

Escrever na classe `Empresa` os seguintes métodos:

- a) Um construtor que inicie os atributos;
- b) Um método para a entrada dos dados da matriz quantidades;
- c) Um método para a entrada de dados do vetor preços;
- d) Um método que calcule e retorne, em uma `String`, a quantidade total de estoque de cada produto em todas as lojas;
- e) Um método que retorne o valor total (em reais) do estoque.

A seguir, escrever outra classe com o método `main()`, a fim de testar os métodos da classe `Empresa`. Para entrada e saída de dados, usar os métodos `showInputDialog()` e `showMessageDialog()` da classe `JOptionPane`.

5. (Profa. Liliane) Uma faculdade, precisando gerenciar seu processo seletivo, contratou sua empresa para implantar um novo sistema de controle de vestibular. Como a tarefa foi atribuída a você, escreva as seguintes classes:

A. Faculdade, que possui um vetor de `Strings` com o nome dos cursos e uma matriz de inteiros que contém a quantidade de vagas disponíveis em cada um dos cursos para os três turnos (ou seja, seria uma matriz na qual as linhas representam os cursos disponíveis e as colunas representam os turnos do dia – manhã, tarde e noite).

Vetor:

Ciência da Computação

Sistemas de informação

Engenharia da Computação

Matriz:

50	0	75
50	100	0
100	100	100

- a. Crie um construtor padrão Faculdade que apenas chama o método setVagas();
- b. Um método setVagas, que não recebe parâmetros e que executa as seguintes ações:
- i. Pede que o usuário digite a quantidade de cursos da faculdade
 - ii. Preenche o vetor de cursos com o nome de cada curso
 - iii. Preenche a matriz de vagas perguntando, para cada curso, a quantidade de vagas para cada um dos três turnos. Por exemplo: Digite a quantidade de vagas do curso de computação no turno da manhã.
- c. Um método int busca, que recebe um nome de curso como parâmetro e retorna sua posição no vetor de cursos ou -1 se não encontrar.
- d. Um método vagas, que recebe dois parâmetros, um do tipo String referente ao nome do curso, e outro inteiro, referente ao turno do curso (0 – manhã, 1 – tarde e 2 – noite). Esse método deve retornar um inteiro, referente à quantidade de vagas do curso no turno, ou retorna -1 caso o curso não seja encontrado.
- e. Um método relacaoCandidatosPorVaga, que recebe como parâmetro um ArrayList de Candidatos, o nome de um curso e um inteiro referente ao turno e retorna um double com a relação candidatos por vaga.
- B. Candidato, que deve ter como atributos um cpf (String), curso desejado (String) e turno (int), seu construtor e métodos modificadores e de acesso.
- C. Crie a classe Vestibular, que possui um método main, que preenche um ArrayList de candidatos, cria uma classe Faculdade e preenche os seus atributos. O ArrayList

de candidatos deve ser preenchido através de um loop até que a resposta do usuário seja “fim”. Preenchidos o ArrayList e os atributos da Faculdade (cursos e vagas disponíveis em cada um dos turnos), a classe deve receber do usuário um curso e seu respectivo turno para o cálculo de candidatos por vaga.

Bibliografia

LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.

DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice – Hall (Pearson), 2010.