Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Programação Orientada a Objetos
Professor: Marco Antonio M. Carvalho

Lista de Exercícios 07 – Programando em Java

Instruções

- *Todos os exercícios que envolvem programas devem ser resolvidos por programas em linguagem **Java**;*
- *Na solução dos exercícios, devem ser utilizados os conceitos listados no cabeçalho desta lista;*
- *Eventuais dúvidas podem ser sanadas com o professor.*

1. Crie um programa que leia 3 inteiros a partir do teclado e determina:
    a. O maior;
    b. O menor;
    c. O produto;
    d. A média.

2. Crie um programa que leia dois inteiros e determine se o primeiro é um múltiplo do segundo.

3. Crie um programa que leia o raio de um círculo e imprima seu diâmetro, área e circunferência. Utilize a constante PI definida na classe ***Math***.

4. Crie um programa que leia um caractere e o imprima na tela, juntamente com seu código na tabela ASCII, que deve ser determinado em tempo de execução. Utilize **caixas de diálogo** para realizar a entrada e saída de dados.

5. Crie um programa que armazene 12 números em um vetor alocado **dinamicamente** e determine qual a porcentagem de números menores que 8 e qual a porcentagem de números maiores que 10.

6. Crie um programa que **aloque dinamicamente** e leia uma matriz 10x10 de caracteres e, através de uma função, imprima todos os elementos, exceto os da **diagonal secundária**.

7. Crie uma classe chamada *Conta*, que represente contas bancárias. A classe deve conter como atributo o saldo da conta (um número real). A classe deve possuir um construtor que recebe o saldo inicial para inicializar o atributo, validando se o valor é maior ou igual à zero. Caso o valor seja menor que zero, o atributo deve ser inicializado com zero, e uma mensagem de erro deve ser apresentada. Crie um programa que contenha dois objetos desta classe e utilize cada um dos três métodos:

    a. *Credito*: adiciona um valor ao saldo atual;
    b. *Debito*: subtrai um valor do saldo atual, garantindo que o saldo não ficará negativo. Se o débito for maior que o saldo, a operação não deve ser realizada e uma mensagem apresentada;
    c. *getSaldo*: retorna o saldo.

8. Crie uma classe chamada *NotaFiscal* que um *hardware* utilize para representar uma nota fiscal em uma loja de peças. Uma nota fiscal deve incluir quatro dados como atributos:

a. Número da peça (*string*);
b. Descrição da peça (*string*);
c. Quantidade comprada (inteiro);
d. Preço (número real).

A classe deve incluir *getters* e *setters* para cada um dos atributos. Ainda, deve haver um método *getTotalNota* que calcule e retorne o total de um vetor de objetos. Escreva um programa que teste cada um dos métodos da classe.

9. Crie uma classe *Data* que inclua três atributos: mês (inteiro), dia (inteiro) e ano (inteiro). Crie métodos para:
   a. Funcionar como *getter* e *setter*, validando os dados para garantir que os valores são reais;
   b. Funcionar como um construtor (com parâmetros padronizados) com três parâmetros para inicialização dos três atributos;
   c. Imprimir a data, com os campos separados por /;
   d. Calcular a quantidade de dias do ano até aquele mês, recebendo como parâmetro o número do mês;

   Escreva um programa que teste todos os métodos da classe. Para simplificar o exercício, considere meses ímpares com 31 dias e meses pares com 30 dias.

10. Crie uma classe *Aluno* com atributos que armazenem o nome, a série que cursa e o grau. Crie um *getter* e um *setter*, além de uma variável *static* que conte a quantidade de objetos criados, a ser utilizada no construtor e no finalizador. Crie um vetor com 15 objetos desta classe e preencha os dados através do *setter*.

11. Crie uma classe *MoedaViciada*, que simula um cara ou coroa. Seu programa deve ler qual é a opção do usuário (cara ou coroa), e simular a jogada da moeda. No entanto, a opção do usuário só deve ter 30% de chances de ocorrer. Deve haver um método para cada opção. Utilize a classe ***Random***.