

## LISTA DE EXERCÍCIOS DE FIXAÇÃO – 03

Notas: - Pode ser utilizado como ajuda os livros contidos na bibliografia básica da disciplina.

- 1) (Baseado em 2.24 – DEITEL, 2005) Crie uma classe **BuscaMaiorMenor** que lê cinco inteiros e determina o maior e o menor inteiro no grupo. A classe deve conter a seguinte estrutura:

- `n1 (int), n2 (int), n3 (int), n4 (int), n5 (int);`
- `maior – int;`
- `menor – int;`
- Um método **set** e um **get** para cada variável de instância (Obs.: Os métodos **setMaior** e **setMenor** para declaração das variáveis `maior` e `menor`, respectivamente, devem ser calculados a partir das variáveis `n1`, `n2`, `n3`, `n4` e `n5`).
- Deve possuir um **construtor** que inicializa todas as variáveis (Obs. Utilização dos métodos **setMaior** e **setMenor** para inicialização das variáveis `maior` e `menor`, respectivamente).

Crie uma classe **BuscaMaiorMenorTeste** que instancie um objeto da classe **BuscaMaiorMenor**, e demonstre as capacidades da classe **BuscaMaiorMenor**. Todas as entradas de valores de variáveis e saída de mensagens devem ser realizadas dentro da classe **BuscaMaiorMenorTeste**.

- 2) Crie uma classe chamada **Retangulo** que tenha a capacidade de armazenar os seguintes valores (variáveis de instância):

- `x – int (valor da altura);`
- `y – int (valor da largura);`
- `area – int (Área total do Retângulo);`
- Um método **set** e um **get** para cada variável de instância (Obs.: O método **setArea()** para declaração da área deve ser calculado a partir das variáveis `x` e `y`).
- Deve possuir um construtor que inicializa todas as variáveis (Obs. Utilização do método **setArea** para inicialização da variável `area`) e o construtor padrão.

Crie uma classe **RetanguloTeste** que instancie um objeto da classe **Retangulo**, e demonstre as capacidades da classe **Retangulo**. Todas as entradas de valores de variáveis e saída de mensagens devem ser realizadas dentro da classe **RetanguloTeste**.

- 3) Um motorista monitorou vários tanques cheios de gasolina do seu carro registrando a quilometragem dirigida e a quantidade de litros utilizados para cada tanque cheio. Crie uma classe chamada **ConsumoMedio**, que possibilita armazenar o valor de três quilometragens e seus respectivos valores de gasto de combustível (ambos como inteiros). Crie também 4 variáveis de instância (`con1`, `con2`, `con3`, `cont`) do tipo ponto flutuante (`float`) para armazenamento do consumo (`Km/l`). Forneça também os seguintes métodos:

- **CalculaConsumo1**: Calcula o consumo `Km/l` do 1º abastecimento e armazena na variável `con1`;
- **CalculaConsumo2**: Calcula o consumo `Km/l` do 2º abastecimento e armazena na variável `con2`;

- CalculaConsumo3: Calcula o consumo Km/l do 3º abastecimento e armazena na variável con3;
- CalculaConsumoTotal: Calcula a média de consumo de todos os abastecimentos e armazena na variável cont;
- Métodos **get** e **set** para todos os atributos;
- Pelo menos 2 construtores.

Crie uma classe **ConsumoMedioTeste** para realização de testes da classe ConsumoMedio.

- 4) Crie uma classe **Poupanca**. Utilize uma variável private jurosAnuais para armazenar a taxa de juros anual e uma variável de instância private balanço para indicar a quantidade que o poupador tem em depósito. Forneça um método calcularJurosMensais para calcular os juros mensais multiplicando balanço por jurosAnuais dividido por 12 – esses juros devem ser adicionados pelo mesmo método ao balanço. Forneça um método setJurosAnuais que configure o valor jurosAnuais para um novo valor. Escreva um programa para **testar** a classe Poupança. Instancie dois objetos Poupança, saver1 e saver2 com saldos de R\$2.000,00 e R\$3.000,00, respectivamente. Configure jurosAnuais como 4% e então calcule o juro mensal e imprima os novos saldos para os dois poupadores. Em seguida, configure o jurosAnuais para 5%, calcule a taxa do próximo mês e imprima os novos saldos para os dois poupadores.
- 5) Crie uma classe LimiteCredito que determinará se um cliente de uma loja de departamentos excedeu o limite de crédito em uma conta-corrente. A classe deve conter as seguintes variáveis de instância (todas do tipo inteiro):
  - Número da conta;
  - Saldo no início do mês;
  - Total de todos os itens cobrados desse cliente no mês;
  - Total de créditos aplicados ao cliente no mês;
  - Limite de crédito autorizado.Forneça os seguintes métodos:
  - Métodos get e set para todos os atributos;
  - CalculaNovoSaldo: Calcula o novo saldo (=saldo inicial + despesas - créditos) e exibe o novo saldo. Se o limite de crédito for excedido, o programa deve exibir a mensagem “Limite de crédito excedido!”.

Crie uma classe LimiteCreditoTeste para demonstrar as funcionalidades da classe LimiteCredito.