

LISTA DE EXERCÍCIOS DE FIXAÇÃO – 02

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

- 1) Apesar do aluguel de Joãozinho ter sido pago por um ano, estava complicado para ele receber o dinheiro que o pai enviava por mês para cobrir suas despesas pessoais que incluíam alimentação, lavanderia, xerox, festas, etc. Joãozinho, para resolver logo esse problema, procurou uma agência bancária para abrir uma conta em seu nome. Ao chegar à agência ele esperou 50 minutos para finalmente ser atendido. O funcionário que o atendeu pediu que Joãozinho fornecesse algumas informações pessoais para a abertura da conta: nome completo, endereço residencial, número do rg, número do cpf, e qual o tipo de conta que gostaria de abrir (Conta corrente ou Conta Poupança). Depois de ter seu pedido de abertura de conta aprovado, Joãozinho recebeu o número de sua conta e o número da agência a qual ela estava vinculada. Na conta aberta por Joãozinho só era possível realizar três operações básicas: saque, depósito e impressão de saldo. Lembrando que, se Joãozinho tentasse efetuar a operação de saque com saldo insuficiente, o sistema não aprovaria a transação e apresentaria uma mensagem de erro. Joãozinho que é um garoto muito curioso logo quis testar todas as funcionalidades de sua conta efetuando um depósito e depois um saque. Antes e depois de cada operação ele tirou o saldo da conta. Escreva uma aplicação Java que reproduza essa situação usando os conceitos de classe (atributos - métodos) e encapsulamento vistos em sala de aula. *Obs:* A implementação deve conter no mínimo duas classes.
- 2) O site **Veja** lançou uma notícia de que o governo dos EUA estuda a viabilidade de acabar com a exigência de vistos para brasileiros que queiram entrar em território americano. De acordo com a repórter Mariana Pereira “entre os países que mais requisitam vistos de entrada para os Estados Unidos, o Brasil só fica atrás de China, Índia e México. São 300.000 solicitações por ano e o índice de aprovação é atualmente de 95%.”

Mariazinha ao ver essa notícia ficou toda empolgada, pois seu maior sonho é conhecer a Disneylândia. Mais do que depressa ela correu atrás de Joãozinho e pediu para que ele comprasse uma passagem para ela. Joãozinho não querendo estragar a empolgação de Mariazinha concordou em comprar a passagem usando uma aplicação que ele mesmo desenvolveu. Nessa aplicação ele selecionava a operação de compra de passagem aérea que solicitava as seguintes informações: cidade e país de origem; cidade e país de destino; dia, mês e ano da data de ida; dia, mês e ano da data de volta, nome completo e cpf do passageiro; forma de pagamento (cartão de crédito ou boleto bancário). Ao fornecer todos os dados o sistema apresentava na tela um texto no seguinte formato:

“Obrigado por escolher a nossa agência aérea”
Origem: cidade - país Destino: cidade – país
Valor: R\$ XXX,XX
Nome do passageiro: XXX XXXXX XXXX
CPF: XXX.XXX.XXX-XX
Data de ida: DD/MM/AAAA
Data de volta: DD/MM/AAAA
“Desejamos que tenha um excelente vôo!!!”

Escreva uma aplicação Java que reproduza essa situação usando os conceitos de classe (atributos - métodos) e encapsulamento vistos em sala de aula. *Obs:* A implementação deve conter no mínimo duas classes.

- 3) (3.13 – DEITEL, 2005) Crie uma classe chamada *Fatura* para que uma loja de suprimentos de Informática possa utilizá-la para representar uma fatura de um item vendido na loja. Uma fatura deve incluir quatro partes das informações como variáveis de instância: o número (tipo *String*), a descrição (tipo *String*), a quantidade comprada de um item (tipo *int*) e o preço por item (*double*). Sua classe deve ter um método *get* e um *set* para cada variável de instância. Além disso, crie um método para calcular o valor da fatura (isto é, um método que multiplique a quantidade pelo preço por item) e depois retorna o valor como um *double*. Se a quantidade não for positiva, ele deve ser configurada como 0. Se o preço por item não for positivo, ele deve ser configurado como 0.0. Escreva um aplicativo de teste chamado *FaturaTeste* que demonstra as capacidades da classe *Fatura*.
- 4) (3.14 – DEITEL, 2005) Crie uma classe chamada *Empregado* que inclui três partes de informações como variáveis de instância – um primeiro nome (tipo *String*), um sobrenome (tipo *String*) e um salário mensal (*double*). Forneça um método *set* e um método *get* para cada variável de instância. Se o salário mensal não for positivo, configure-o como 0.0. Escreva um aplicativo de teste chamado *EmpregadoTeste* que demonstra as capacidades da classe *Empregado*. Crie dois objetos *Empregado* e exiba o salário anual de cada objeto. Então dê um aumento de 10% e exiba novamente o salário anual de cada *Empregado*.
- 5) (3.15 – DEITEL, 2005) Crie uma classe chamada *Data* que inclui três informações como variáveis de instância: um mês (tipo *int*), um dia (tipo *int*) e um ano (tipo *int*). Forneça um método *set* e um *get* para cada variável de instância. Forneça um método *exibeData* que exibe o mês, o dia e o ano separados por barras normais (/). Escreva um aplicativo de teste chamado *DataTeste* que demonstra as capacidades da classe *Data*.