

## Exercícios de Java Swing

1. Crie um programa utilizando a interface java swing que possua uma janela de login onde permita o usuário alternar de uma janela para uma segunda janela, também criada pelo usuário, e essa segunda janela deve ficar na posição 50x350 definida no código do programa. Também deverá ser usado Labels, áreas de texto e Botões para desenvolvimento de sua interface.(PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006.)
2. Utilizando o exercício anterior como exemplo faça um programa que além dos componentes citados anteriormente utilize os gerenciadores de layout *FlowLayout*, *GridLayout*, *BorderLayout* e *GridBagLayout* para setar *JLabels* na segunda tela do programa. O Programa também pode ter mais de duas telas.(PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006).
3. Faça um programa que utilize os seguintes componentes swing: *Jbutton*, *JToggleButton*, *JRadioButton*, *JCheckBox* e *JComboBox* para desenvolvimento de um programa para um estúdio de música que grava músicas de vários cantores de vários ritmos diferentes. Armazenar os dados em memória usando Java Collections Framework (JCF) (PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006) e (DEITEL, Harvey M. JAVA Como Programar, São Paulo: Deitel, 2010).

**Para as questões a seguir, utilizar o gerenciador de layout GridBagLayout.**

4. Construa um programa para conversão de valores monetários em valores correspondente em diferentes moedas. A interface do programa deve ter 3 partes:
  - a) um campo para entrada de um número de ponto flutuante, correspondente a um valor em reais;
  - b) um grupo de botões para seleção da moeda para a qual o valor em reais deve ser convertido: quando o usuário selecionar um botão, o método de conversão monetária correspondente deve ser chamado.

c) um campo onde é mostrado o resultado da conversão monetária solicitada.

(CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

5. Implemente uma interface para desenvolver uma minicalculadora onde a primeira parte deve conter um campo para entrada de dois valores numéricos (de tipo double), que são passados como argumentos em chamadas às funções da minicalculadora. A segunda parte deve conter os botões, um para cada função da minicalculadora. Quando o usuário pressiona um botão, é chamada a função correspondente, com os argumentos apropriados, os quais são especificados no campo de texto. A terceira parte contém um campo onde é mostrado o resultado de uma chamada a uma função da minicalculadora.(CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

a) Modifique o programa da minicalculadora de modo a incluir um botão “Limpar”, que apaga a entrada no campo reservado para os argumentos das funções.

b) Modifique o programa da minicalculadora de maneira que, ao ser selecionada uma determinada função, seja feito um teste para determinar se cada um dos argumentos especificados pelo usuário é realmente um valor numérico. Caso isso não se verifique, deve ser emitida uma mensagem apropriada. Use o método `showMessageDialog`, da classe `JOptionPane`, para emitir essa mensagem.

c) Modifique o programa da minicalculadora para que a interface inclua um botão correspondente a cada algoritmo, como alternativa para entrada dos valores numéricos de argumentos de uma função. Modifique também o programa de modo que a entrada de dados seja feita como em uma calculadora usual, isto é, usando o operador de forma infixada, por exemplo,  $(a-b)*c$ .

6. Construa um programa para ajudar um estudante de primeiro grau a ganhar maior destreza no cálculo das quatro operações aritméticas, com operadores inteiros entre 0 e 99. A operação de divisão deve retornar um quociente e o resto da divisão inteira de dois números. Use a função `Math.random`, para obter dois números

inteiros positivos, cada qual representado com até dois algarismos. O seu programa deve então exibir uma pergunta para o estudante como, por exemplo:  
Quanto é 14 dividido por 3?  
(digite o quociente e o resto, separados por espaços e/ou vírgula).

O estudante deve fornecer a resposta em um campo de texto (JtextField). Como indicado no caso da divisão, a resposta deve consistir do quociente e do resto, nessa ordem, separados por um ou mais espaços, ou por uma vírgula e zero ou mais espaços. O programa deve verificar se a resposta dada está correta ou não. No primeiro caso, deve emitir a mensagem “Muito bem amiguinho!”, e no segundo caso, a mensagem “Resposta errada amiguinho. Tente novamente”. Nesse último caso, deve permitir que o estudante tente uma nova resposta repetidamente, até acertar. (CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

7. Construa um interface que implemente uma interface simplificada para aposta em uma miniloteria esportiva, em que um cartão é constituído de 5 jogos. A interface deve conter um título ----”Loteria Esportiva”----, um campo para entrada do nome do apostador e, para cada jogo, um rótulo com a identificação dos dois times e um grupo de três botões ----”1”, “x” e “2”---- representando as opções para o resultado do jogo (respectivamente, time 1 vencedor, empate e time 2 vencedor). Apenas uma dessas opções pode ser selecionada para cada jogo. (CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)
8. Implemente uma interface para apostar na seleção vencedora da Copa do Mundo. Deve-se criar um evento para selecionar um país onde a bandeira de cada país deve ser exibida no mesmo instante e na mesma janela. Isso pode ser feito alterando-se o ícone associado ao rótulo inserido na janela, por meio de uma chamada ao método setIcon, da classe JLabel. O ícone a ser mostrado deve ser obtido a partir de um arquivo .gif, de nome igual ao da seleção de futebol escolhida. Um objeto da classe JComboBox implementa a escolha de uma única opção, dentre as possibilidades apresentadas. Para mostrar uma lista de itens, dos quais um ou mais podem ser escolhidos, pode ser usado um objeto da classe Jlist. Implemente também na interface um campo de texto para entrada do nome do

apostador, e outro campo de texto, para entrada do valor da aposta. (CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

9. Utilize os componentes: *JTextField*, *JTextArea*, *JList*, *JScrollPane* e *TableModel* para desenvolvimento de um programa que cadastra pessoas e suas idades e salva todas elas em uma tabela. Armazenar os dados em componentes JCF (PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006.) e (DEITEL, Harvey M.. JAVA Como Programar, São Paulo: Deitel, 2010).

10. Construa um programa para processar pedidos de compras de produtos de uma empresa exportadora. A interface desse programa deve conter as seguintes partes:

- a) um campo de texto para entrada do nome do cliente;
- b) um campo de texto para entrada do endereço do cliente;
- c) um menu de opções para seleção do país de origem do pedido;
- d) um menu de opções para seleção de um produto, que informe também o preço desse produto por unidade de venda;
- e) um campo de texto para entrada da quantidade desejada do produto, expressa em unidades de venda desse produto;
- f) um campo para exibir a lista dos produtos selecionados, com a quantidade pedida de cada um e o preço total correspondente
- g) dois botões ----"Confirma" e "Cancela"----, para confirmação e cancelamento do pedido, respectivamente;

Quando o botão de confirmação for pressionado, o programa deve exibir uma janela informando os dados do pedido: a lista dos produtos solicitados, com a quantidade, preço unitário e preço total de cada produto, o valor da tarifa de transporte e o custo total do pedido (incluída a tarifa de transporte). (CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

11. Construa uma interface que implemente um terminal de agência bancária simplificado, provendo as seguintes funções:

- a) abertura de conta;
- b) fechamento de conta;

- c) solicitação de saldo;
- d) saque;
- e) depósito.

Para realizar uma operação de abertura ou fechamento de conta, o usuário deve fornecer uma senha especial (conhecida apenas pelos funcionários da agência), a qual deve ser verificada pelo programa. Para uma operação de solicitação de saldo, saque, ou de depósito, o usuário deve informar o número da conta bancária e a senha de acesso correspondente. Utilizar JCF para armazenar os dados em memória (CAMARÃO, Carlos. Programação de Computadores em JAVA. Rio de Janeiro: LCT, 2003.)

12. Especifique um programa que apresente ao usuário um velocímetro e permita que ele incremente ou decrémente o valor mostrado pelo velocímetro. O velocímetro deve mostrar valores inteiros a ser inicializado com o valor 100. Utilize os conceitos de Interface abordados até o momento. (PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006.)
13. Desenvolva o seguinte programa (PINHEIRO, Francisco A. C.. Fundamentos de Computação e Orientação a Objetos Usando JAVA. Rio de Janeiro: LCT, 2006.) e (DEITEL, Harvey M.. JAVA Como Programar, São Paulo: Deitel, 2010):
  - a) Faça um programa que mostre duas janelas em sequencia do método de execução *showMessageDialog*, a primeira deverá mostrar uma mensagem do tipo *WARNING\_MESSAGE* e a segunda apenas uma mensagem de termino.
  - b) Utilize o método *showConfirmDialog* para criar um programa implementado junto com o exercício anterior.
  - c) Utilizando as questões anteriores desenvolver um programa que implementa a entrada de valores informados pelo usuário através do método *showInputDialog*.
  - d) Por ultimo faça a implementação de um programa utilizando os exercícios anteriores e implementado um novo método de *showOptionDialog*.