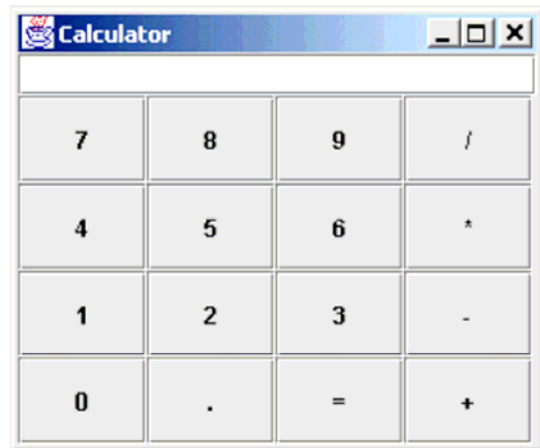


Exercícios de Java - Swing

“Componentes e Montagem de Interfaces e Eventos”

1. Implemente a seguinte calculadora.



2. Escreva um programa que jogue “adivinha o número”: o programa escolhe um número a ser adivinhado, selecionando-o randomicamente no intervalo de 1-1000. O programa mostra uma mensagem (**JLabel**) – Eu tenho um número entre 1 e 1000, você pode adivinhá-lo? Entre com seu chute.

- Um **JTextField** deve ser usado na captura do valor. Para cada tentativa de adivinhação a cor de fundo deve mudar ou para vermelho ou azul. Vermelho indica que o usuário está chegando mais perto e azul indica que o usuário está se afastando do número. Um **JLabel** deve mostrar ou “**Mais Perto**” ou “**Mais Longe**” como auxílio ao usuário.
- Quando a resposta estiver correta, você deve mostrar a frase “**Correto!**” e não permitir mais a edição no **JTextField**.
- Um **JButton** deve permitir um novo jogo. Quando o **JButton** é clicado, um novo número randômico deve ser gerado e o **JTextField** ser editável.

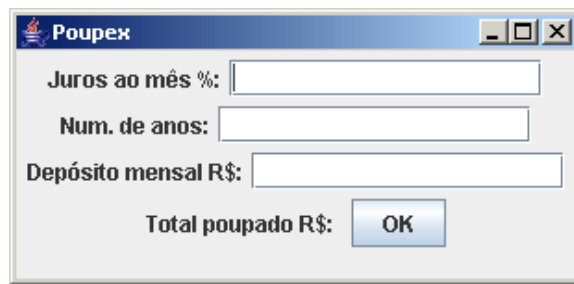
3. Escreva um programa capaz de converter o valor em Reais fornecido numa caixa de entrada (TextField) em outro valor monetário conforme a seleção de uma moeda (no mínimo 3 moedas diferentes) numa caixa de seleção (Combo) indicando o resultado em um rótulo (Label). Se necessário adicione botões.

4. Escreva uma aplicação capaz de calcular o valor futuro de um montante corrigido a taxa de juros fixa, sem depósitos adicionais por um período de n meses. A valor inicial, a taxa de juros (valor real) e o número de meses (valor inteiro) devem ser fornecidos pelo usuário através de caixas de entrada (TextField). Os cálculos devem ser exibidos num rótulo (Label) apenas depois do acionamento de um botão (Button).

5. Declare uma classe Java que modela um objeto Livro, que contém ainda um método construtor. Um livro possui um título e quantidade de páginas. Use Strings para representar o título. Use inteiros para representar a quantidade de páginas. O construtor deve receber através de argumentos, os dois dados suficientes para criar um livro. Codifique uma classe chamada LivroGUI, que constrói uma interface gráfica que manipula informações do modelo de dados proposto. A GUI deve ser capaz de criar e listar Livros. A GUI é composta por um frame que contém pelo menos dois labels (“Título:” e “Páginas:”), dois textfield (para coletar informações sobre um livro) e ainda dois botões, chamados “Adicionar” e “Listar”.

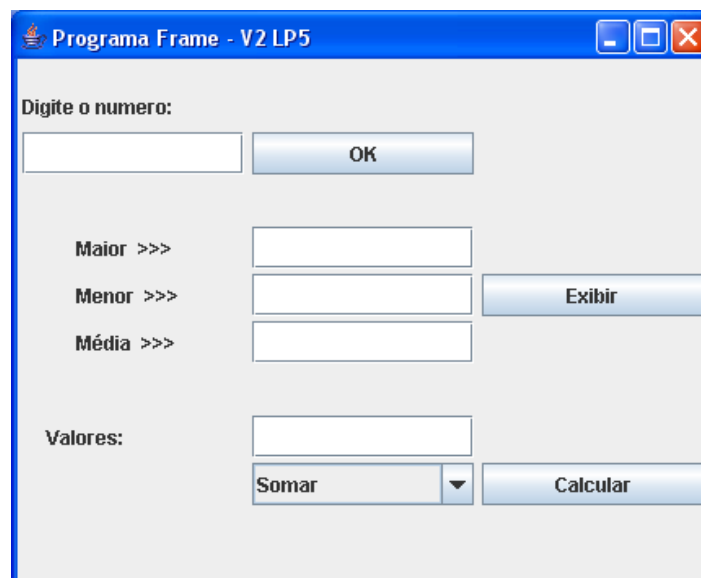
- Ao ser pressionado, o botão “Adicionar” cria um novo livro a partir das informações contidas nos TextField. O novo livro criado deve ser adicionado à próxima posição livre de um array de livros de tamanho 200.
- Ao ser pressionado, o botão “Lista”, imprime as informações de todos os livros já cadastrados em um JList.

6. Implemente a aplicação abaixo:



7. Construa um programa visual Java (JFrame) que possua a interface segundo mostra a figura abaixo. Pede-se implementar as seguintes funcionalidades:

Construir a seguinte Interface:



O usuário deste JFrame deverá entrar com vários números em quantidade indeterminada, através do botão OK, onde os mesmos serão cadastrados. Feito isso, o programa terá as seguintes funções:

- ✓ No botão EXIBIR, devem aparecer os seguintes resultados:
 - O menor valor daqueles digitados será mostrado no campo indicado por MENOR
 - O maior valor daqueles digitados será mostrado no campo indicado por MAIOR
 - A média aritmética dos números digitados será mostrada no campo indicada por MÉDIA
- ✓ A segunda parte deste JFrame se refere ao botão CALCULAR. No campo VALORES, o usuário deverá digitar uma sequência de 12 números, seguidamente sem espaço entre eles. Além disso, o usuário deverá escolher uma opção dentre aquelas disponíveis no ComboBox, que são: "Somar" e "Multiplicar". Ao clicar o botão CALCULAR, o programa deverá "recortar" dois a dois os números (use o método *substring* da classe String), e verificar a opção escolhida. Se for "Somar", o programa deverá soma-los (não esquecendo da conversão) e exibir esta soma. Da mesma forma, com a opção "Multiplicar", só que neste caso, multiplica-se os valores.

Exemplo:

Valores = 276893451626 Opção = "Somar"
Soma = 27 + 68 + 93 + 45 + 16 + 26 => **275**

Ou

Valores = 276893451626 Opção = "Multiplicar"
Soma = 27 x 68 x 93 x 45 x 16 x 26 => **3196402560**

8. Monte uma classe Aluno para atender à interface gráfica fornecida a seguir. Os atributos serão aqueles atendidos pela interface e os métodos serão todos aqueles que julgar necessário. A aplicação trata do cadastramento de alunos para uma instituição privada de cursos. Portanto, para atender às operações de “Inserir”, “Gravar” e “Consultar”, você deverá ter uma classe BDAlunos que implemente uma Collection para agrupar um conjunto de objetos Alunos e manipula-los, segundo o enfoque dos botões no formulário.

Componentes de Interface Gráfica

Nome:

Endereço:

Cidade: Estado: PR

Sexo: ☒ Masculino ☐ Feminino

Opções: ☐ Curso 1 ☐ Curso 2 ☐ Curso 3

Interesses

Redes
Internet
Compiladores
Segurança
BD

Observações

Inserir
Gravar
Consultar
Sair

9. Fazer um programa Java que controla os carros de uma locadora. Cada carro possui: código, nome, marca, modelo, preço, valor de locação, dias de locação, valor total do aluguel.

Cada carro possui algumas operações, como:

1. Cadastrar carros (todos os carros são modelo sedan e da Fiat);
2. Alterar o valor de locação;
3. Exibir as informações do carro;
4. Calcular valor da multa;
5. Calcular valor total do aluguel;

Para calcular o valor da multa, o usuário informa a quantidade de dias locados e, se for maior que os dias de locação, a diferença desses dias, acrescidos de 15% será o valor da multa. O valor total do aluguel será o valor de locação X os dias da locação + valor da multa.

No programa principal:

- a. Cadastrar um acervo de N carros da locadora (vetor de objetos).
- b. Calcular o valor total de aluguel e o valor de multa de todos os carros (utilizando os métodos correspondentes)
- c. Permitir o usuário duas opções:
 - i. Mostrar as informações de todos os carros;
 - ii. Pesquisar um determinado carro pelo código, e caso encontrado, altere seu valor de locação;

Monte uma aplicação visual em Java com JFrames para dar suporte a essas operações descritas. Monte um JFrame somente, contendo um **JTabbedPane** contendo uma aba para cada operação acima (a, b, c, i e ii), ou seja, serão 5 ao todo. Seu formulário deverá também ter opções de Sair e Limpar Campos. Monte um Menu contendo a chamada a cada aba individualmente. Seu JFrame deverá, obrigatoriamente ter um JComboBox e JRadioButton.

10. Refaça o exercício pontuado de locadora feito em sala para atender a seguinte interface abaixo.

The image shows a graphical user interface for a DVD rental system, titled "Locadora de DVDs". The interface features a menu bar at the top with options: "Sócios", "DVDs", "Consulta Sócios", "Consulta DVDs", and "Sair". Below the menu bar, there are four tabs: "SÓCIOS", "DVDs", "Consulta Sócios", and "Consulta DVDs". The "Consulta DVDs" tab is currently selected. The main area of the interface contains a label "Digite o nome do filme" followed by a text input field and a button labeled "Consultar !". Below this, there is a section titled "Dados do DVD" which contains a large, empty rectangular area with a vertical scrollbar on the right and a horizontal scrollbar at the bottom, indicating a list or table of data.