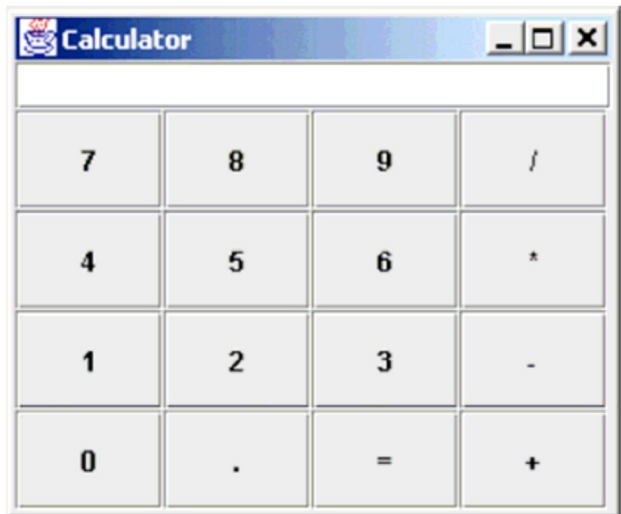


IESP

Professora: Dra Alana Moraes

Exercícios de Java - Swing

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. 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.

7. 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:

- Cadastrar um acervo de N carros da locadora (vetor de objetos).
- Calcular o valor total de aluguel e o valor de multa de todos os carros (utilizando os métodos correspondentes)
- 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.