

Para cada um dos problemas abaixo, desenvolva um programa na linguagem Java:

Tópicos: entrada/saída, operadores, estruturas de desvio de fluxo condicional

1. Calcular o IMC (índice de massa corporal) de uma pessoa. Produza um alerta em caso de divisão por zero.
2. Construa um programa que calcule a área de um círculo, tendo como entrada o valor do raio, que deve ser positivo. Use o valor da constante PI vindo da biblioteca matemática.
3. Construa um programa que leia um número inteiro e identifique se ele é par ou ímpar.
4. Construa um programa que simule uma transferência bancária, entre duas contas. Primeiro, inicialize cada conta com R\$ 100. Depois, permita que o usuário defina quanto deve transferir, da conta1 para a conta2, porém a transferência só deve ser realizada caso haja saldo suficiente.
5. Distinguir, com base na média parcial do aluno, se ele está aprovado, reprovado ou na final. Aplique as regras da UFERSA.
6. Construa um programa que calcule para o aluno sua média parcial e informe sua situação parcial (Aprovado, Recuperação ou Reprovado). Caso ele esteja em Recuperação, solicite a nota da 4ª prova, e informe a situação (Aprovado ou Reprovado). Utilize as regras da UFERSA.
7. Construa um programa que calcule uma equação do 2º grau. Crie um método com retorno para calcular o delta.
8. Construa um programa que leia um número inteiro digitado pelo usuário. Caso o número pertença ao intervalo de 1 a 5, exiba o número por extenso. Caso o número não pertença a este intervalo, exiba a mensagem "valor invalido".

Tópicos: métodos, estruturas de desvio de fluxo com repetição

9. Construa um programa que leia do usuário um número inicial e um número final. Em seguida, exiba na tela uma sequência com os números desse intervalo informado pelo usuário. Exemplo: caso o usuário entre com os números 4 e 10, o resultado do programa seria: 4 5 6 7 8 9 10
 - a) Construa este programa utilizando a estrutura **while**.
 - b) Construa este programa utilizando a estrutura **do-while**.
 - c) Construa este programa utilizando a estrutura **for**.
10. Construa um programa que leia do usuário um número inicial e um número final. Em seguida, exiba na tela uma sequência apenas com os números ímpares dentro deste intervalo informado pelo usuário. Exemplo: caso o usuário entre com os números 4 e 10, o resultado seria: 5 7 9
 - a) Construa este programa utilizando a estrutura **while**.
 - b) Construa este programa utilizando a estrutura **do-while**.
 - c) Construa este programa utilizando a estrutura **for**.
11. Verificar se a senha, informada durante a execução, é correta. Quando a senha estiver correta, exiba “senha correta” e o programa é encerrado. Quando a senha estiver errada, exiba “senha incorreta”, e permita a entrada novamente até que a senha informada seja correta.
 - a) Resolva esse problema utilizando a estrutura **while**.
 - b) Resolva esse problema utilizando a estrutura **do-while**.
12. Construa um programa para exibir a tabuada de qualquer número inteiro (1 a 9), fornecido pelo usuário. Utilize estrutura de repetição.
13. Construa um programa que calcule o rendimento mensal de um investimento em poupança. Variáveis: investimento inicial, investimento mensal, quantidade de meses, saldo acumulado, taxa de juros mensal, rendimento mensal.
14. Construa um programa que identifique se um número é primo.
15. Construa um programa com um método que calcule o somatório dos números inteiros de um intervalo, definido por um número inicial e um número final. Use uma estrutura de repetição. Exemplo: caso as entradas fossem 4 e 9, o resultado seria: 39
16. Construa um programa com um método que calcule o número fatorial de um número. Use uma estrutura de repetição. Exemplo: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ ou $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
17. Construa um programa que simule uma calculadora. Disponibilize um menu de opções e simule a opção desejada, exibindo novamente o menu, até que o usuário escolha sair. Menu de opções:
 - 1 - potenciação
 - 2 - raiz quadrada
 - 3 - fatorial
 - 0 - sair

Dica: utilize a função da biblioteca matemática.
Dica: utilize a função da biblioteca matemática.
Dica: utilize o método da questão anterior.

Tópicos: vetores, matrizes, strings

18. Construa um programa para ler e exibir um vetor de inteiros. Em tempo de execução, o usuário pode definir o tamanho do vetor.
19. Construa um programa para ler e exibir uma matriz de inteiros. Em tempo de execução, o usuário pode definir o tamanho da matriz.
20. Construa um programa e um método que produza um novo vetor de inteiros com a ordem inversa do vetor original passado por parâmetro.
21. Construa um programa e um método que inverta a frase digitada pelo usuário. Exemplo, se string1 for “bom dia”, string2 será “aí d’mob”.
22. Construa um programa que identifique o maior e o menor número de um vetor de inteiros.

Tópicos: orientação a objetos, arquivos

23. Construa um programa que registre objetos da classe Pessoa, com os seguintes atributos: cpf, nome, idade, sexo, peso, altura, imc. Permita que o usuário defina a quantidade de pessoas em tempo de execução. Ao cadastrar uma pessoa, calcule o IMC (Índice de Massa Corporal). Ao final, exiba a lista de pessoas (com seus respectivos atributos).
24. Evolua a questão anterior. Permita que o programa salve o resultado em um arquivo binário. Permita que o programa abra o arquivo binário e exiba os dados de cada objeto Pessoa na tela.
25. Construa um programa que conte a quantidade de letras de uma palavra.
Exemplo: “casa” Resultado: c=1 a=2 s=1
26. Construa um programa que conte a quantidade de vezes que as palavras de uma frase aparecem.
Exemplo: “estude muito sempre sempre estude” Resultado: estude=2 muito=1 sempre=2
27. Evolua a questão anterior. Permita que o programa salve o resultado em um arquivo de texto CSV. Permita que o programa abra o arquivo de texto CSV e exiba o conteúdo na tela.

Tópicos: recursividade

28. Construa um programa com um método para calcular o número fatorial de um número utilizando recursividade.
29. Calcule o somatório dos números inteiros do intervalo entre dois números. No mesmo programa, implemente esse cálculo em dois métodos:
 - a) Resolva o problema com estrutura de repetição.
 - b) Resolva o problema com recursividade.
 - c) Analise os pontos positivos e negativos de cada versão.

Tópicos: programação genérica

30. Utilizando o recurso Generic de Java, faça um método para exibir um objeto (usando seu método toString), e outro método para exibir os elementos de um vetor.
31. Utilizando o recurso Generic de Java, construa um programa que tenha funcionalidades de CRUD (create, read, update e delete) que se adeque para diferentes classes (ex: String, Pessoa, Produto, etc).
32. Construa um programa CODIFICADOR que receba um arquivo de texto de entrada e codifique ele usando um padrão de troca de letras. Após esse processamento, gere um arquivo codificado.
Construa outro programa, que funcionará como DECODIFICADOR, que seja capaz de ler o arquivo codificado e produzir um arquivo de texto decodificado, que deve ser o mesmo texto original.
Padrão de troca de letras para codificar um texto: Z ⇔ P E ⇔ O N ⇔ L I ⇔ A T ⇔ R

Tópicos: Collection

33. Utilizando a Collection de Java, faça exemplos de uso das classes abaixo e explicita suas diferenças:
 - a) List: ArrayList, Vector, LinkedList
 - b) Set: HashSet, LinkedHashSet, TreeSet
 - c) Queue: PriorityQueue, LinkedList
 - d) Deque: LinkedList
 - e) Map: HashMap, LinkedHashMap, TreeMap