

Lista de exercícios

Exercícios Básicos

Usar if-else

- 1) Definir a idade de uma pessoa e verificar se ela é maior de idade ou não.
- 2) Para cada produto informado (nome, preço e quantidade), escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos pelo número de unidades compradas, segundo a tabela abaixo:
 - a. Até 10 unidades: valor total
 - b. De 11 a 20 unidades: 10% de desconto
 - c. De 21 a 50 unidades: 20% de desconto
 - d. Acima de 50 unidades: 25% de desconto
- 3) Implemente um jogo Jokenpo (Pedra, papel ou tesoura). Será o jogador contra a máquina. O código tem que gerar as posições aleatoriamente e comparar com o que escolhemos.

Usar switch-case

- 4) Implemente o programa da calculadora utilizando uma instrução switch-case para determinar a operação que deve ser executada, conforme o usuário escolheu no menu de opções.

Estrutura de Repetição

Usar for

- 5) Construa a tabuada de um número. (ex: $1 \times 1 = 1$, $1 \times 2 = 2$, ..., $1 \times 10 = 10$);
- 6) Gere 10 números aleatórios entre 0 e 100; mostre todos na tela (em uma única linha); identifique o menor e o maior deles.
- 7) Calcule a série de Fibonacci para um número inteiro não negativo informado pelo usuário. A série de Fibonacci inicia com os números $F_0 = 0$ e $F_1 = 1$, e cada número posterior equivale à soma dos dois números anteriores ($F_n = F_{n-1} + F_{n-2}$). Por exemplo, caso o usuário informe o número 9, o resultado seria 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Usar while/ do - while

- 8) Leia um número até que o usuário digite 10;
- 9) Altere o exercício 7 para o algoritmo contar a série Fibonacci até um determinado número. Por exemplo, caso declarado o número 25, o resultado seria: 0, 1, 1, 2, 3, 5, 8, 13, 21.

Vetores e Matrizes

- 10) Leia uma matriz quadrada e determine a soma da diagonal principal.
- 11) Ordene um vetor de 100 números inteiros gerados aleatoriamente. (Pesquisar sobre Bubble Sort)
- 12) Crie dois vetores de 50 posições com valores inteiros aleatórios, ordene cada vetor individualmente, e combine os dois vetores gerando um novo vetor de 100 posições, ordenando esse novo.

Métodos

- 13) Faça um método que calcule a média de um aluno de acordo com o critério definido neste curso. Além disso, faça um outro método que informe o status do aluno de acordo com a tabela a seguir:
 - Nota acima de 6: "Aprovado"
 - Nota entre 4 e 6: "Verificação Suplementar"
 - Nota abaixo de 4: "Reprovado"
- 14) Leia do usuário o tempo em segundos e escreva em horas, minutos e segundos. Utilize cinco métodos, para a leitura e escrita de dados e para obtenção de horas, minutos e segundos a partir do tempo em segundos.
Ex: 7023 segundos equivalem a 1h57min3seg.
- 15) Crie uma classe Calculadora que faça as operações de soma, subtração, multiplicação, divisão e potenciação. Cada operação deve ser um método e todos os métodos devem ser estáticos. Para fazer a escolha das operações use como base o exercício 4.

Orientação a Objetos

- 16) Faça um programa para controle de empréstimo de livros, com as classes Emprestimo, Livro e Pessoa.
- 17) Faça um programa para representar a árvore genealógica de uma família. Para tal crie uma classe Pessoa que permita indicar além de nome e idade, o pai e a mãe. Tenha em mente que pai e mãe também são do tipo Pessoa.
- 18) Faça uma classe Automóvel, identificando seus atributos e métodos. Herdando tais características, crie três classes: Carro, Moto e Guincho. Herdando Carro, faça uma classe CarroAutomático, na qual, para poder ligar, o freio precisa estar sendo pressionado. Herdando Moto, faça uma classe MotoPartidaPedal, onde, para ligar, o acelerador precisa estar puxado. O guincho pode carregar tanto carro quanto moto. Utilize os 4 pilares da POO.

Exemplo:

Automóvel
- numRodas: int - velMax: int - ligado: boolean
+ ligar()

- 19) Implemente um sistema de agenda telefônica. Faça (ao menos) as interações de adicionar e deletar números.

Prova

20) O projeto SuperMercadoProva simula o sistema de gerenciamento de vendas de um supermercado, com controle de estoque, verificação da existência produtos escolhidos no pedido e demais funcionalidades descritas no diagrama de classes do mesmo. A prova consiste na resolução de três tarefas, são elas:

- Corrigir o bug. (fácil)
- Desenvolver um método onde receba um valor e retorne o troco. (médio)
- Desenvolver um método onde recebe o valor do troco e calcule a menor quantidade de notas. (difícil)

