**TÓPICOS PARA SEREM ABORDADOS:**

1. **Condições Lógicas:**
   * Compreender a estrutura de condicionais (SE, SENÃO SE, SENÃO) e como aplicá-las para tomar decisões baseadas em condições (ex: se um número é positivo, negativo ou zero, ou se um aluno é aprovado, em recuperação ou reprovado).
2. **Estruturas de Repetição:**
   * Conhecer como utilizar loops (PARA, ENQUANTO) para iterar sobre listas ou realizar operações repetitivas, como a ordenação de elementos.
3. **Funções e Recursão:**
   * Entender a definição e uso de funções, incluindo a recursão, como no cálculo do fatorial de um número e na verificação de números primos.
4. **Manipulação de Listas:**
   * Saber como declarar, ler e manipular listas (ou arrays) em pseudocódigo, incluindo operações como inserção e ordenação de elementos.
5. **Ordenação:**
   * Familiaridade com métodos de ordenação, como o método de seleção, e a capacidade de implementar a lógica de ordenação em um algoritmo.
6. **Lógica de Programação:**
   * Compreender conceitos gerais de lógica de programação, como operadores de comparação (maior que, menor que, igual), e a aplicação de lógica condicional em algoritmos.
7. **Cálculo de Médias e Descontos:**
   * Conhecimento básico sobre como calcular médias e aplicar regras para determinar descontos em compras.
8. **Pseudocódigo:**
   * Habilidade para interpretar e escrever pseudocódigos, que é uma forma simplificada de descrever algoritmos sem utilizar sintaxe de linguagem de programação específica.

**CONTEÚDO:**

**Condições Lógicas**

As condições lógicas são essenciais na programação, permitindo que você tome decisões baseadas em critérios específicos. O uso de estruturas condicionais, como SE, SENÃO SE e SENÃO, permite que o programa execute diferentes ações dependendo do resultado de uma condição. Por exemplo, você pode verificar se um número é positivo, negativo ou zero, e executar ações distintas para cada caso.

**Estruturas de Repetição**

As estruturas de repetição, como os loops PARA e ENQUANTO, são utilizadas para executar um bloco de código várias vezes. Elas são úteis para iterar sobre elementos de uma lista, calcular somas, médias ou realizar qualquer tarefa que exija repetição. Pratique a implementação de loops em seus algoritmos para se familiarizar com essa estrutura.

**Funções e Recursão**

As funções são blocos de código que podem ser chamados a partir de outras partes do programa. Elas ajudam a organizar o código e a evitar repetições. A recursão é um conceito em que uma função se chama a si mesma para resolver um problema. Um exemplo clássico é o cálculo do fatorial de um número, que pode ser expresso de maneira recursiva.

**Manipulação de Listas**

As listas (ou arrays) são estruturas de dados que permitem armazenar múltiplos valores. Você deve saber como declarar, ler e manipular listas em pseudocódigo. A ordenação de listas é uma habilidade importante, e métodos como o método de seleção são comuns para esse propósito. Familiarize-se com a lógica de ordenação e pratique a implementação de algoritmos que organizam dados.

**Lógica de Programação**

A lógica de programação envolve o uso de operadores de comparação (maior que, menor que, igual) para criar condições e tomar decisões em um algoritmo. Compreender como esses operadores funcionam é fundamental para o desenvolvimento de algoritmos eficazes. Além disso, a habilidade de calcular médias e aplicar regras de desconto em compras é uma aplicação prática que você encontrará frequentemente.

**Pseudocódigo**

O pseudocódigo é uma forma de descrever algoritmos de maneira simplificada, sem se preocupar com a sintaxe de uma linguagem específica. Praticar a escrita de pseudocódigos o ajudará a estruturar suas ideias antes de implementar um algoritmo em uma linguagem de programação. Tente escrever pseudocódigos para diferentes problemas e refine-os conforme necessário.

**QUESTÃO 1**

**Contexto:**

Você está desenvolvendo um programa que verifica se um número é par ou ímpar. Para isso, usa uma estrutura condicional que avalia o valor informado pelo usuário. Considere o seguinte pseudocódigo:

iniciar

ler número

se (número % 2 == 0)

imprimir "O número é par"

senão

imprimir "O número é ímpar"

fim

**Comando:**

Com base no pseudocódigo acima, qual é a principal função do operador %?

a) Verificar se o número é divisível por 10.

b) Retornar o resultado da divisão inteira.

c) Verificar se o número é positivo ou negativo.

d) Retornar o resto da divisão entre dois números.

e) Multiplicar dois números e retornar o produto.

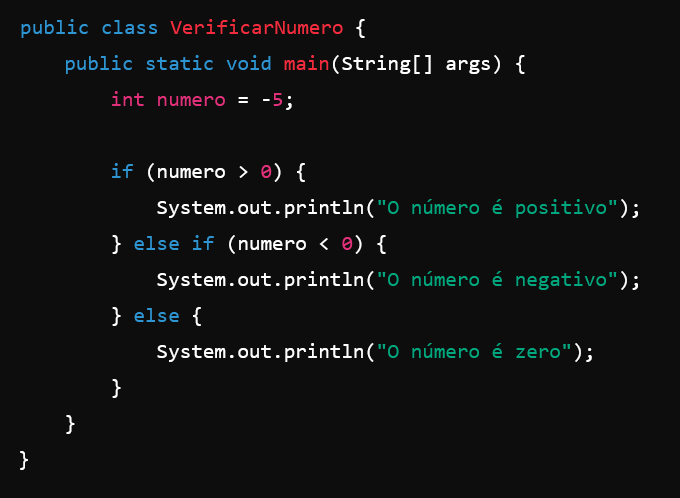
**Resposta correta:** d) Retornar o resto da divisão entre dois números.

Essa questão avalia o entendimento do operador % (módulo), fundamental em lógica de programação para verificar divisibilidade, como no caso de números pares e ímpares.

**QUESTÃO 2**

**Contexto:**

Você está desenvolvendo um programa simples em Java que verifica se um número é positivo, negativo ou zero. Considere o seguinte código Java:



**Comando:**

No código acima, qual parte da estrutura condicional será executada quando a variável numero for igual a 0?

a) System.out.println("O número é positivo");

b) System.out.println("O número é negativo");

c) System.out.println("O número é zero");

d) Nenhuma das opções acima será executada.

e) Todas as opções serão executadas.

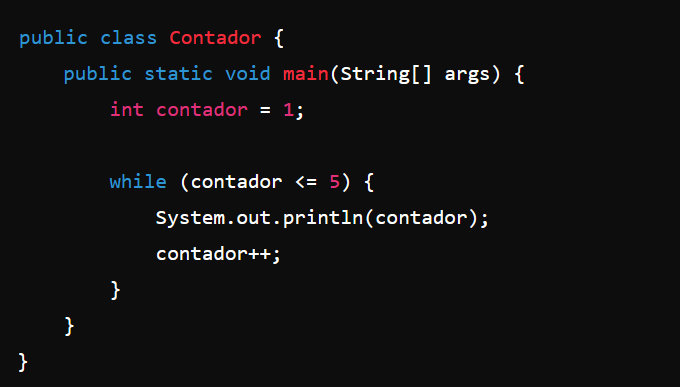
**Resposta correta:** c) System.out.println("O número é zero");

Essa questão testa a compreensão de estruturas condicionais (if, else if, else) e execução de código em Java.

**QUESTÃO 3**

**Contexto:**

Você está desenvolvendo um programa em Java que imprime os números de 1 a 5 usando um laço de repetição while. Considere o seguinte código:



**Comando:**

Qual será a saída do programa?

a) 1 2 3 4 5

b) 1 2 3 4 5 6

c) 0 1 2 3 4

d) O programa apresentará um erro de compilação

e) O laço while nunca será executado

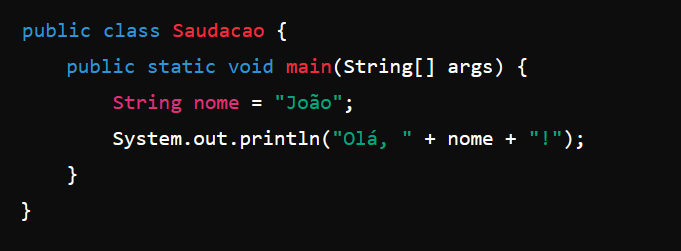
**Resposta correta:** a) 1 2 3 4 5

Essa questão avalia o entendimento básico de laços de repetição (while) e a manipulação de variáveis em Java.

**QUESTÃO 4**

**Contexto:**

Você está desenvolvendo um programa em Java que imprime uma saudação personalizada com base no nome fornecido pelo usuário. Considere o seguinte código:

****

**Comando:**

Qual será a saída do programa?

a) Olá, !

b) Olá, João!

c) Olá, João

d) João Olá!

e) O programa apresentará um erro de compilação

**Resposta correta:** b) Olá, João!

Essa questão avalia o entendimento de concatenação de strings e saída de dados usando System.out.println em Java.

**QUESTÃO 5**

**Contexto:**

Você está desenvolvendo um sistema de gerenciamento de alunos para uma escola. O sistema deve calcular a média final de um aluno com base nas notas de três provas. As regras para aprovação são as seguintes:

1. A média deve ser igual ou superior a 7,0 para aprovação.
2. Se a média estiver entre 5,0 e 6,9, o aluno deve fazer uma prova de recuperação.
3. Se a média for inferior a 5,0, o aluno será reprovado.

**Comando:**

Um aluno obteve as seguintes notas nas três provas: 6,0, 8,0 e 7,0. Qual será a ação que o sistema deve tomar em relação ao aluno?

a) O aluno será aprovado automaticamente.

b) O aluno será reprovado.

c) O aluno deve fazer uma prova de recuperação.

d) O aluno precisa refazer a última prova.

e) O aluno terá uma média de 7,5 e será aprovado.

**Resposta correta:** a) O aluno será aprovado automaticamente.

**Justificativa:** A média do aluno é (6,0 + 8,0 + 7,0) / 3 = 7,0, que é igual a 7,0, portanto ele é aprovado. Essa questão avalia a capacidade de calcular médias e aplicar condições lógicas em um contexto de programação.

**QUESTÃO 6**

**Contexto:**

Você está desenvolvendo um algoritmo em pseudocódigo para calcular o desconto aplicado a uma compra. As regras para o desconto são as seguintes:

1. Se o valor da compra for inferior a R$ 100, não há desconto.
2. Se o valor da compra estiver entre R$ 100 e R$ 300, o desconto é de 10%.
3. Se o valor da compra for superior a R$ 300, o desconto é de 20%.

Considere o seguinte pseudocódigo:

INÍCIO

LER valorCompra

SE valorCompra < 100 ENTÃO

desconto = 0

SENÃO SE valorCompra >= 100 E valorCompra <= 300 ENTÃO

desconto = valorCompra \* 0.10

SENÃO

desconto = valorCompra \* 0.20

FIM\_SE

valorFinal = valorCompra - desconto

ESCREVER valorFinal

FIM

**Comando:**

Se o valor da compra for R$ 250, qual será o valor final a ser pago após aplicar o desconto conforme o algoritmo?

a) R$ 250

b) R$ 225

c) R$ 200

d) R$ 240

e) R$ 275

**Resposta correta:** b) R$ 225

**Justificativa:** O desconto é de 10% sobre R$ 250, que é R$ 25. Portanto, o valor final é R$ 250 - R$ 25 = R$ 225. Essa questão avalia a compreensão de condições lógicas e cálculos de desconto em pseudocódigo.

**QUESTÃO 7**

**Contexto:**

Você está desenvolvendo um algoritmo em pseudocódigo para calcular o fatorial de um número inteiro não negativo. O fatorial de um número nnn (denotado como n!n!n!) é o produto de todos os inteiros de 1 até nnn. As regras são as seguintes:

1. Se n=0n = 0n=0, o fatorial é 1.
2. Para n>0n > 0n>0, o fatorial é calculado como n!=n×(n−1)!n! = n \times (n - 1)!n!=n×(n−1)!.

Considere o seguinte pseudocódigo:

FUNÇÃO fatorial(n)

SE n = 0 ENTÃO

RETORNAR 1

SENÃO

RETORNAR n \* fatorial(n - 1)

FIM\_SE

FIM\_FUNÇÃO

INÍCIO

LER numero

resultado = fatorial(numero)

ESCREVER resultado

FIM

**Comando:**

Se o usuário inserir o número 5, qual será o valor retornado pela função fatorial?

a) 5

b) 10

c) 15

d) 120

e) O algoritmo apresentará um erro

**Resposta correta:** d) 120

**Justificativa:** O fatorial de 5 é calculado como 5! = 5 = 120. Essa questão avalia a compreensão de recursão e a lógica envolvida no cálculo do fatorial em pseudocódigo.

**QUESTÃO 8**

**Contexto:**

Você está desenvolvendo um algoritmo em pseudocódigo que organiza uma lista de nomes em ordem alfabética. O algoritmo deve ler uma quantidade fixa de nomes e, em seguida, ordená-los usando o método de seleção. O algoritmo segue as seguintes etapas:

1. Ler a quantidade de nomes a serem inseridos.
2. Ler cada nome e armazená-los em uma lista.
3. Ordenar a lista usando o método de seleção.
4. Exibir a lista ordenada.

Considere o seguinte pseudocódigo:

FUNÇÃO ordenarNomes(lista)

PARA i DE 0 ATÉ tamanho(lista) - 1 FAÇA

menorIndice = i

PARA j DE i + 1 ATÉ tamanho(lista) - 1 FAÇA

SE lista[j] < lista[menorIndice] ENTÃO

menorIndice = j

FIM\_SE

FIM\_PARA

SE menorIndice != i ENTÃO

troque(lista[i], lista[menorIndice])

FIM\_SE

FIM\_PARA

FIM\_FUNÇÃO

INÍCIO

LER quantidade

DECLARE nomes[quantidade]

PARA k DE 0 ATÉ quantidade - 1 FAÇA

LER nomes[k]

FIM\_PARA

ordenarNomes(nomes)

ESCREVER nomes

FIM

**Comando:**

Se o usuário inserir os seguintes nomes na ordem: "Carlos", "Ana", "Beatriz", "Fernando", qual será a ordem correta após a execução do algoritmo?

a) Carlos, Ana, Beatriz, Fernando

b) Ana, Beatriz, Carlos, Fernando

c) Fernando, Carlos, Ana, Beatriz

d) Beatriz, Ana, Fernando, Carlos

e) Beatriz, Carlos, Ana, Fernando

**Resposta correta:** b) Ana, Beatriz, Carlos, Fernando

**Justificativa:** Após a execução do algoritmo de ordenação, a lista de nomes ficará em ordem alfabética, resultando na sequência: "Ana", "Beatriz", "Carlos", "Fernando". Essa questão avalia a compreensão da lógica de programação relacionada à manipulação e ordenação de listas em pseudocódigo.