**1. O que é atribuição em programação? Como é realizada em Java?**

A atribuição é o processo de armazenar um valor em uma variável.

Em Java, usamos o operador de atribuição = para realizar essa tarefa.

**2. Explique o que é a classe Scanner em Java e qual é sua utilidade.**

A classe Scanner em Java permite a leitura de dados da entrada padrão (normalmente o teclado). Ela é amplamente usada para interagir com o usuário e obter informações.

**3. Quais são os principais tipos de dados em Java utilizados para armazenar**

**números inteiros, números decimais, caracteres e valores lógicos (true ou**

**false)?**

Inteiros: *int*

Números decimais: *double*

Caracteres: *String/char*

Valores lógicos: *boolean*

**4. Como a concatenação é realizada em Java e qual é sua importância na**

**manipulação de strings?**

A concatenação é o ato de combinar strings (cadeias de caracteres) para formar uma única string maior. Isso é comumente usado para criar mensagens de saída ou manipular texto.

**5. O que é uma variável na programação? Qual é a diferença entre uma variável e uma constante?**

Variáveis são espaços de memória usados para armazenar dados temporários. As variáveis podem ter seu valor modificado a qualquer momento, enquanto as variáveis finais e constantes recebem um valor e não podem ser alteradas.

**6. Explique a diferença entre os operadores + e += em relação à atribuição em**

**Java.**

**+:** operador aritmético e concatenar variáveis e Strings.

**+=:** soma ele mesmo mais algum número ou variável.

**7. Por que é importante converter tipos de dados ao realizar operações entre**

**eles? Dê um exemplo.**

Quando você realiza operações entre tipos de dados incompatíveis, podem ocorrer erros ou resultados inesperados, pois o computador não sabe como lidar com esses tipos de dados de maneira automática. A conversão de tipos de dados permite que você ajuste os valores para que eles sejam compatíveis e possam ser processados adequadamente.

|  |
| --- |
| *public class* ***ConversaoTipoDadosExemplo*** *{*  *public static void main(String[] args) {*  *int numeroInteiro = 10;*  *String numeroString = "5";*  *// Converter a String em um int*  *int numeroConvertido =* ***Integer.parseInt****(numeroString);*  *// Realizar a operação de adição*  *int resultado = numeroInteiro + numeroConvertido;*  *System.out.println("O resultado da soma é: " + resultado);*  *}*  *}* |

**8. Qual é a diferença entre os operadores de comparação == e = em Java?**

**==:** é usado para comparar igualdade entre valores ou referências.

**=:** é usado para atribuir um valor a uma variável.

**9. O que é um fluxo de entrada e como o System.in é usado em conjunto com o Scanner para coletar dados de entrada do usuário?**

Em Java, um fluxo de entrada (input stream) refere-se a uma sequência de dados que está fluindo em direção ao programa. O System.in é um objeto da classe InputStream que representa o fluxo de entrada padrão do sistema, normalmente associado à entrada do teclado.

Para coletar dados de entrada do usuário em Java, você pode usar a classe Scanner em conjunto com o System.in. O Scanner é uma classe que fornece métodos convenientes para analisar e ler dados de diferentes tipos a partir de um fluxo de entrada, como o teclado (representado por System.in).

|  |
| --- |
| import java.util.Scanner;  public class **EntradaDeDados** {  public static void main(String[] args) {  *// Crie um objeto Scanner associado ao System.in*  **Scanner scanner = new Scanner(System.in);**  *// Solicite ao usuário que insira um número*  System.out.print("Digite um número inteiro: ");  *// Use o Scanner para ler o número digitado pelo usuário*  int numero = scanner.**nextInt();**  *// Exiba o número digitado pelo usuário*  System.out.println("Você digitou: " + numero);  *// Não se esqueça de fechar o Scanner quando não for mais necessário*  scanner.close();  }  } |

**10. Explique como funciona o processo de arredondamento de números em Java e quando isso pode ser necessário em uma aplicação.**

Com método estático round() na classe Math:

Math.round()