

TEMA 2 - JAVA

Guia de Estudos: Introdução ao Java com NetBeans

1. Introdução ao Java e ao NetBeans

O **Java** é uma das linguagens de programação mais utilizadas no mundo devido à sua **portabilidade, segurança e desempenho**. É usada para desenvolver aplicativos desktop, sistemas web, jogos e até aplicações para dispositivos móveis.

O **NetBeans** é um ambiente de desenvolvimento integrado (IDE) gratuito que facilita a criação, depuração e execução de programas em Java.

Configuração Inicial no NetBeans

1. Baixe e instale o **NetBeans** e o **JDK (Java Development Kit)**.
2. Abra o NetBeans e crie um **novo projeto Java**.
3. No projeto, crie uma **nova classe** e nomeie-a, por exemplo, MeuPrograma.java.
4. Dentro da classe, escreva o código dentro do método main, que é o ponto de entrada do programa.

2. Estrutura Básica de um Programa Java

Todo programa em Java deve conter uma classe e um método main, como mostrado abaixo:

```
public class MeuPrograma {  
    public static void main(String[] args) {  
        System.out.println("Olá, mundo!");  
    }  
}
```

Explicação:

- `public class MeuPrograma` → Declaração da classe principal. O nome do arquivo deve ser o mesmo da classe.
- `public static void main(String[] args)` → Método principal que será executado primeiro.
- `System.out.println("Olá, mundo!");` → Comando para exibir texto na tela.

Saída no prompt:

Olá, mundo!

3. Tipos de Variáveis em Java

Principais tipos primitivos em Java:

Tipo	Descrição	Exemplo
int	Número inteiro	int idade = 25;
double	Número decimal	double altura = 1.75;
char	Caractere único	char inicial = 'A';
boolean	Verdadeiro ou falso	boolean temCarteira = true;
String	Texto (cadeia de caracteres)	String nome = "Carlos";

◆ Exemplo de declaração e exibição de variáveis:

```
public class VariaveisExemplo {  
    public static void main(String[] args) {  
        int idade = 30;  
        double salario = 2500.75;  
        char inicial = 'J';  
        boolean empregado = true;  
        String nome = "João";  
  
        System.out.println("Nome: " + nome);  
        System.out.println("Idade: " + idade + " anos");  
        System.out.println("Salário: R$ " + salario);  
        System.out.println("Inicial do nome: " + inicial);  
        System.out.println("Empregado: " + empregado);  
    }  
}
```

📌 Saída no prompt:

Nome: João

Idade: 30 anos

Salário: R\$ 2500.75

Inicial do nome: J

Empregado: true

4. Operações Matemáticas em Java

Operações básicas:

- Soma: +
- Subtração: -
- Multiplicação: *
- Divisão: /
- Resto da divisão (módulo): %

◆ Exemplo prático:

```
public class OperacoesMatematicas {  
    public static void main(String[] args) {  
        int numero1 = 10;  
        int numero2 = 5;  
  
        int soma = numero1 + numero2;  
        int multiplicacao = numero1 * numero2;  
  
        System.out.println("Soma: " + soma);  
        System.out.println("Multiplicação: " + multiplicacao);  
    }  
}
```

📌 Saída no prompt:

Soma: 15

Multiplicação: 50

5. Concatenação e Exibição com System.out.println

Concatenação

Concatenar significa juntar valores em uma única saída. Isso pode ser feito usando +.

◆ Exemplo:

```
public class ConcatenacaoExemplo {  
    public static void main(String[] args) {
```

```
String produto = "Notebook";

double preco = 2599.99;

int quantidade = 2;


System.out.println("Produto: " + produto + ", Preço: R$ " + preco);

System.out.println("Quantidade: " + quantidade + " unidades");


String mensagem = String.format("Total da compra (%d unidades): R$ %.2f",
quantidade, preco * quantidade);

System.out.println(mensagem);

}

}
```



Saída no prompt:

Produto: Notebook, Preço: R\$ 2599.99

Quantidade: 2 unidades

Total da compra (2 unidades): R\$ 5199.98

6. Boas Práticas na Nomeação de Variáveis e Classes

1. **Nomes descritivos** → Use nomes que representem claramente o valor armazenado.
 2. `int idadeCliente = 30;`
 3. **CamelCase** → Em Java, nomes de variáveis começam com letra minúscula e as palavras seguintes com maiúscula.
 4. `String nomeCompleto = "Ana Souza";`
 5. **Nomes de classes iniciam com maiúscula**
 6. `public class RelatorioClientes { }`
 7. **Evite abreviações sem sentido**
 `int i = 10;`  `int quantidadeProdutos = 10;`
 8. **Use constantes para valores fixos**
 9. `final double PI = 3.1416;`
-

7. Comentários em Java

Os comentários ajudam a documentar o código e são ignorados pelo compilador.

◆ Tipos de comentários:

// Comentário de uma linha

/*

Comentário de várias linhas

explicando o código.

*/

/**

* Comentário de documentação.

* Usado para gerar documentação automática do código.

*/

◆ Exemplo:

```
public class Comentarios {  
    public static void main(String[] args) {  
        // Exibe uma mensagem na tela  
        System.out.println("Olá, mundo!");  
    }  
}
```

Explicação sobre float e double em Java

Em Java, tanto float quanto double são tipos usados para armazenar números decimais, mas têm diferenças importantes:

Tipo	Tamanho	Precisão	Exemplo de uso
float	32 bits	Aproximadamente 6 a 7 dígitos decimais	float preco = 1.5f;
double	64 bits	Aproximadamente 15 a 16 dígitos decimais	double preco = 1.5;

Por que double é a resposta correta?

No código fornecido, os preços dos produtos foram declarados como **double**, que é o tipo **padrão** para valores decimais em Java.

Se quiséssemos usar float, precisaríamos adicionar o sufixo **f** ao valor, pois Java assume que valores decimais são double por padrão.

Exemplo:

```
float preco = 1.5f; // Correto
```

```
float outroPreco = 2.75; // Erro! Precisa do 'f' no final
```

Portanto, o uso de float causaria erro se não tivesse o sufixo f.