**Aula – Fundamentos Java para Desenvolvimento Web com STS Boot**

1. **Introdução ao Java**

**Java é:**

* Uma linguagem **orientada a objetos**.
* **Portável** (funciona em qualquer SO com **JVM**).
* Muito utilizada para **aplicações web empresariais** (principalmente com **Spring Framework**).

**Ciclo de execução:**

1. Você escreve um código **.java**.
2. O compilador converte em **bytecode** (**.class**).
3. A **JVM** interpreta e executa o **bytecode**.
4. **Estrutura básica de um programa Java**

public class OlaMundo {

**public static void main(String[] args) {**

System.out.**println**("Olá, mundo!");

}

}

**Explicação:**

* **public class** **OlaMundo** → definição de uma classe chamada OlaMundo.
* **public static void main(String[] args)** → ponto de entrada da aplicação.
* **System.out.println** → imprime no console. Atalho **sysout** + CTRL + barra de espaço.

1. **Entrada de Dados**

import **java.util.Scanner**;

**Scanner** ler = **new** **Scanner**(**System.in**);

int num = ler.**nextInt**();

String nome = ler.**next**();

1. **Saída de Dados**

int num = 5;

System.out.**print**("Número: " + num);

System.out.**println**(**String.format**("Número %d %d %s é positivo", num, 5, "juju"));

System.out.**printf**("Número %d é positivo\n", num);

System.out.**println**(**String.format**("Elemento %d não encontrado", num));

1. **Tipos de dados e variáveis**

| **Tipo** | **Exemplo** | **Descrição** |
| --- | --- | --- |
| int | int idade = 30; | Número inteiro |
| double | double preco1 = 19.99; | Número decimal |
| float | float preco2 = 21.99F; | Número decimal |
| char | char letra = 'A'; | Caractere único |
| String | String nome = "Heleno"; | Texto |
| boolean | boolean ativo = true; | Verdadeiro ou falso |

1. **Operadores básicos**

int a = 10, b = 5;

int soma = a + b; // 15

int subtracao = a - b; // 5

int multiplicacao = a \* b; // 50

int divisao = a / b; // 2

boolean comparacao = a > b; // true

1. **Estruturas de controle**

**If / Else**

int idade = 20;

**if** (idade >= 18) {

System.out.**println**("Maior de idade");

} **else** {

System.out.**println**("Menor de idade");

}

**Laço For**

**for** (int i = 1; i <= 5; i++) {

System.out.**println**("Número: " + i);

}

**Laço For in**

int[] vet = **new** int[] {5, 6, 55};

**for** (int **elemento** : vet) {

System.out.**println**("Número: " + **elemento**);

}

**Laço While**

int i = 1;

**while** ( i <= 5 ) {

System.out.**println**("Número: " + i);

i++;

}

1. **Fundamentos de Orientação a Objetos**

**Classe e Objeto**

public class **Pessoa** {

**String** nome;

**int** idade;

void **apresentar**() {

System.out.**println**("Olá, meu nome é " + nome);

}

}

**Usando a classe:**

public **class Programa** {

**public static void main(String[] args) {**

Pessoa p1 = **new** Pessoa();

p1.nome = "Ana";

p1.idade = 25;

p1.apresentar();

}

}

1. **Preparando para o Spring Boot no STS**

Antes de criar a aplicação web, você precisa:

1. **Instalar o Java JDK 21** (**ou versão recomendada**).
2. **Não instalar STS (Spring Tool Suite), fazer download do .ZIP**.
3. Conhecer os conceitos básicos que vimos acima, pois:
   * No Spring, classes se tornam **componentes**.
   * Métodos podem ser **endpoints** web.
   * Objetos podem representar **entidades** do banco.
4. **Criando um projeto no STS**
5. **File → New → Spring Starter Project**.
6. Dê um nome ao projeto: **meu-primeiro-app**.
7. Selecione **Maven** e **Java 21**.
8. No campo **Dependencies**, escolha:
   * **Spring Web e Thymeleaf**
9. Finalize e aguarde o **Maven** baixar as dependências.

**Exemplo de Controller:**

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

**@Controller**

public class **HomeController** {

**@GetMapping**("/")

public String **index()** {

**return** "index"; // Vai buscar index.html no templates

} }