



**UNIVERSIDADE
FEDERAL DA
FRONTEIRA SUL**

Programação I:

Mercado de Trabalho e Ambientes de desenvolvimento

Autor: Samuel Feitosa
Contribuições: Jefferson Caramori



Orientação a objetos

A Amplitude da Área de Desenvolvimento de Software:

- A área de desenvolvimento de software é vasta e abrange uma variedade de disciplinas e atividades.

Diversidade de Linguagens de Programação e Paradigmas:

- Há uma diversidade de linguagens de programação disponíveis, cada uma seguindo diferentes paradigmas e adequadas para diversos propósitos. (Java, Ruby, C++, Python)

Popularidade da Programação Orientada a Objetos (POO):

- A Programação Orientada a Objetos (POO) é amplamente adotada pelos programadores devido à sua eficiência e capacidade de modelagem do mundo real.

Natureza Especial da Programação Orientada a Objetos:

- A POO é uma abordagem especial de programação que se aproxima da forma como concebemos e expressamos conceitos na vida real, permitindo uma modelagem mais intuitiva e organizada.



Discussão: Por quê é tão importante?

Vamos discutir esta pergunta em duas etapas:

- a) Linguagens de Programação
- b) Valorização do Mercado Profissional.



1º ponto: Linguagens de Programação

- Vamos acessar o ranking da TIOBE, que lista as linguagens de programação mais utilizadas.
 - <https://www.tiobe.com/tiobe-index/>
- Vamos procurar na Wikipedia as características das 5 melhores colocadas no ranking.
 - C, Java, Python, C++, C#.
- Apenas a linguagem C não é orientada a objetos.



2º ponto: Valorização Profissional

- Saber POO não é capricho, é uma exigência do mercado de trabalho.
- Existem vários tipos de desenvolvedores, e saber o que está fazendo indica:
 - Conseguir abstrair os conceitos do mundo real.
 - Traduzi-los problemas do mundo real para o software.
 - Utilizar os conceitos de POO corretamente.
- Os conceitos de POO são os mesmos para qualquer linguagem de programação.



Pilares da Programação Orientada a Objetos

Chamaremos de pilares os principais conceitos de POO que estudaremos neste semestre:

- **Abstração:** Processo de simplificação e representação de características essenciais de um objeto, ocultando detalhes irrelevantes.
- **Encapsulamento:** Ocultação dos detalhes de implementação de um objeto, permitindo o acesso controlado apenas aos métodos e atributos relevantes.
- **Herança:** Mecanismo que permite a criação de novas classes baseadas em classes existentes, compartilhando características e comportamentos.
- **Polimorfismo:** Capacidade de um objeto de ser tratado como um objeto de seu tipo ou de um tipo mais amplo, possibilitando o uso de métodos de forma genérica.

The background features a series of overlapping, curved green shapes on the left side, creating a sense of depth and movement. The colors range from a bright lime green to a darker forest green. The rest of the background is a plain, light gray.

Ambientes de desenvolvimento, instalação e configuração



Ambientes de Desenvolvimento

- Convencionamos chamar de IDE.
 - sigla para Integrated Development Environment ou Ambiente de Desenvolvimento Integrado em inglês, é um software que reúne uma variedade de características e ferramentas para auxiliar no desenvolvimento de software, visando aprimorar e agilizar o processo de programação, permite desenvolver aplicações mais rapidamente, analisar código, erros de digitação e autocompletar.
- Existem muitas IDEs para Java:
 - Netbeans, Eclipse, IntelliJ, VSCode, etc.
 - Nessa disciplina vamos usar o VSCode e o Netbeans.
- VSCode: muito utilizado hoje em dia para várias linguagens.
- Netbeans: possui um bom builder para programação de interfaces gráficas.



Ferramentas Necessárias

- Para desenvolver aplicações em Java, precisamos basicamente de duas coisas:
 - JDK: Java Development Kit.
 - IDE: VSCode/Netbeans.
- Para rodar qualquer aplicação em Java, é preciso ter instalado o Java Runtime Environment (JRE).
 - **JRE** (Java Runtime Environment): é um conjunto de programas e bibliotecas que permite a execução de aplicativos Java em um computador. Ele inclui a JVM (Java Virtual Machine), bibliotecas Java padrão e outros componentes necessários para executar aplicativos Java. Não contém ferramentas de desenvolvimento, como compiladores e depuradores.
 - **JDK** (Java Development Kit): é um conjunto completo de ferramentas para desenvolver aplicativos Java. Inclui o JRE, bem como compiladores Java (como javac), ferramentas de depuração (como jdb), bibliotecas de classes do Java, documentação e outros utilitários de desenvolvimento.



Instalação do JDK

- No Linux/Ubuntu é comum a utilização da OpenJDK.
 - A instalação desta ferramenta pode ser feita via gerenciador de pacotes.
- Caso desejem, é possível instalar a JDK “oficial”, disponibilizada pela Oracle.
 - <https://www.oracle.com/java/technologies/javase-downloads.html>
 - Escolha a opção JDK Download e selecionar o Sistema Operacional utilizado.
- Após instalar, vá até o terminal e execute o comando abaixo para verificar a instalação (comando que exibe a versão do compilador Java (javac)):

java -version

```
openjdk 17.0.8.1 2023-08-24
```

```
OpenJDK Runtime Environment Temurin-17.0.8.1+1 (build 17.0.8.1+1)
```

```
OpenJDK 64-Bit Server VM Temurin-17.0.8.1+1 (build 17.0.8.1+1, mixed mode, sharing)
```



Instalação do VSCode

- É um Ambiente de Desenvolvimento Integrado (IDE), desenvolvido pela Microsoft e muito usado para diferentes linguagens de programação.
 - <https://code.visualstudio.com/>
 - Fazer download da última versão.
 - Escolher a versão do instalador para o Sistema Operacional desejado.
 - Seguir os procedimentos de instalação.
- Após instalar, abra o VSCode e instale as extensões Java: disponível no menu: Run > Install Additional Debuggers., pesquise por “Java Language Support” e “Extension Pack for Java,” clique em instalar.

Leitura complementar: [Desenvolvendo aplicações Java com o VS Code](#)



Instalação do Netbeans

Desde 2018, o NetBeans passou a ser desenvolvido pela Apache Foundation.

- <https://netbeans.apache.org/download/>
- Fazer download da versão: Apache NetBeans 21
- Escolher a versão do instalador para o Sistema Operacional desejado.
- Seguir os procedimentos de instalação.



Testando as Ferramentas

“Hello World”



Sintaxe Java

- Um programa em Java pode ser composto de um ou mais arquivos-fonte, denominados unidades de compilação. Podem conter:
 - Uma declaração de pacote (package).
 - Uma ou mais diretivas de importação (import).
 - Uma ou mais declarações de classes (class).
 - Uma ou mais declarações de interfaces (interface).
- Todo o programa precisa ter o método main, que define seu ponto de inicialização.



Sintaxe Java

- Obrigaç o de definir classes dentro de arquivos com o mesmo nome: Em Java, uma classe p blica deve ser definida em um arquivo com o mesmo nome da classe seguido da extens o ".java".
- Sistema de tipos est tico: Java   uma linguagem fortemente tipada, o que significa que o tipo de cada vari vel deve ser declarado explicitamente. Al m disso, Java   uma linguagem de tipos est tica, o que significa que os tipos s o verificados em tempo de compila  o.
- Utiliza  o de ponto e v rgula para terminar instru  es: cada instru  o deve ser terminada com um ponto e v rgula (;).
- Sintaxe de m todo: M todos em Java seguem a sintaxe `tipoRetorno nomeDoMetodo(par metros) { // corpo do m todo }`.
- Tratamento de exce  es com try-catch-finally: Java utiliza blocos try, catch e finally para lidar com exce  es.
- Palavras-chave reservadas: Java possui um conjunto espec fico de palavras-chave reservadas que n o podem ser usadas como nomes de vari veis, m todos ou classes (Ex: public, class, static, void, main).



Exemplo de Hello World!

Hello.java

```
/* Este é um arquivo exemplo de Hello World! */
public class Hello { // Declaração da Classe Hello
    public static void main(String[] args) { // Declaração do método principal/Main
        // Instrução em Java que imprime a string "Hello world!" no console
        System.out.println("Hello world!");
    }
}
```

- Uso de maiúsculas e minúsculas.
 - nome ≠ NOME, public ≠ Public.