

Programação I

Mercado de Trabalho e Instalação de Ferramentas

Samuel da Silva Feitosa

Aula 2

Mercado de Trabalho

Orientação a Objetos

- A área de desenvolvimento de software é bastante ampla.
 - Existem várias linguagens de programação em diferentes paradigmas.
- Programação Orientada a Objetos é uma das mais usadas pelos programadores.
- É uma maneira especial de programar, muito mais próxima da forma como expressamos todas as coisas na vida real.

Pilares da Orientação a Objetos

- Chamaremos de **pilares** os principais conceitos de POO que estudaremos neste semestre:
 - Abstração
 - Encapsulamento
 - Herança
 - Polimorfismo
- Esta disciplina possui uma grande carga conceitual.

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 dois tipos de desenvolvedores:
 - Aqueles que fazem.
 - Aqueles que fazem e sabem o que estão fazendo.
- Saber o que está fazendo indica:
 - Conseguir abstrair os conceitos do mundo real.
 - Traduzi-los para o software.
 - Utilizar os conceitos de POO corretamente.
- Os **conceitos de POO** são os mesmos para qualquer linguagem de programação.

Ambientes de Desenvolvimento

Instalação e Configuração

Ambientes de Desenvolvimento

- Convencionamos chamar de IDE.
 - PT: Ambiente de Desenvolvimento Integrado.
 - EN: Integrated Development Environment.
- É um software que auxilia no desenvolvimento das aplicações.
 - Permite desenvolver aplicações mais rapidamente.
 - Analisa código, erros de digitação, 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:** Vamos adotar o Netbeans.
- Para **rodar** qualquer aplicação em Java, é preciso ter instalado o Java Runtime Environment (JRE).
 - Este já vem com o JDK (programadores).
 - Porém, é necessário para usuários comuns.

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>
- Fazer download da versão **Java SE 17 (LTS)**.
 - **LTS** significa Long-Term Support.
 - Escolher a opção JDK Download e selecionar o Sistema Operacional utilizado.

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.

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 13 (NB 13.0)
 - 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.

Exemplo Hello World!

```
/*
    Este é o famoso programa Hello World!
    Leia com atenção os comentários indicativos.
*/
package hello; // declaração do pacote

public class Hello { // declaração da classe Hello
    // declaração do método main, que dá início ao programa
    public static void main(String[] args) {
        // impressão da mensagem na tela
        System.out.println("Hello World!");
    }
}
```

- Uso de maiúsculas e minúsculas.
 - nome ≠ NOME, public ≠ Public.
- Comentários em linha ou múltiplas linhas.