

Introdução ao Ecossistema e Documentação Java

O que é Java?

“Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela *Sun Microsystems* em 1995.” Oracle, 2021. Disponível em: <https://www.java.com/pt-BR/download/help/whatis_java.html>

Características:

- Compilada
- Interpretada
- Fortemente tipada
- Linguagem de alto nível
- Executada em uma máquina virtual JVM (*Java Virtual Machine*)

O que é JVM?

“JVM (*Java Virtual Machine*) é uma máquina virtual responsável pela tradução dos *ByteCodes* oriundos do compilador *Javac* (*Java Compiler*) em código de máquina de cada sistema operacional”.

Características:

- Execução de pilhas
- Gerenciamento de memória
- Gerenciamento de *threads*
- Otimização de código (Compilação *JIT* - *Just In Time*)
- *Garbage Collector* (GC)

Quais diferenças entre JRE e JDK?

- **JRE (Java Runtime Environment)** - responsável por executar os programas em Java.
- **JDK (Java Development Kit)** - utilitários que permite o desenvolvimento de programas em Java. Já possui a JVM para executar os programas.

Quais os tipos de plataforma Java?

- **Java SE (Java Standard Edition)** - contém as especificações do Java e pode ser implementado por diversas empresas como a *Oracle*, *OpenJDK*, *Azul Zulu*, etc.
- **Java EE (Java Enterprise Edition)** - contém todas as especificações do *Java SE* e um número de programas úteis para que executem em servidores. Em 2019 foi renomeado para *Jakarta EE*.
- **Java ME (Java Micro Edition)** - contém especificações para desenvolvimento de programas para dispositivos pequenos como celulares, *PDA's*.

Palavras Reservadas

| | | | | |
|----------|----------|------------|-----------|--------------|
| abstract | continue | goto | package | synchronized |
| assert | default | if | private | this |
| boolean | do | implements | protected | throw |
| break | double | import | public | throws |
| byte | else | instanceof | return | transient |
| case | enum | int | short | try |
| catch | extends | interface | static | void |
| char | final | long | strictfp | volatile |
| class | finally | native | super | |
| const | float | new | switch | |

Estrutura Básica de uma Classe

```
public class Order{
    //atributos e métodos
}
```

Estrutura com atributos e métodos

```
public class Order{  
    //atributos  
    private final String code;  
    private final BigDecimal totalValue;  
    //métodos  
    public BigDecimal calculaFee(){  
    }  
}
```

Estrutura Completa de uma Classe

```
package com.dio.base;  
  
import java.math.BigDecimal;  
  
public class Order{  
  
    private final String code;  
    private final BigDecimal totalValue;  
  
    public Order(String code, BigDecimal totalValue){  
        this.code = code;  
        this.totalValue = totalValue;  
    }  
  
    public BigDecimal calculaFee(){  
        return this.totalValue.multiply(new BigDecimal("0.99"));  
    }  
}
```

Modificadores de Acesso

Tipos de modificadores:

- **Public** - qualquer classe de qualquer pacote poderá acessar o atributo ou método.
- **Protected** - qualquer classe definida no mesmo pacote ou subclasse.
- **Sem modificador** - apenas classes definidas no mesmo pacote.
- **Private** - apenas a própria classe.

O que são métodos?

Métodos são funções que definem o comportamento de uma classe.

Tipos de métodos:

- **Métodos construtores** - definem como uma classe será instanciada “construída”;
- **Métodos comum** - definem comportamentos que podem ou não estar atribuídas às regras de negócio. Ex: calcular taxas de um pedido etc.

O que é Javadoc?

Javadoc é uma ferramenta para documentação no formato HTML que se baseia nos comentários do código-fonte.

Os comentários precisam conter *tags* para que a documentação fique legível.

Tags Javadoc

- @@see - associa a outras classes ou métodos;
- @ since - descreve desde quando uma classe ou método foi adicionado.
- @ throws - descreve os tipos de exceções que podem ser lançadas por um método.
- @@ version - descreve a versão da classe ou método.

Todo este conteúdo foi produzido pelo instrutor Daniel Hatanaka da Digital Innovation One.

Disponível em: <https://github.com/hatanakadaniel/dio-introduction-java-ecosystem>