

LINGUAGEM DE PROGRAMAÇÃO ORIENTADA À OBJETOS – LPOO

Aula 02 – Instalação e Configuração do Java

Profa. Thais Rocha - thais.rocha@docente.unip.br



TÓPICOS DA AULA

- Novidades do Java e o Futuro da Linguagem Java
- Configuração e instalação do Java

PERSPECTIVA DE FUTURO DA LINGUAGEM JAVA

- Mercado ainda com bastante oportunidades, devido a grande quantidade de sistemas já desenvolvidos em Java
- Integração de sistemas em Java com outros projetos em outras linguagens
- Java é recomendado para sistemas grandes e complexos (sistemas bancários, logísticos, ERP)
- Projetos com complexidade média/baixa (cadastros/relatórios, loja virtual com integração com NF) é mais recomendado outras linguagens

MITOS E VERDADE SOBRE O JAVA

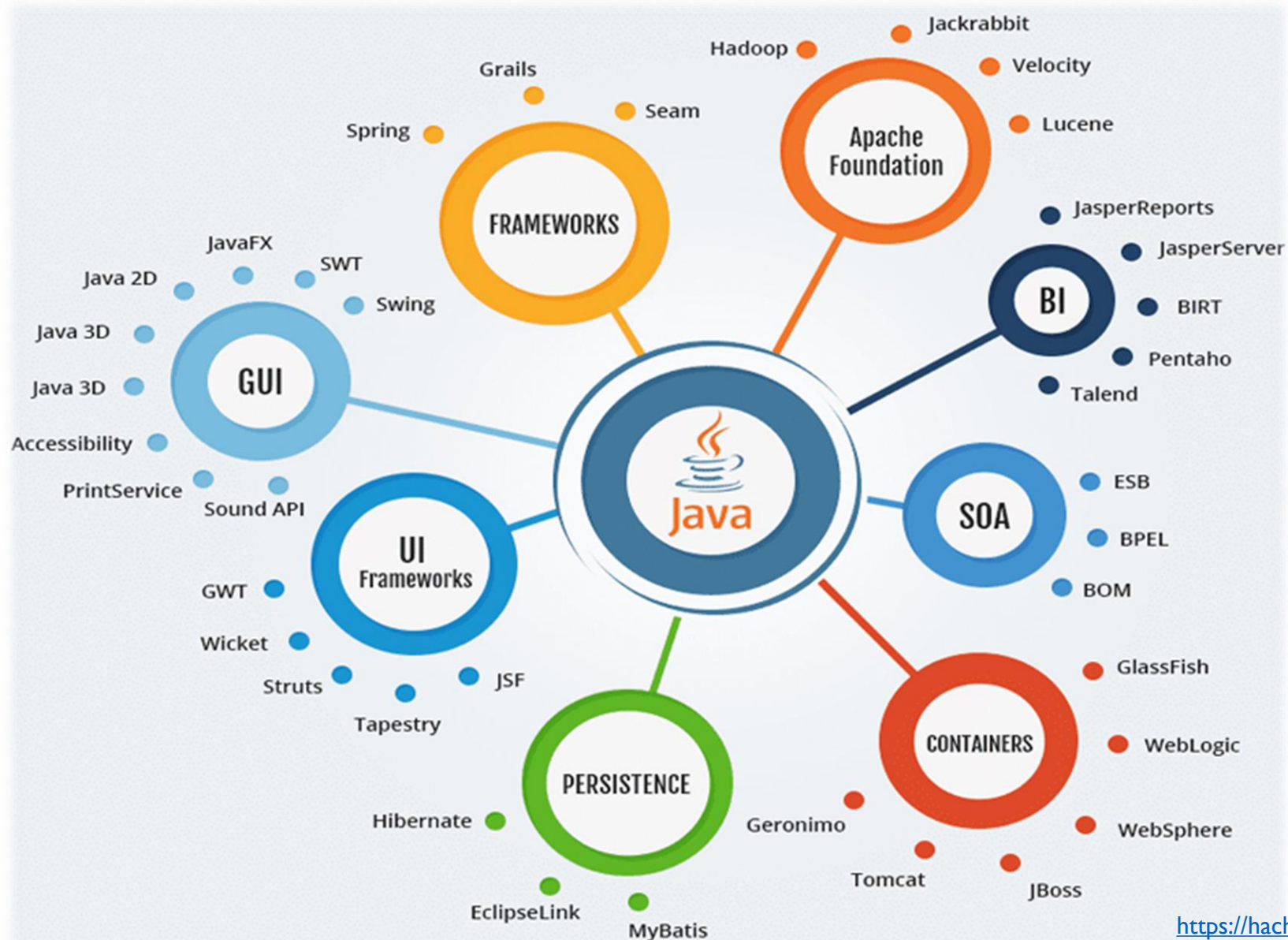
- Os sistemas desenvolvidos em Java são lentos.
- Java é pesado, usa muita memória e processamento.
- Java é difícil de aprender.
- Java vai morrer, igual a linguagem COBOL.
- Em Java é preciso escrever muito código pra pouca coisa.



○ QUE APRENDER EM JAVA ?

- Lógica de programação (IMPORTANTE)
- Orientação a objetos -> Classes, Herança e Interface
- Java Core -> String, Dates, Collections (List, Set, Map), Estrutura de repetição e controle (for, while, if / else)
- Frameworks -> Spring e JSF (Primefaces)

ECOSISTEMA DO JAVA



CONCEITOS IMPORTANTES

Qual a diferença entre JDK, JRE e JVM ?

Uma grande confusão é gerada sobre quem está começando a aprender sobre o mundo Java é **a diferença entre JDK, JRE e JVM.**

Java é muito conhecido por trazer o conceito de **multi-plataforma**. Na verdade esse é o motivo do grande sucesso do Java à mais de vinte anos! O famoso **WORA** (*Write once, run anywhere.*), "Escreva uma vez, execute em qualquer lugar".

CONCEITOS IMPORTANTES (cont.)

O fluxo é basicamente o seguinte:

- Você escreve o seu código-fonte (extensão .java).
- Você utiliza o **JDK** para compilar o seu código-fonte e gerar o arquivo *bytecode* (arquivo com a extensão .class).
- Para executar o seu programa, a **JVM** lê o seu arquivo compilado (.class) e as bibliotecas padrões do Java que estão no **JRE**.

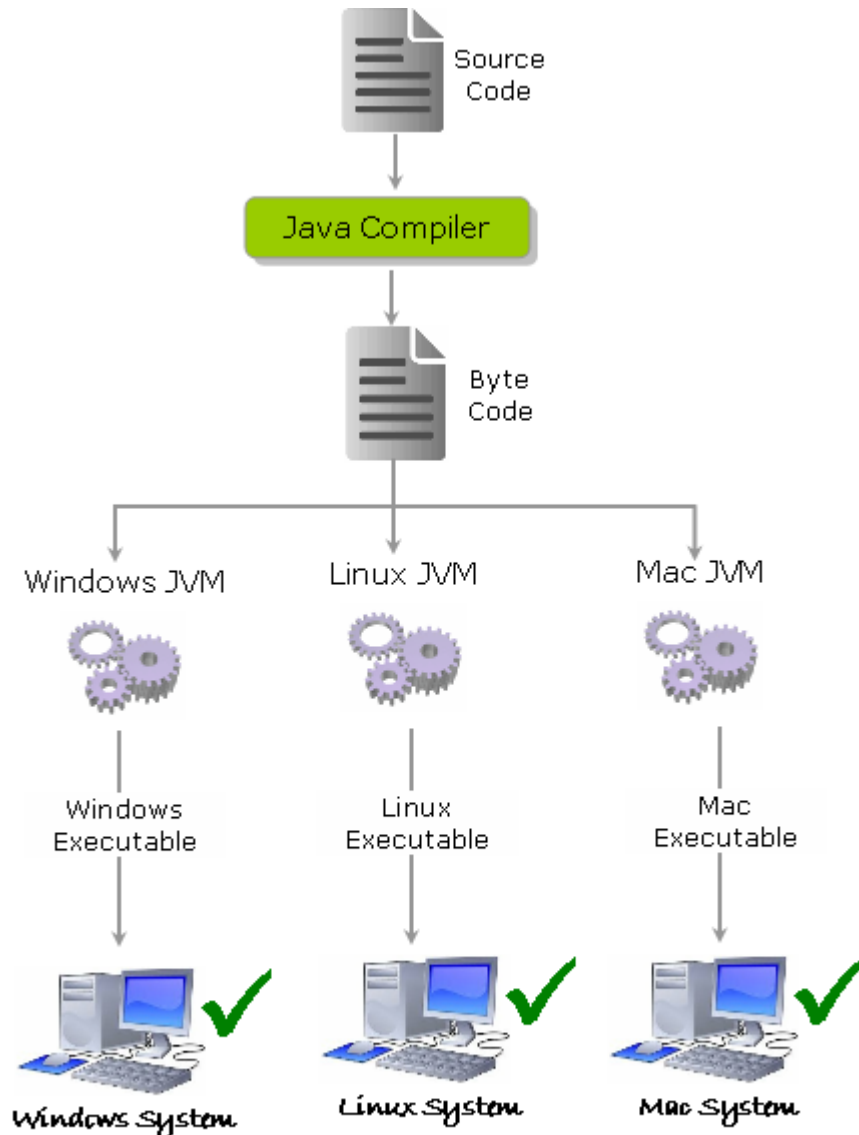
Pronto, seu programa está rodando e todo mundo está feliz! :)

CONCEITOS IMPORTANTES (cont.)

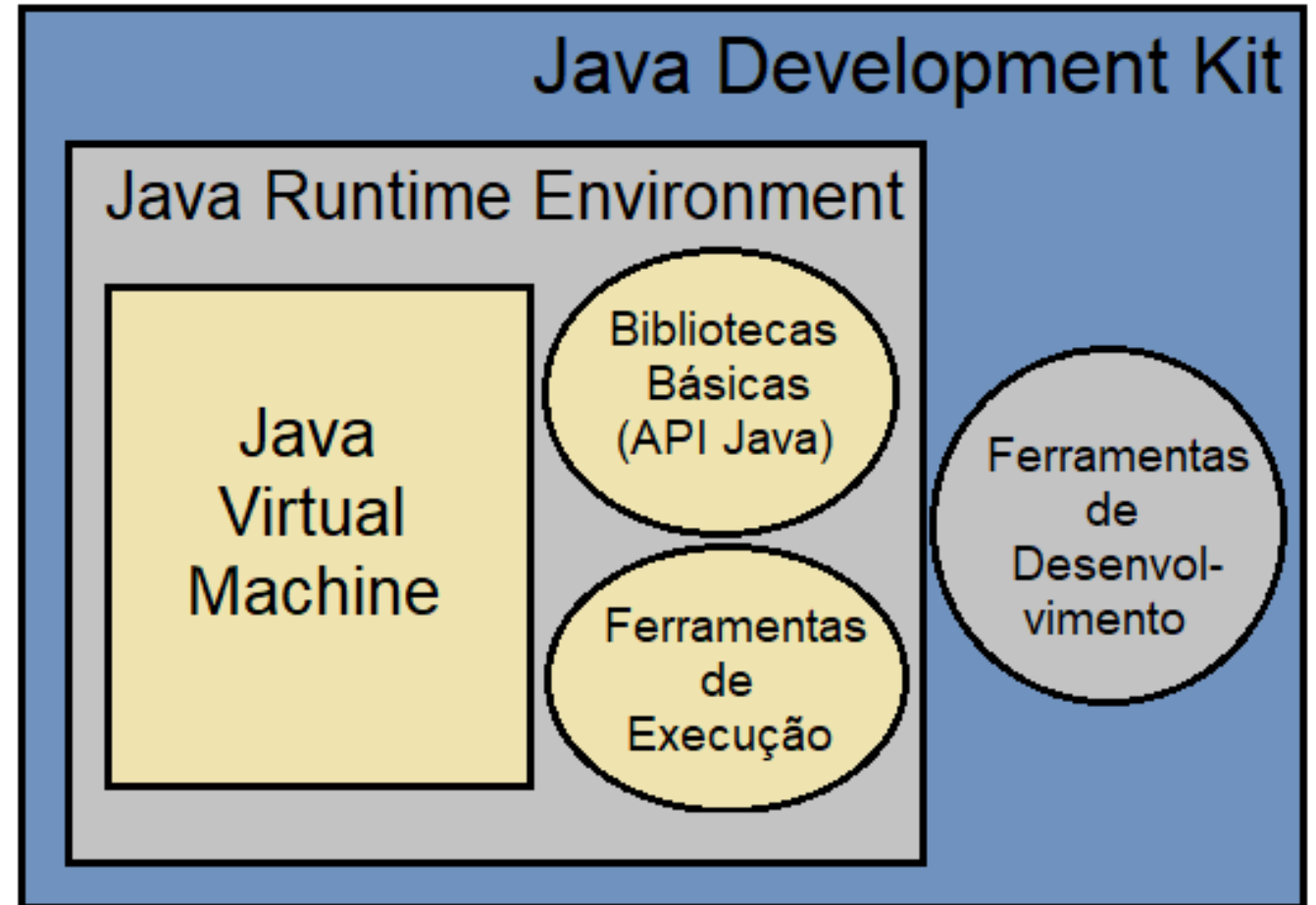
Então, de maneira resumida, já deu pra perceber pra quê serve cada um:

- **JDK (Java Development Kit)** - é o Kit de Desenvolvimento Java responsável por compilar código-fonte (.java) em bytecode (.class)
- **JVM (Java Virtual Machine)** - é a Máquina Virtual do Java responsável por executar o bytecode (.class)
- **JRE (Java Runtime Environment)** - Ambiente de Execução do Java que fornece as bibliotecas padrões do Java para o JDK compilar o seu código e para a JVM executar o seu programa.

VISÃO GERAL

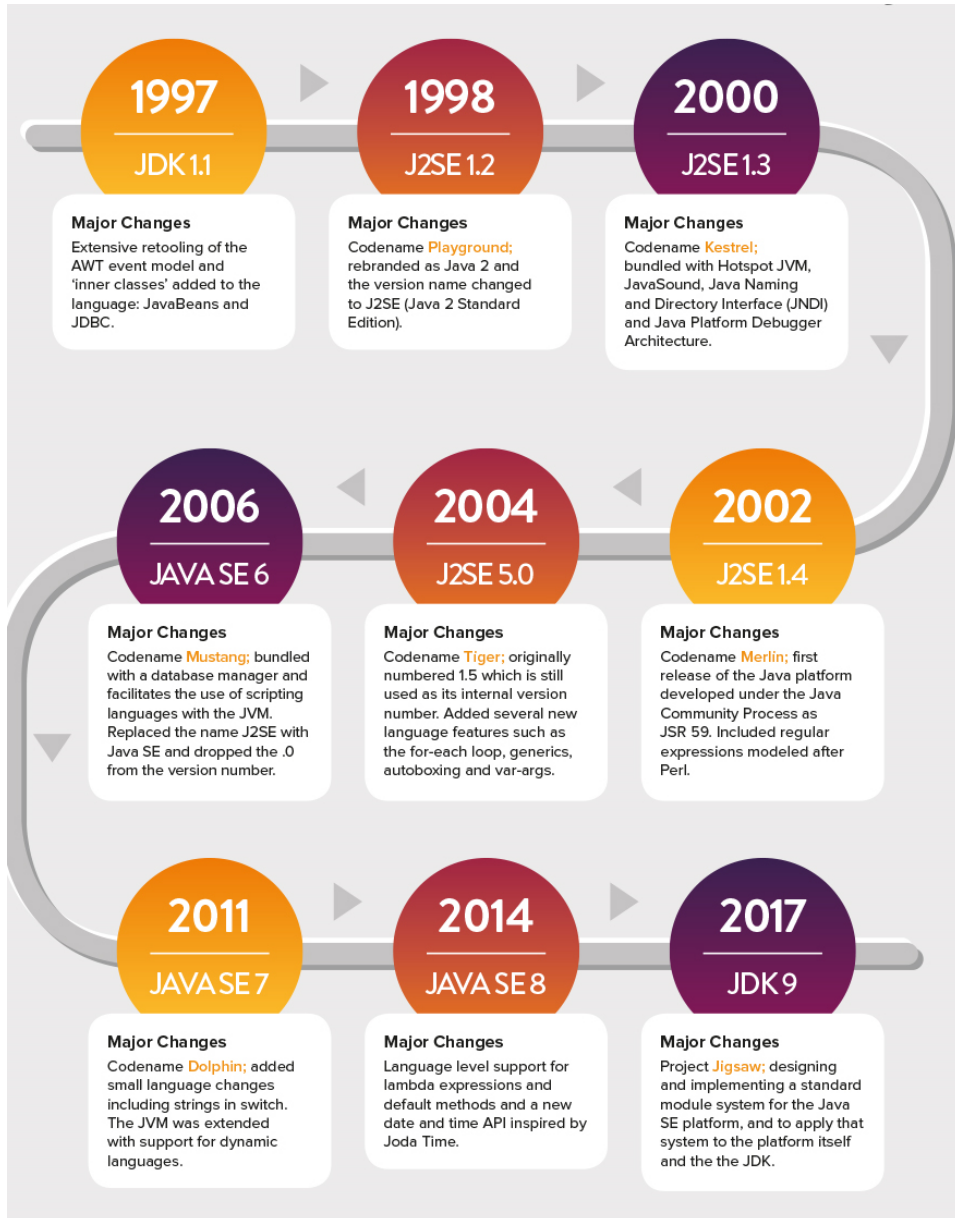


Write Once Run Any where



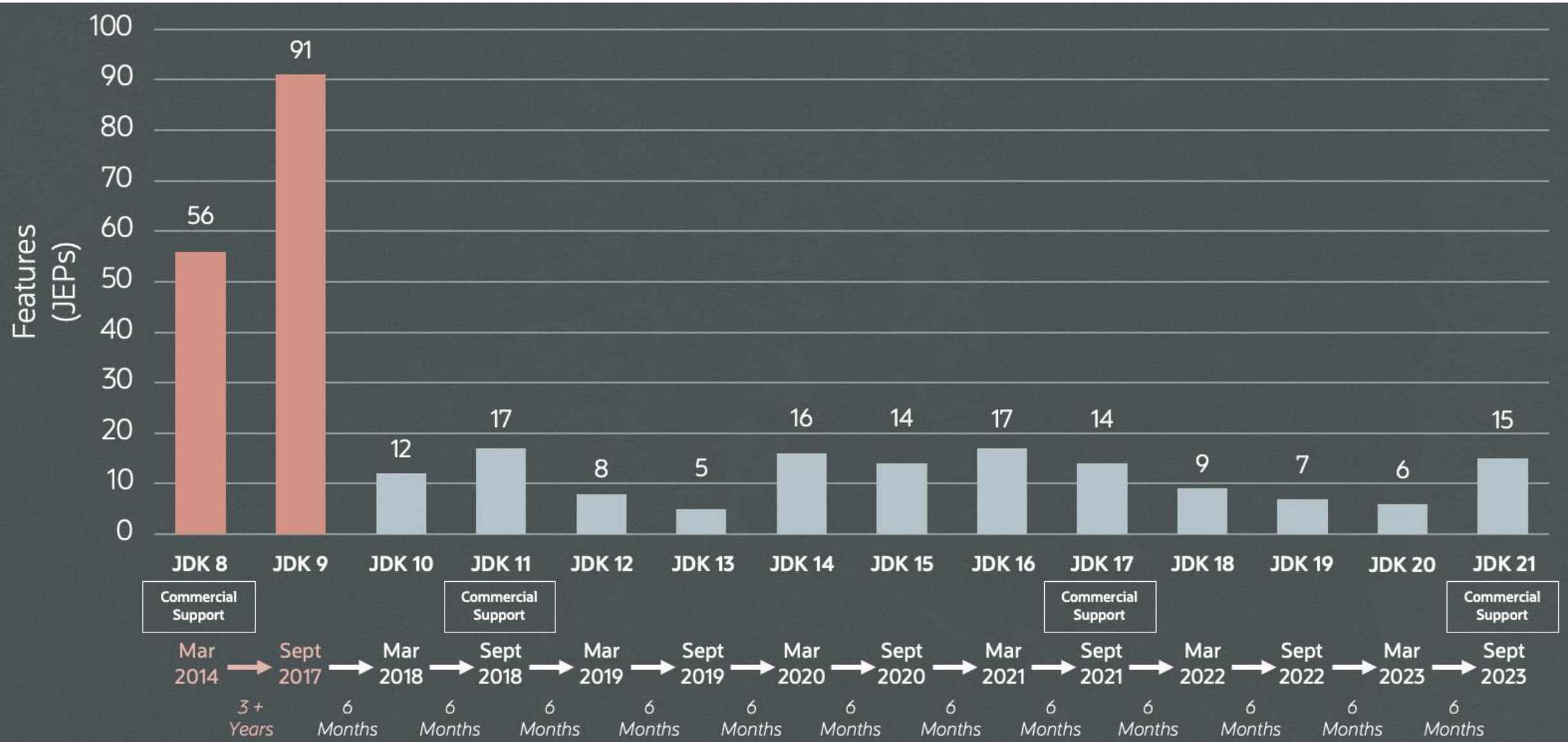
<https://dicasdejava.com.br/>

JAVA TIMELINE



- Oracle decidiu lançar uma nova versão a cada **6 meses (Mar / Set)** após 2017
- Java 10 (Março 2018)
- Java 11 (Setembro 2019)
- Java 12 (Março 2019)
- Java 13 (Setembro 2019)
- Java 14 (Março 2020)
-
- **Java 21 (Setembro 2023)**

Timeline



NOVIDADES NO JAVA

Java Text Blocks -> <https://www.baeldung.com/java-text-blocks>

Java 8 String Joiner -> <https://www.baeldung.com/java-string-joiner>

Binary Numbers in Java -> <https://www.baeldung.com/java-binary-numbers>

Java – Random Long, Float, Integer and Double -> <https://www.baeldung.com/java-generate-random-long-float-integer-double>

Period and Duration in Java -> <https://www.baeldung.com/java-period-duration>


INTRODUÇÃO A ORIENTAÇÃO A OBJETO

- Um modelo de programação ou paradigma de programação é um conjunto de princípios, ideias, conceitos e abstrações utilizado para o desenvolvimento de uma aplicação
- O modelo de programação mais adotado no desenvolvimento de sistemas corporativos é o modelo orientado a objetos.

INTRODUÇÃO A ORIENTAÇÃO A OBJETO

- O principal objetivo da programação orientada a objetos é **facilitar a manutenção das aplicações.**
- Orientação a objetos é uma maneira de programar que ajuda na organização e resolve muitos problemas enfrentados pela **programação procedural.**

OVERVIEW



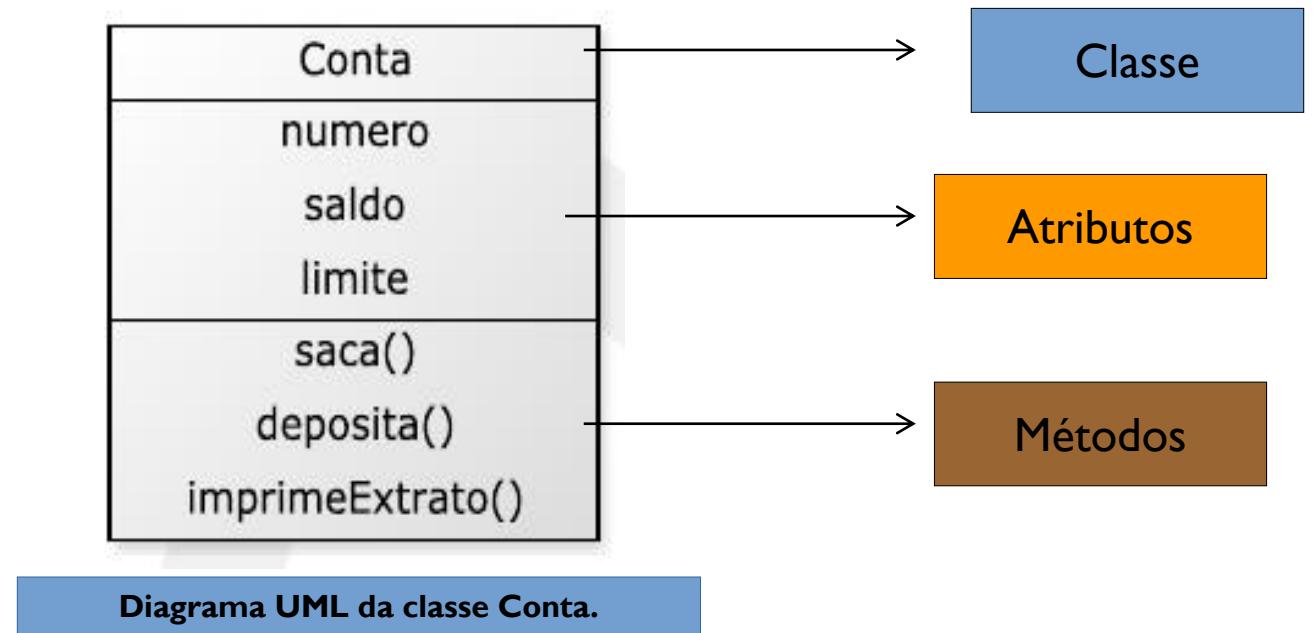
**Classes
(Objetos)**

Atributos

Métodos

CLASSES

Podemos representar uma classe através de diagramas UML. O diagrama UML de uma classe é composto pelo nome da mesma e pelos atributos e métodos que ela define.



CLASSES

Declaração de uma classe em Java:

```
[<modificadores_da_classe>] class <nome_da_classe>  
  
    [extends <nome_superclasse>]  
  
    [implements <interface_1>, <interface_2>, ...] {  
  
        // variáveis e métodos da classe  
    }
```

INSTALANDO O JDK

Para iniciar, vamos baixar um Java Development Kit (JDK). Este kit contém bibliotecas e executáveis que nos permitem fazer muitas coisas com código Java.

Mais importante ainda, com nosso JDK instalado, seremos capazes de compilar o código Java e, em seguida, executar programas Java completos.

Acesse: <https://www.oracle.com/java/technologies/javase-downloads.html>

- Baixar a versão JDK 15.0.2 (Windows Installer)

CONFIGURANDO AS VARIÁVEIS JAVA_HOME e PATH

Ao preparar o ambiente de desenvolvimento Java, precisamos instalar o JDK e **configurar onde a instalação do JDK está localizada**, para isso usamos a variável de ambiente JAVA_HOME e a variável PATH.

Links uteis:

- <https://mauriciogeneroso.medium.com/configurando-java-4-como-configurar-as-vari%C3%A1veis-java-home-path-e-classpath-no-windows-46040950638f>
- https://www.java.com/pt-BR/download/help/path_pt-br.html
- https://www.youtube.com/watch?v=EV4Hqvl_N8c
- <https://www.youtube.com/watch?v=jARiy3DZdwg>

FERRAMENTAS PARA DESENVOLVIMENTO

NetBeans 20

<https://netbeans.apache.org/download/nb122/nb122.html>

Eclipse

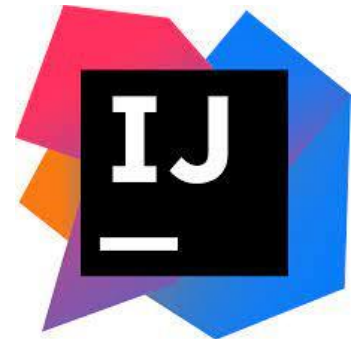
Intelij Community



NetBeans



eclipse



VALIDANDO A INSTALAÇÃO E CONFIGURAÇÃO

Para conferir e validar a instalação do Java, podemos fazer de 2 formas:

- **Via prompt de comando**

- Abre o prompt de comando (cmd no Windows) e digite `java -version + <Enter>`

- **Eclipse ou NetBeans IDE**

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Testando a configuração do Java ... ");  
        System.out.println("Versão do Java:" + System.getProperty("java.version"));  
    }  
}
```

PRÓXIMA AULA

- Criar projetos e classes em Java
- Primeiro programa (Hello World)
- The main() function



The screenshot shows the W3Schools website's Java tutorial page. The browser's address bar displays 'w3schools.com/java/default.asp'. The site's navigation bar includes links for 'Tutorials', 'Exercises', 'Certificates', and 'Services', along with a search bar. A secondary navigation bar lists various programming topics: 'HTML', 'CSS', 'JAVASCRIPT', 'SQL', 'PYTHON', 'JAVA' (which is highlighted in green), 'PHP', and 'HO'. A sidebar on the left contains a 'Java Tutorial' menu with 'Java HOME' highlighted in green, followed by links to 'Java Intro', 'Java Get Started', 'Java Syntax', 'Java Output', 'Java Comments', 'Java Variables', 'Java Data Types', 'Java Type Casting', 'Java Operators', 'Java Strings', 'Java Math', 'Java Booleans', and 'Java If...Else'. The main content area features the title 'Java Tutorial' and a green button labeled '< Home'. Below this, a green box contains the heading 'Learn Java', a brief description of Java as a popular programming language used for mobile apps and web development, and a green button labeled 'Start learning Java now »'.

