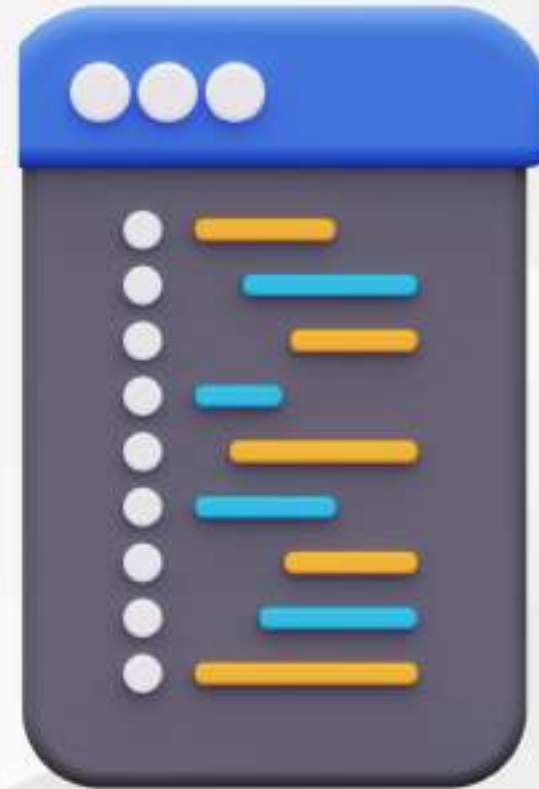


# POO

Programação Orientada a Objetos

Material 002

Professor Maromo



# Agenda

JAVA - HISTÓRICO



Histórico da plataforma e da linguagem Java

LINGUAGEM



Características, tecnologias Java, porque usar

AMBIENTE



Apresentação da plataforma e do ambiente Java



# Histórico



- **Projeto Green da Sun Microsystems (1991)**
  - Patrick Naughton, Mike Sheridan e James Gosling.
- **Primeira demonstração do projeto (1992)**
  - Sistema em handheld com interface a touchscreen [Star seven].
- **Lançamento da Plataforma Java (1995)**
  - John Gage (Sun) e Marc Andreessen (Netscape), anunciaram a JVM e a API Java. Inserido no Netscape Navigator.





**John Gage**

[http://upload.wikimedia.org/wikipedia/commons/d/de/John\\_Gage.jpg](http://upload.wikimedia.org/wikipedia/commons/d/de/John_Gage.jpg)



**Marc Andreessen**

[http://farm4.static.flickr.com/3285/2960434767\\_4014c753f9\\_o.jpg](http://farm4.static.flickr.com/3285/2960434767_4014c753f9_o.jpg)



**Patrick Naughton**

[http://s3.amazonaws.com/quarkbase\\_test.com/patrick-naughton-41297.jpg](http://s3.amazonaws.com/quarkbase_test.com/patrick-naughton-41297.jpg)



**James Gosling**

<http://www.javaalone.com/wp-content/uploads/2010/06/james->



# **Características da Linguagem**

[github.com/maromo71](https://github.com/maromo71) 

## Características da Linguagem

- Simples
- Orientada a Objetos
- Multithread
- Interpretada
- Neutra de Arquitetura
- Portável
- Robusta
- Segura
- Alto desempenho





## Simples



- Permite desenvolvimento de software em diferentes SOs e arquitetura de hardware. [Programador não se preocupa com a infraestrutura]
- Criadores de Java optaram por não implementar o uso do conceito de herança múltipla, de sobrecarga de operadores, ponteiros nem a operação aritmética com esse tipo de dado. [MENDES, 2009]

## Orientada a Objetos



- Java é uma linguagem de programação que nasceu orientada a objetos.
- Foi projetada dessa forma desde o seu início, e essa é uma das características fundamentais da linguagem.



## Multithread



- Em Java, você pode criar múltiplos threads de execução para realizar várias tarefas ao mesmo tempo.
- Isso é particularmente útil para aplicações que requerem alta responsividade ou que precisam realizar várias operações simultaneamente, como servidores web, aplicações de processamento de dados em tempo real, entre outras aplicações.



## Interpretada



- Java combina aspectos de linguagens compiladas e interpretadas. O código é primeiro compilado em bytecode, que é então interpretado ou compilado em tempo de execução pela JVM. Essa abordagem permite a portabilidade entre diferentes plataformas de hardware, ao mesmo tempo em que busca otimizar o desempenho através da compilação JIT.



## Independência de Arquitetura



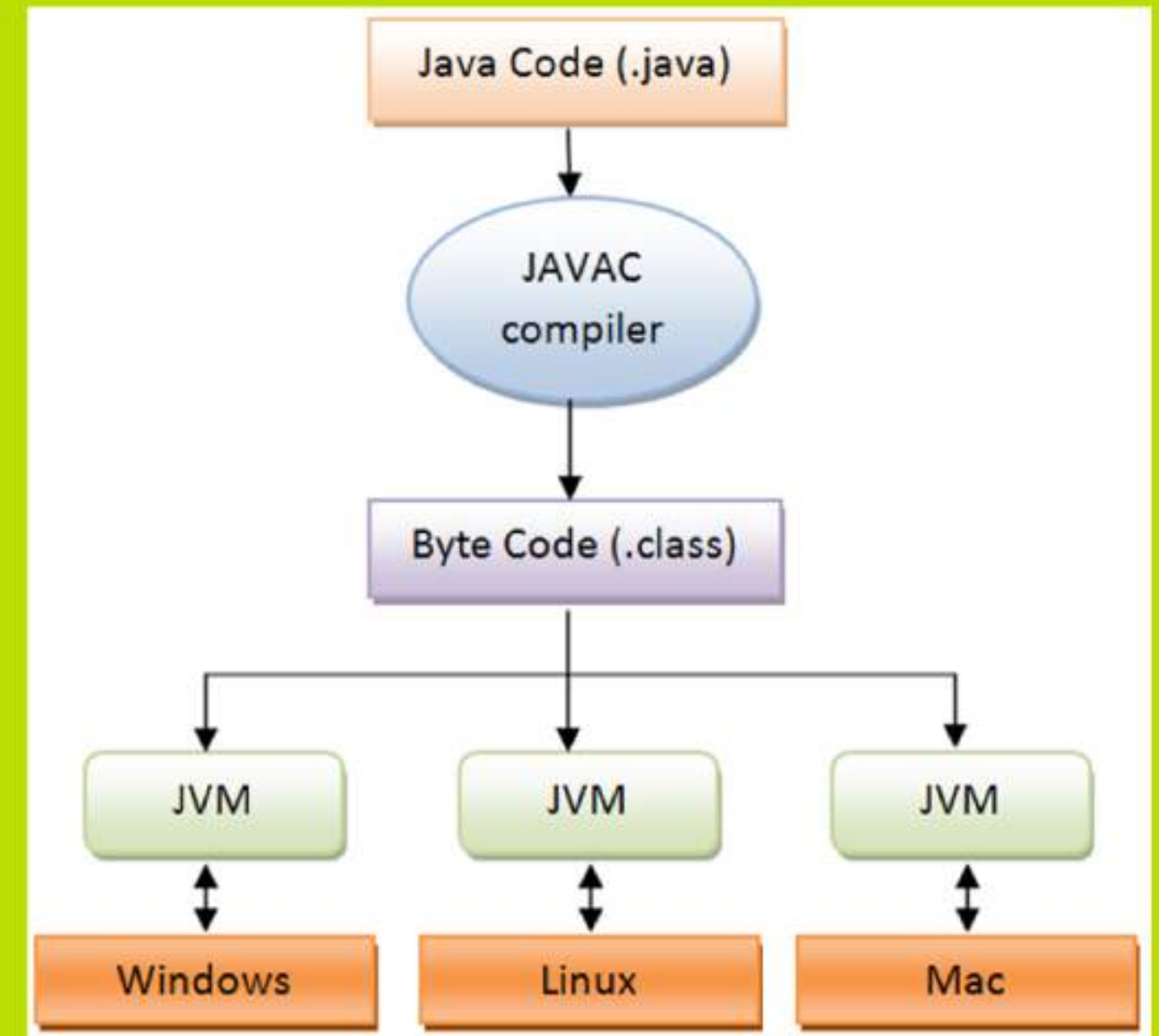
- Uma das características fundamentais de Java é a sua independência de arquitetura, frequentemente resumida pelo slogan "escreva uma vez, execute em qualquer lugar" (write once, run anywhere - WORA).
- Essa independência é alcançada através do uso de bytecode e da Máquina Virtual Java (JVM):
  - **Bytecode:** Quando você compila um programa Java, o código-fonte é transformado em bytecode, que é uma representação intermediária do código. O bytecode é independente de plataforma e não é específico para qualquer sistema operacional ou arquitetura de hardware.
  - **Máquina Virtual Java (JVM):** A JVM é responsável por executar o bytecode. Existem implementações da JVM para quase todas as principais plataformas de hardware e sistemas operacionais. A JVM interpreta ou compila o bytecode em código de máquina nativo específico para a plataforma em que está sendo executado.



## Portabilidade



- A portabilidade de Java torna a linguagem particularmente atraente para desenvolvedores que desejam criar software que possa ser facilmente distribuído e executado em uma variedade de dispositivos e sistemas operacionais





## Alto Desempenho

- Executa um código que foi previamente analisado e convertido para bytecodes.
- Recursos de garbage collector (executado em segundo plano).





## Robusta



- **Verificação de Tipo Forte:** Java é uma linguagem fortemente tipada, o que significa que os tipos de todas as variáveis devem ser conhecidos em tempo de compilação. Isso ajuda a detectar erros de tipo cedo no processo de desenvolvimento, aumentando a confiabilidade do código.
- **Manipulação de Exceções:** Java possui um sistema de manipulação de exceções que força os desenvolvedores a lidar com situações excepcionais, como falhas de entrada/saída. Isso promove a escrita de código que pode se recuperar graciosamente de erros inesperados.
- **Restrições de Acesso:** Java permite que os desenvolvedores definam níveis de acesso (como public, private, protected) para classes, métodos e variáveis. Isso ajuda a encapsular o código e proteger a integridade dos dados.



Segura



- Java foi projetada com a segurança em mente, incluindo recursos como a verificação de **bytecode** e **restrições de acesso a classes e métodos**. Isso ajuda a proteger o código contra atividades maliciosas.
- Pode-se garantir que, em um ambiente de rede, nenhum programa Java permitirá que outro programa escrito em qualquer linguagem possa se esconder em um código Java a fim de se instalar automaticamente (MENDES, 2009).





# AMBIENTE JAVA



# Ambientes de desenvolvimento Java



- JSE (Java Standard Edition)
  - Seu uso é voltado a PCs e servidores.
  - Contém todo o ambiente necessário para a criação e execução de aplicações desktop e web de pequeno e médio porte.
- Pode-se dizer que essa é a plataforma principal, já que, o JEE e o JME tem sua base aqui.



# Ambientes de desenvolvimento Java



- JEE (Java Enterprise Edition)
  - Voltada para o desenvolvimento de softwares corporativos.
  - Baseados em componentes que são executados em um servidor de aplicação.
- JME (Java Micro Edition)
  - Ambiente de desenvolvimento para dispositivos móveis ou portáteis, como telefones celulares e palmtops.



# Ambientes de desenvolvimento Java



- JRE (Java RuntimeEnvironment)
  - Significa Ambiente de Tempo de Execução
  - É um pacote de softwares, que é executado como um aplicativo do sistema operacional e que interpreta a execução de programas Java
- A JRE é composta pela JVM somada ao conjunto de API's. (JVM + API's = JRE)



# Ambientes de desenvolvimento Java



- API (Application Programming Interface)
  - Significa Interface de Programação de Aplicativos
  - Biblioteca (ou uma série delas) com funções e procedimentos públicos que permitem aos programadores desenvolverem aplicações fazendo uso de recursos já definidos.



# Ambientes de desenvolvimento Java



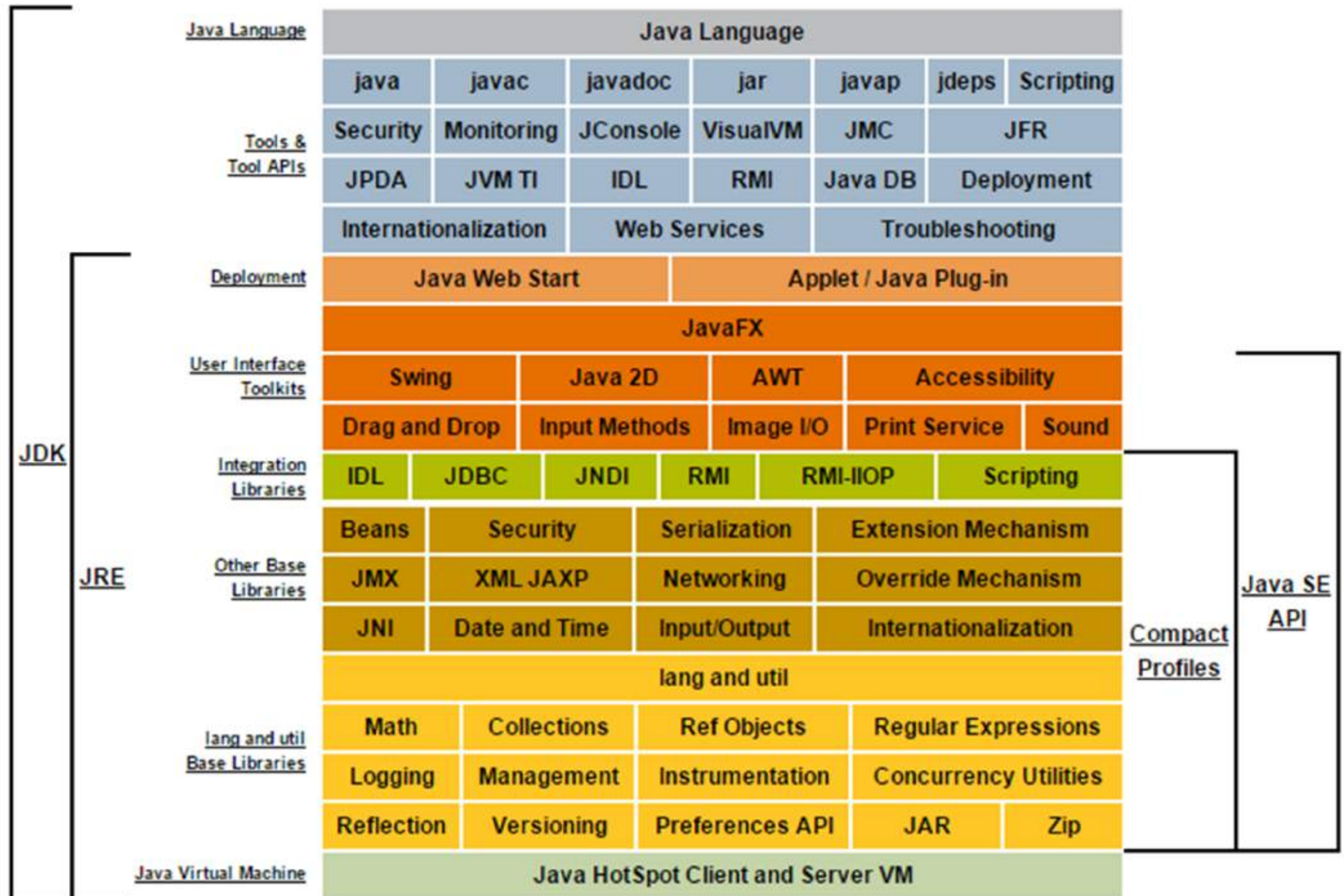
- JVM (Java Virtual Machine)
  - Significa Máquina Virtual Java
  - Software que emula uma CPU e Memória para a execução de programas Java.

# Ambientes de desenvolvimento Java



- JDK (Java Development Kit) ou SDK (Software Development Kit)
  - Significa Kit de Desenvolvimento Java
  - Conjunto de ferramentas para a compilação, documentação e debug de aplicativos Java.
  - Composto pela JRE somada as ferramentas de desenvolvimento.









# TIPOS DE PROGRAMA JAVA



# Tipos de programa

- **Stand-Alone:**

- Aplicação baseada na JSE, que tem total acesso aos recursos do sistema (memória, disco, rede, dispositivos, etc)
  - Um servidor pode executar uma aplicação Stand-Alone, por exemplo, um WebServer.
- Uma estação de trabalho pode executar uma aplicação de Automação Comercial.



# Tipos de programa

- **Java Applets:**

- Pequenas aplicações, que não tem acesso aos recursos de hardware e depende de um navegador que suporte a JSE para serem executados, geralmente usados para jogos, animações, teclados virtuais, etc.

- **Java Servlets:**

- Programas escritos e preparados para serem executados dentro de servidores web baseados em JEE, geralmente usados para gerar conteúdo dinâmico de websites.



# Tipos de programa

- **Java Midlets:**

- Pequenas aplicações, extremamente seguras, e construídas para serem executadas dentro da JME, geralmente, celulares, Palm Tops, controladores eletrônicos, computadores de bordo, smart cards, tecnologia embarcada em veículos, etc.

- **JavaBeans:**

- Pequenos programas, que seguem um padrão bem rígido de codificação, e que tem o propósito de serem aproveitados em qualquer tipo de programa Java, sendo reaproveitados, e podendo ser chamados a partir de: stand-alone, applets, servlets e midlets













- **Java é Lento:** Com a introdução de tecnologias como a compilação Just-In-Time (JIT) e melhorias contínuas na JVM, Java tornou-se muito mais rápido. Em muitos casos, o desempenho de Java é comparável ao de linguagens compiladas nativamente.
- **Java é o Mesmo que JavaScript:** Apesar dos nomes semelhantes, Java e JavaScript são linguagens completamente diferentes. Java é uma linguagem de programação fortemente tipada e orientada a objetos, enquanto JavaScript é uma linguagem de script dinamicamente tipada usada principalmente para desenvolvimento web.
- **Java Só é Usado para Desenvolvimento Web:** Embora Java seja popular no desenvolvimento de aplicações web, é usado em uma ampla variedade de aplicações, incluindo aplicações móveis (Android), sistemas embarcados, aplicações empresariais, e muito mais.
- **Java Não é Adequado para Aplicações de Pequena Escala:** Algumas pessoas acreditam que Java é adequado apenas para grandes projetos empresariais. No entanto, Java pode ser usado efetivamente em projetos de qualquer tamanho, desde pequenos scripts até grandes sistemas distribuídos.



- **Java é Inseguro:** Embora tenham ocorrido vulnerabilidades de segurança no passado, especialmente relacionadas a applets Java, a linguagem em si possui recursos robustos de segurança. A Oracle e a comunidade Java continuam a trabalhar para corrigir vulnerabilidades e melhorar a segurança.
- **Java é Verbose:** Algumas críticas apontam que Java requer mais linhas de código para realizar tarefas simples em comparação com algumas linguagens modernas. Embora isso possa ser verdade em alguns casos, a verbosidade também pode contribuir para a legibilidade e manutenção do código. Além disso, as versões mais recentes de Java introduziram recursos que reduzem a verbosidade.
- **Java Está Morrendo:** Apesar de declarações ocasionais de que Java está se tornando obsoleto, continua sendo uma das linguagens de programação mais populares e amplamente usadas no mundo. A linguagem continua a evoluir, com novas versões trazendo melhorias e recursos modernos.





Aug 2023	Aug 2022	Change	Programming Language		Ratings	Change
1	1			Python	13.33%	-2.30%
2	2			C	11.41%	-3.35%
3	4	▲		C++	10.63%	+0.49%
4	3	▼		Java	10.33%	-2.14%
5	5			C#	7.04%	+1.64%
6	8	▲		JavaScript	3.29%	+0.89%
7	6	▼		Visual Basic	2.63%	-2.26%
8	9	▲		SQL	1.53%	-0.14%
9	7	▼		Assembly language	1.34%	-1.41%
10	10			PHP	1.27%	-0.09%



## Exemplo de Programa

---

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```





# Obrigado

# **fim**

Até a próxima aula







## Referências Bibliográficas

- ❑ Mendes; **Java com Ênfase em Orientação a Objetos**, Novatec.
- ❑ Deitel; **Java, como programar** – 10º edição. Java SE 7 e 8
- ❑ Arnold, Gosling, Holmes; **A linguagem de programação Java** – 4º edição.
- ❑ Apostilas da Caelum.