TREINA RECIFE
DESENVOLVIMENTO GERENCIAL E PROFISSIONAL

Introdução ao Java

Objetivos do curso

Neste curso você irá aprender a implementar a lógica da aplicação usando a linguagem Java:

- Descrever a abordagem orientada a objetos (OO)
- Explicar a sintaxe do Java e convenções
- Usar operadores e construções
- Usar a API Java
- Desenvolver uma aplicação exemplo

Estrutura do curso

- Apresentações
- Introdução ao Java
- Tipos primitivos, operadores e fluxo de controle
- Texto, data, hora e objetos numéricos
- Classes e objetos
- Herança
- Interfaces
- Matrizes, vetores e laços
- Collections, generics
- Exceptions
- Aplicação, JDBC, Annotations

Dinâmica das aulas

- No início do encontro faremos uma breve recapitulação do conteúdo acumulado até o momento.
- Atividades práticas serão o foco
- Atividades para casa serão recomendadas para fortalecer o conteúdo visto
- As intervenções teóricas serão utilizadas para apresentar os assuntos
- O material visual pode ser compartilhado com a turma, mas recomendamos que sejam explorados outros conteúdos online

Ferramentas

- JDK 11 / OpenJDK 11
  - https://www.oracle.com/br/java/technologies/javase/jdk11-archive-downloads.html
  - https://jdk.java.net/archive/

- Eclipse IDE for Enterprise Java and Web Developers
  - https://www.eclipse.org/downloads/packages/release/2023-12/r/eclipse-ide-enterprise-java-and-web-developers

- MySQL Community Server (v8.0)
  - https://dev.mysql.com/downloads/
  - MySQL Wokbench
  - Connector/J

Objetivos da Aula

- Origens da linguagem Java
- Java: portabilidade e neutralidade
- “Olá mundo”

WHAT IS JAVA

Java é uma linguagem de programação simples, de propósito geral, orientada a objetos, distribuída, interpretada, robusta, segura, independente de plataforma, portável, de alta performance, concorrente e dinâmica.

Um programa...

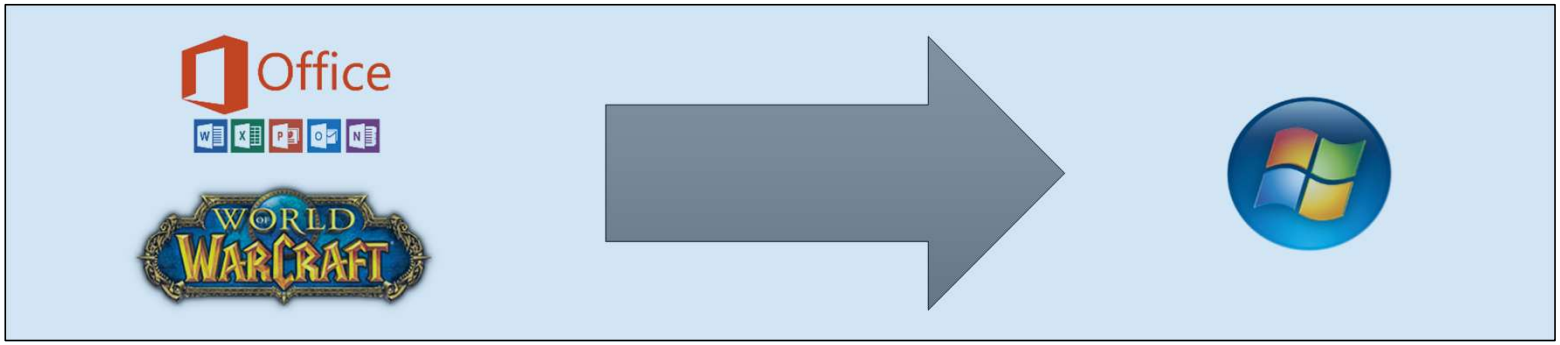
Código fonte | Compilador | Binário | Sistema operacional

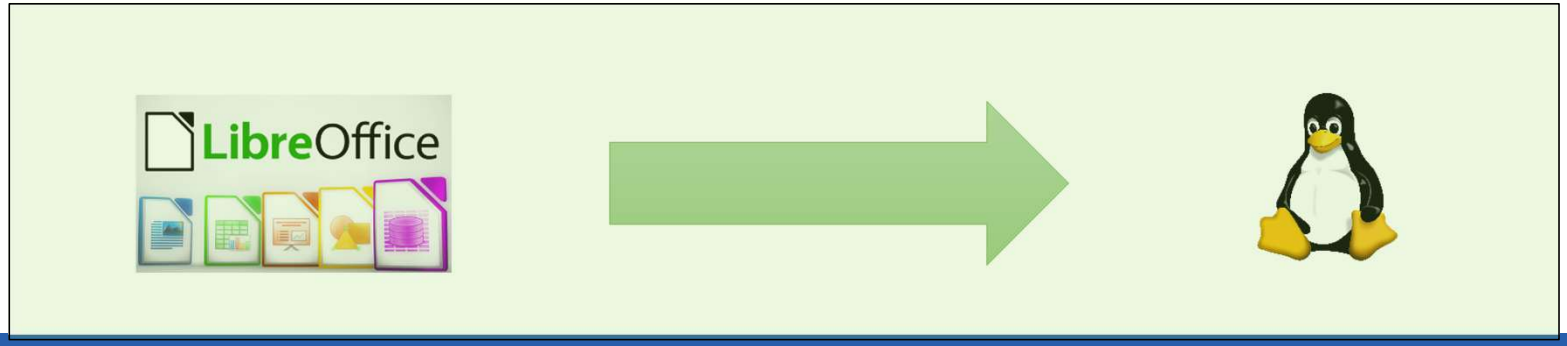
PROGRAMA

Um programa...

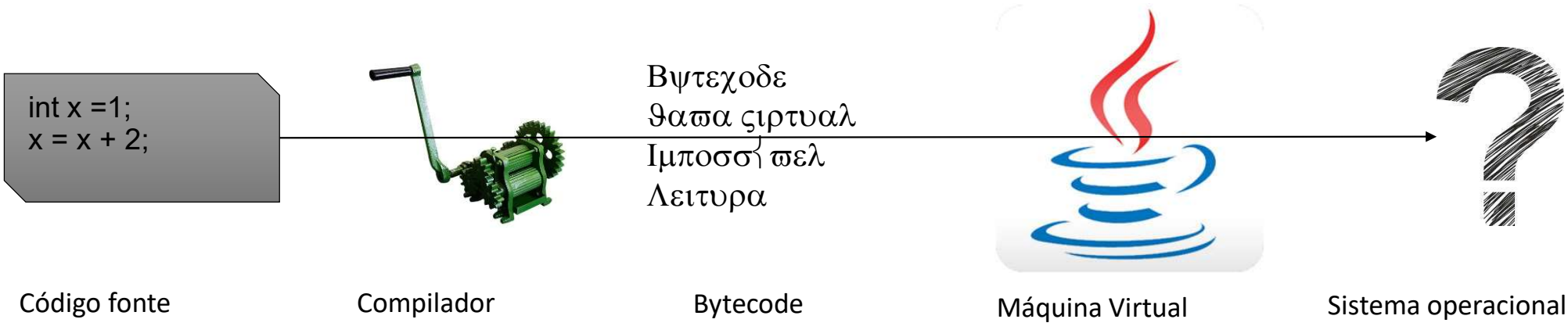
Binário

Sistema operacional





Em Java...

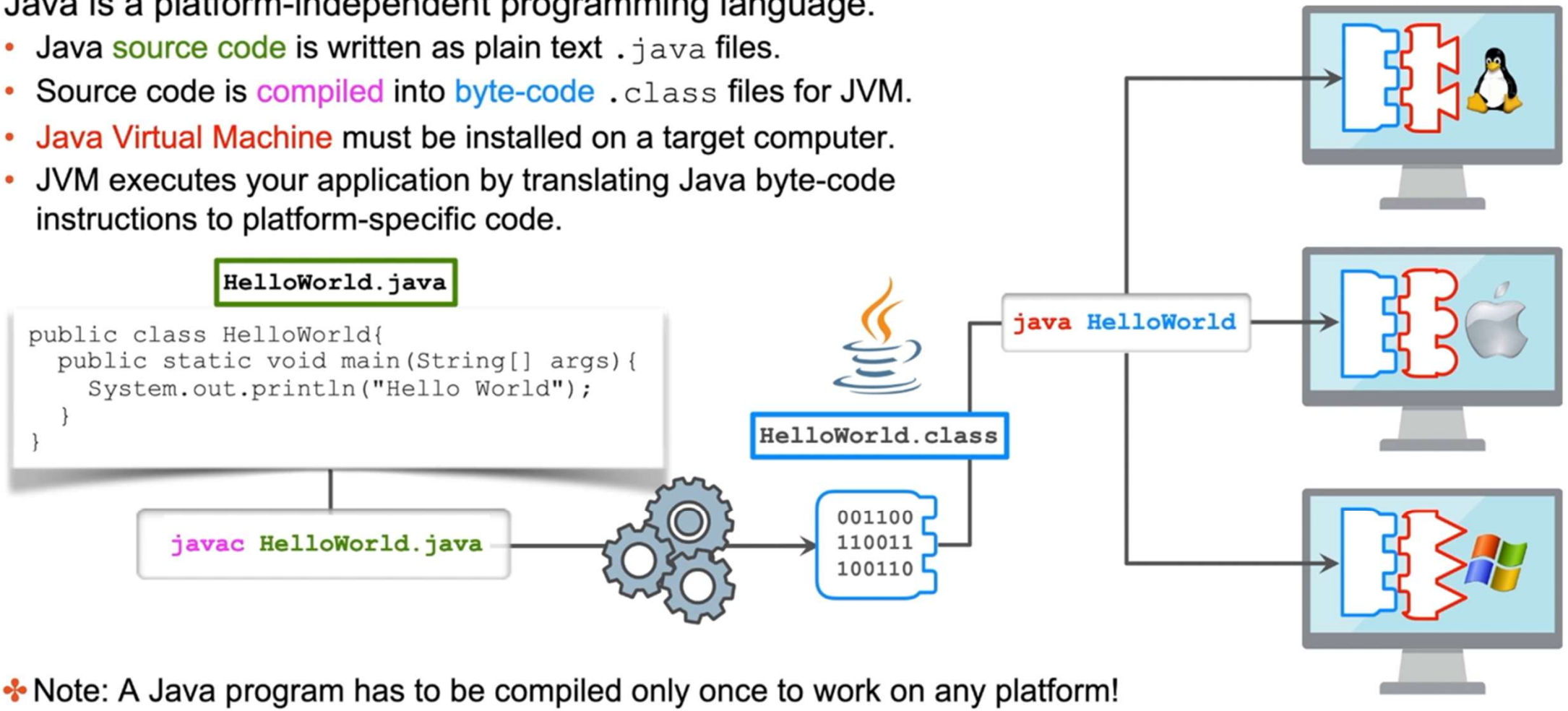

int x =1;
x = x + 2;
Βψτεχοδε
θαωα ςιρτυαλ
Ιμποσσ৲ ωελ
Λειτυρα
Código fonte
Compilador
Bytecode
Máquina Virtual
Sistema operacional


PROGRAMA

How Java Works?

Java is a platform-independent programming language.

- Java source code is written as plain text `.java` files.
- Source code is compiled into byte-code `.class` files for JVM.
- Java Virtual Machine must be installed on a target computer.
- JVM executes your application by translating Java byte-code instructions to platform-specific code.



✤ Note: A Java program has to be compiled only once to work on any platform!

Em palavras...

- Criamos o arquivo texto com extensão “.java”
- Efetuamos a compilação do código (javac);
  - O compilador gera um arquivo **.class** (*bytecode*);
- Ao ser executado, o *bytecode* é interpretado por uma “máquina virtual” instalada no computador ou dispositivo (JVM);

Plataforma

Plataformas

- **Java Card** (Smart card Edition)
- **Java ME** (Micro Edition)
- **Java SE** (Standard Edition). É a base da plataforma; inclui o ambiente de execução e as bibliotecas comuns.
- **Java MP** (Micro Profile)
- **Java EE** (Enterprise Edition). A edição voltada para o desenvolvimento de aplicações corporativas e para internet.

Composição


Java IDE
JDK - Java Development Kit
Developer
Development Tools
java, javac,JShell, javadoc, jar, jdeps,JConsole, VisualVM,
JFR, JPDA, JVM TI, IDL, Java DB, Debugging Tools, Deployment Tools, Monitoring Tools
JRE - Java Runtime Environment
Java App Users
Java API, Runtime Libraries, Byte Code Verifier,
Libraries: User Interface (Swing, JavaFx, Sound, Image I/O .....),
Integration Libraries (JDBC, RMI, . .), Serialization, Netowrking,
lang and util libraries (Math, Collections, Concurrency, Logging, ...).....
JVM - Java Virtual Machine
Java Interpreter, JIT, Garbage Collector, Thread Synchronization,
Class Loader Subsystem
Windows
Linux
Mac
Outros

Composição

- **JVM** = Java Virtual Machine, “programa” responsável pela portabilidade do Java. Não pode ser baixado isoladamente.

- **JRE** = Java Runtime Environment, pacote composto pela JVM e bibliotecas, serve para rodar/executar aplicativos Java (class, jar).

- **JDK** = Java Development Kit, pacote composto pela JRE e demais ferramentas de desenvolvimento, como o compilador.

Kit de Desenvolvimento Java (JDK)

Bibliotecas (pacotes)

- Entrada e Saída
- Interface Gráfica
- Comunicação em Rede
- Threads
- ...

Ferramentas

- Compilador – javac
- Appletviewer
- Interpretador – java
- Gerador de Documentos – javadoc
- Gerador de arquivos – jar

Máquina Virtual Java (JVM)

- Lê bytecodes que são independentes de plataforma.
- Pode ser implementada em software ou hardware.
- Implementada em um ambiente de desenvolvimento Java ou navegador web.
- Provê definições para:
- Conjunto de instruções,
- Conjunto de registradores,
- Pilha,
- Heap,
- Área de memória,
- Formato de arquivos .class

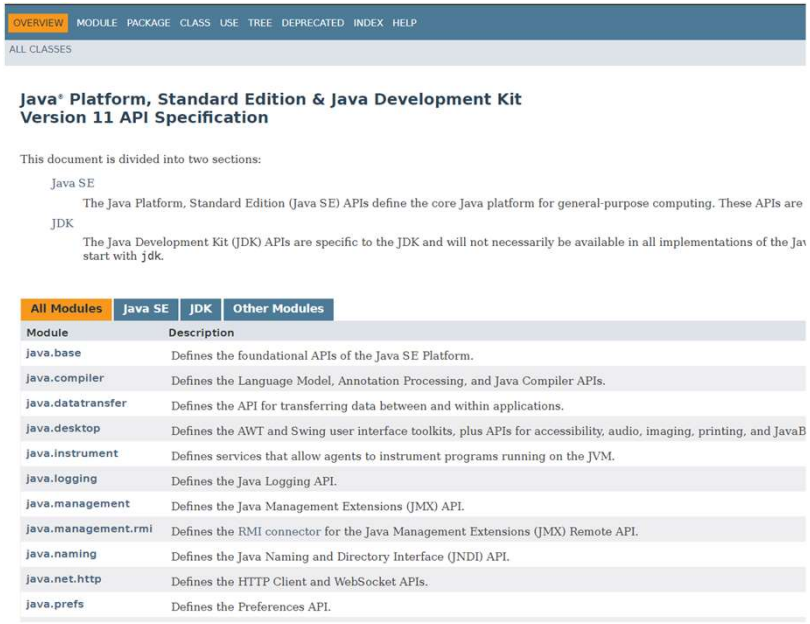
Garbage Collection

- Memória alocada e não mais utilizada deve ser liberada.
- Em outras linguagens o gerenciamento de memória é responsabilidade do programador.
- Sua performance varia entre implementações da JVM.
- É feita automaticamente e, por ser um processo de baixa prioridade, pode não realizar a limpeza de maneira eficiente.
- Os métodos abaixo “sugerem” a chamada do Garbage Collection:
  - System.gc()
  - Runtime.gc()

Documentação Java (API)

- Provê especificação para a Plataforma Java Standard Edition

https://docs.oracle.com/en/java/javase/11/docs/api/index.html



Ambiente Integrado de Desenvolvimento (IDE)

- **Eclipse** IDE
- JetBrains **IntelliJ** IDEA
- Microsoft Visual Studio Code (**VS Code**)
- Apache **Netbeans** IDE

* Notepad, Notepad++, Sublime

Ambiente

Do que iremos precisar

JDK (Open JDK 11)

Eclipse IDE

Git (github)

Variáveis de Ambiente

- São usadas para auxiliar na conversa entre o Java e o SO.
  - **PATH**, informa ao SO onde procurar os executáveis do Java.
  - **JAVA_HOME**, informa onde está o diretório de instalação do Java (JDK).
  - **CLASSPATH**, informa à JVM e aos outros aplicativos onde procurar as classes Java.

```
set JAVA_HOME=C:\Program Files\Java\jdk11
```

Olá Mundo!

OlaMundo.java

```
public class OlaMundo{
    public static void main(String[] args){
        System.out.println("Olá Mundo!");
    }
}
```

Java Keywords, Reserved Words, and a Special Identifier

Keywords available since 1.0
Keywords no longer in use
Keywords added in 1.2
Keywords added in 1.4
Keywords added in 5.0
Keywords added in 9.0

Reserved words for literals values

Special identifier added in 10

if	implements	boolean	assert
else	extends	try	enum
continue	interface	catch	module
break	class	finally	requires
for	static	throw	transitive
do	final	throws	exports to
while	return	new	uses
switch	transient	this	provides
case	void	super	with
default	byte	instanceof	opens to
private	short	native	
protected	int	synchronized	true
public	long	volatile	false
import	char	goto	null
package	float	const	
abstract	double	strictfp	var

Notes

- Keywords and Literals cannot be used as identifiers (names of classes, variables, methods, and so on).


É hora de recapitular