

Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Linguagem de Programação I

Introdução a Java

▶ Tiago Maritan

▶ tiago@ci.ufpb.br



Breve Histórico do Java

- ▶ 1991: Projeto Green (James Gosling - Sun)
 - ▶ Criação de uma linguagem para rodar em **receptores DTV**;
 - ▶ Dispositivos tinham poucos recursos computacionais:
 - ▶ Linguagem tinha que ser **pequena**, gerar **código robusto**;
 - ▶ E ser **portável** (independente de plataforma);
 - ▶ **Oak** : 1º nome da linguagem;
 - ▶ Já existia outra linguagem com esse nome;
 - ▶ Mudou- se o nome para **Java**
 - ▶ Café tomado pela equipe era da **ilha de Java**;



Breve Histórico de Java

- ▶ 1994: Explosão da Web;
- ▶ 1995: Criação do navegador **HotJava** (em Java)
 - ▶ Capaz de rodar código Java na Web (**Applets**);
 - ▶ Java vira uma febre;
- ▶ 1996: Lançamento da Plataforma Java 1.0
 - ▶ Matérias no *New York Times*, *Washington Post* e *Business Week*;

Breve Histórico do Java

- ▶ 1997: Java 1.1
- ▶ 1998: Java 2 (1.2)
 - ▶ Plataforma em 3 versões:
 - ▶ **Java Standard Edition (Java SE):** para desktops;
 - ▶ **Java Micro Edition (Java ME):** para dispositivos embutidos(celulares);
 - ▶ **Java Enterprise Edition (Java EE):** para servidores, apps. corporativas;
- ▶ 2000: Java 2 (1.3)
- ▶ 2003: Java 2 (1.4)

**A plataforma que
veremos no curso**



-
- ▶ 2005: Java 5 (1.5) - JDK 5 (Tiger)
 - ▶ 2006: Java 6 (1.6) - JDK 6 (Mustang)
 - ▶ 2009: Sun é vendida para a Oracle
 - ▶ 2011: Java 7 - JDK 7 (Dolphin)
 - ▶ **2014: Java 8 – JDK 8**
-
- ▶ No curso, estudaremos a versão **Java SE 8**;

Características de Java

- ▶ Familiar e Simples;
- ▶ Orientada a Objetos;
- ▶ Portável;
- ▶ Interpretada;
- ▶ Compatível com Redes;
- ▶ Suporte nativo a *threads*;
- ▶ Acesso Remoto a BD
- ▶ Segura;
- ▶ Robusta;

Características de Java

▶ **Familiar e Simples**

- ▶ Sintaxe parecida com C/C++;
- ▶ Porém mais limpa: sem ponteiros, estruturas, etc;

▶ **Orientada a Objetos (OO)**

- ▶ Prática de programação já consolidada no mercado;
- ▶ OO permite reutilização de diversos trechos de códigos;
- ▶ Recursos OO de Java são essencialmente os mesmos de C++;

▶ **Portável (Multiplataforma)**

- ▶ Programa Java pode rodar em **qualquer plataforma** (HW e SO) **sem precisar adaptar o código**;
- ▶ Ex: Windows, Linux, Mac OS, etc;

Características do Java

- ▶ **Interpretada (Uso de máquinas virtuais)**

- ▶ Interpretador roda *bytecodes* em qualquer máquina em que foi portado;

- ▶ **Comunicação em Rede**

- ▶ Extensa biblioteca para lidar com protocolos TCP/IP (HTTP, FTP, etc);
- ▶ Acesso a objetos pela Internet é similar ao acesso de arquivos locais;

- ▶ **Suporte Nativo a Threads**

- ▶ **Threads:** estratégia que permite que mais de um evento aconteça simultaneamente em um programa;
- ▶ Ex: Conexão de cada usuário é tratada por uma **thread**;
- ▶ Atraente para o desenvolvimento de servidores;

Características do Java

▶ **Acesso remoto a BD**

- ▶ Classes prontas para o acesso a BD

▶ **Segura**

- ▶ Vários mecanismos para controlar a segurança (criptografia dos dados);
- ▶ Desenvolvida para rodar em ambientes distribuídos (rede);
- ▶ Poucos *bugs* encontrados e todos muitos técnicos;

▶ **Robusta**

- ▶ Mecanismo eficiente para **tratamento de erros** e **coleta de lixo**;
- ▶ Minimiza *bugs*, aumenta produtividade;

Onde encontrar Java?

Java em todo lugar



97% dos Desktops usam Java

3 bilhões Celulares com Java

125 Milhões de TVs com Java

100% dos Blu-rays Players usam Java

9 milhões de desenvolvedores

*Fonte Oracle - <http://www.java.com/en/about/>

Popularidade das LPs

Feb 2016	Feb 2015	Change	Programming Language	Ratings	Change
1	2	⬆	Java	21.145%	+5.80%
2	1	⬇	C	15.594%	-0.89%
3	3		C++	6.907%	+0.29%
4	5	⬆	C#	4.400%	-1.34%
5	8	⬆	Python	4.180%	+1.30%
6	7	⬆	PHP	2.770%	-0.40%
7	9	⬆	Visual Basic .NET	2.454%	+0.43%
8	12	⬆	Perl	2.251%	+0.86%
9	6	⬇	JavaScript	2.201%	-1.31%
10	11	⬆	Delphi/Object Pascal	2.163%	+0.59%
11	20	⬆	Ruby	2.053%	+1.18%
12	10	⬇	Visual Basic	1.855%	+0.14%
13	26	⬆	Assembly language	1.828%	+1.08%
14	4	⬇	Objective-C	1.403%	-4.62%
15	30	⬆	D	1.391%	+0.77%



11

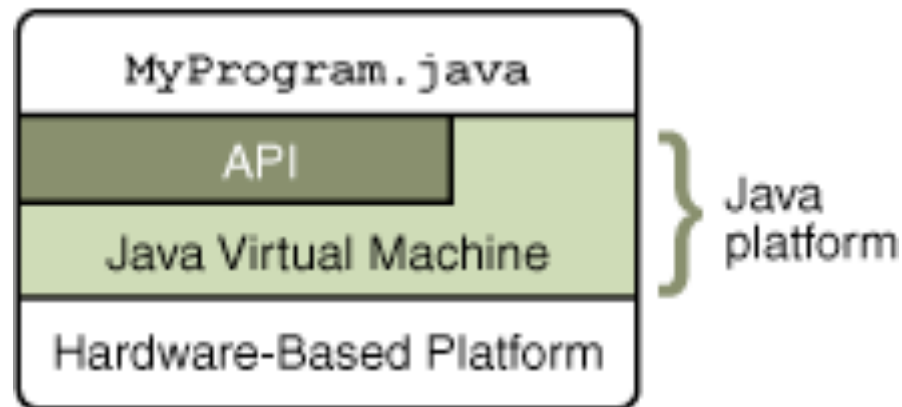
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

O que é (a tecnologia) Java?

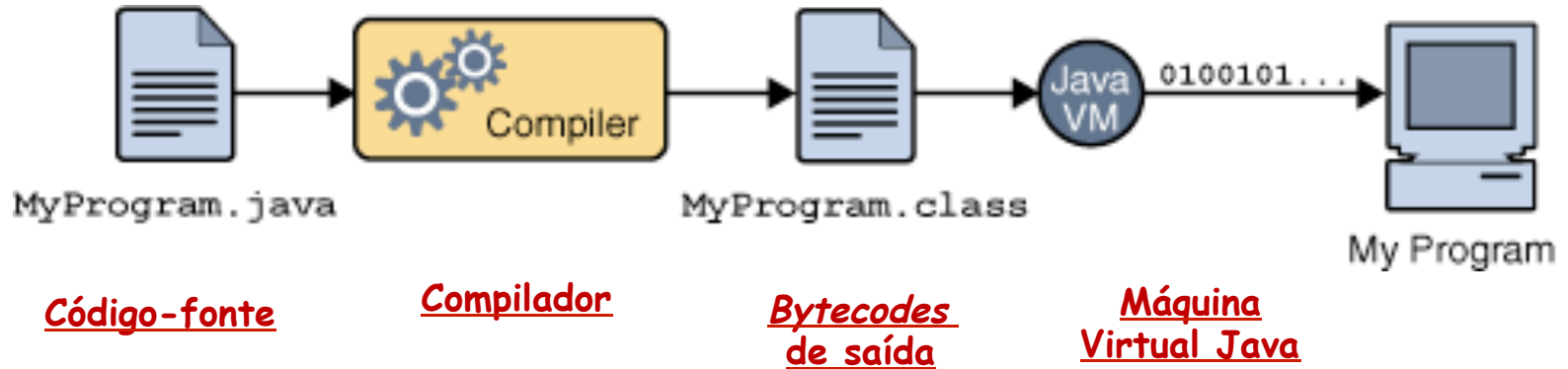
- ▶ Java não é apenas uma Linguagem de Programação;
- ▶ É uma tecnologia que contempla:
 - ▶ I Linguagem de Programação OO +
 - ▶ I Plataforma de Programação;
 - ▶ Conjunto de APIs (bibliotecas) e ambiente de desenvolvimento para aplicações;
 - ▶ Ambiente de execução (JVM) presente em:
 - browsers, mainframes, SOs,
 - TVs, celulares, tablets, eletrodomésticos, etc.

Plataforma Java

- ▶ **Plataforma:** ambiente de SW ou HW no qual um programa roda
 - ▶ Ex: Microsoft Windows, Unix, OS X;
- ▶ **Plataforma Java:** plataforma apenas de SW que roda sobre plataformas de HW e tem dois componentes:
 1. JVM (Java Virtual Machine)
 2. Java API



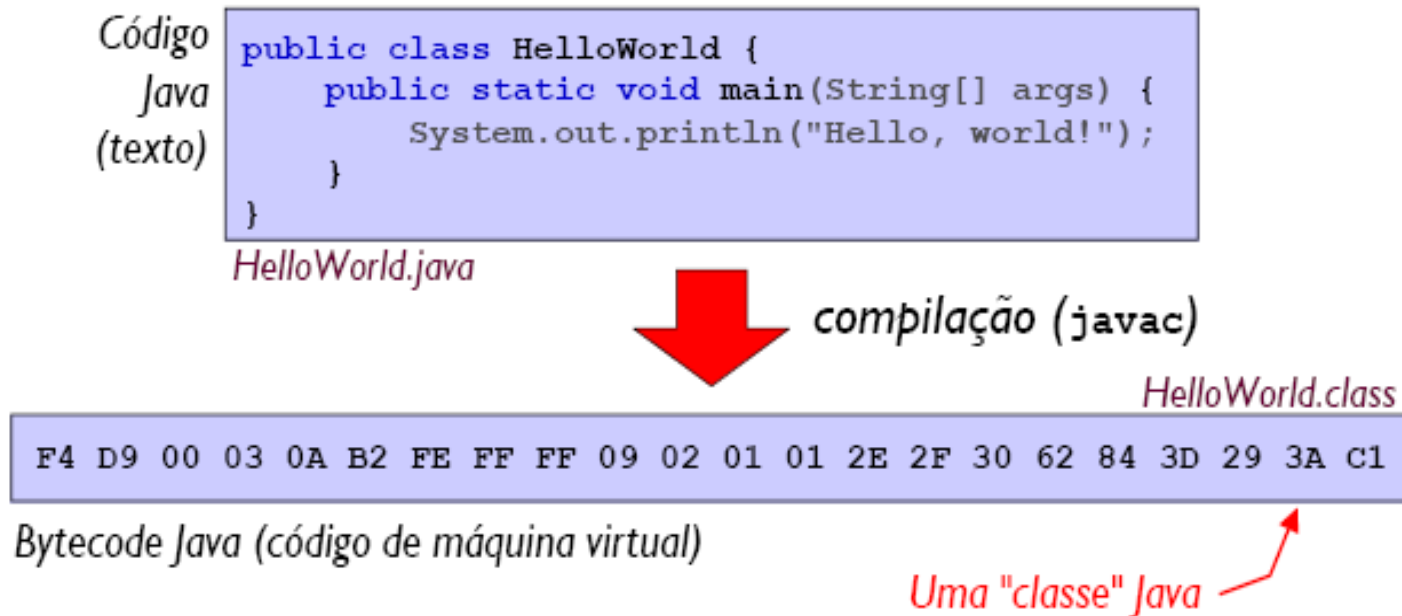
Criação de Programas Java



- ▶ Três tipos básicos de programas em Java:
 1. **Aplicações convencionais:** executadas no Desktop;
 2. **Applets:** executados no browser (cliente);
 3. **Servlets:** executados no lado servidor Web

Compilação para bytecode

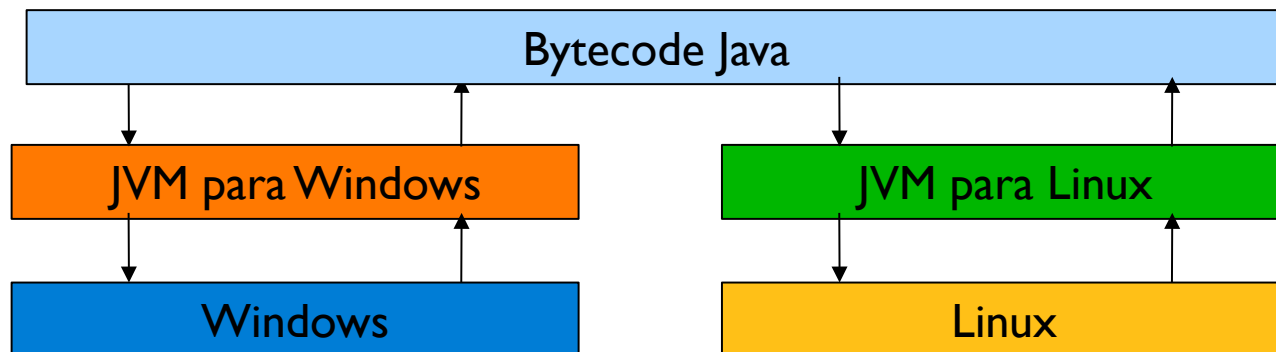
- ▶ Compilador traduz código Java para **bytecodes** (arquivo .class)
 - ▶ **Bytecode:** é um código de máquina que roda em qualquer máquina, basta que haja uma Máquina Virtual Java (JVM)



Máquina Virtual Java (JVM)

“Write once, Run Anywhere”

- ▶ No geral, um programa compilado em uma plataforma não roda em outra
 - ▶ Ex: Programa em C++ compilado no Windows não roda no Unix;
 - ▶ Necessário gerar uma versão do programa para cada plataforma;
- ▶ **Bytecodes + Máquina Virtual (JVM):**
 - ▶ Cada JVM converte os **bytecodes** para a plataforma específica
 - ▶ Mesmo programa roda igualmente em várias plataformas;



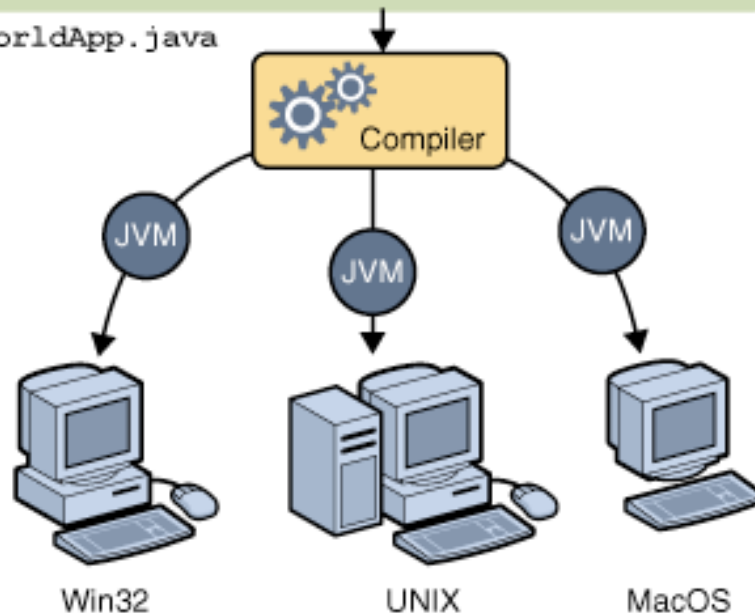
Máquina virtual Java

“Write once, Run Anywhere”

Java Program

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

HelloWorldApp.java

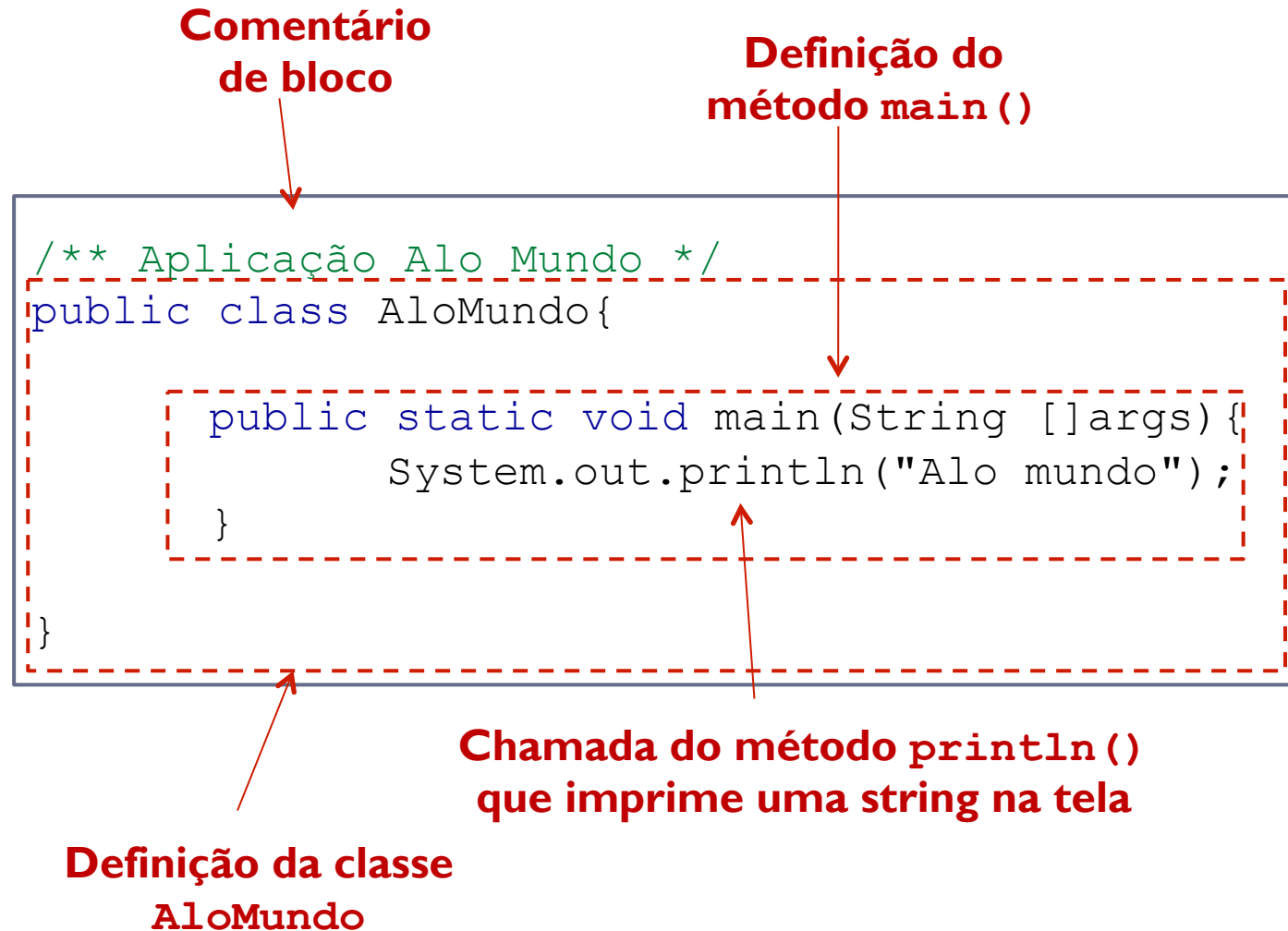


Primeiro Contato com Java

“Alo Mundo” em Java

```
/** Aplicação Alo Mundo */  
public class AloMundo{  
  
    public static void main(String []args){  
        System.out.println("Alo mundo");  
    }  
  
}
```

“Alo Mundo” em Java



Instalação do JDK

- ▶ Para compilar e rodar programas Java, é necessário antes instalar o **Kit de Desenvolvimento Java (JDK)**

I. Baixar e instalar o JDK 8:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Instalação do JDK

2. Configurar as variáveis de ambiente:

- ▶ **JAVA_HOME:** onde está o JDK;
- ▶ **PATH:** onde está o compilador, interpretador de Java;
- ▶ **CLASSPATH:** onde estão as bibliotecas de Java;
- ▶ **Windows:**
 - ▶ Caminho de instalação: **C:\Arquivos de Programas\Java\jdk.1.6.0_27**
 - ▶ Painel de Controle->Sistema->Avançado->Variáveis de ambiente:
 - ▶ JAVA_HOME = **C:\Arquivos de Programas\Java\jdk.1.6.0_27**
 - ▶ PATH = **C:\Arquivos de Programas\Java\jdk.1.6.0_27\bin**
 - ▶ CLASSPATH = **C:\Arquivos de Programas\Java\jdk.1.6.0_27\lib**

Instalação do JDK

► Verificar a instalação

► **Prompt de Comando:** “javac”;

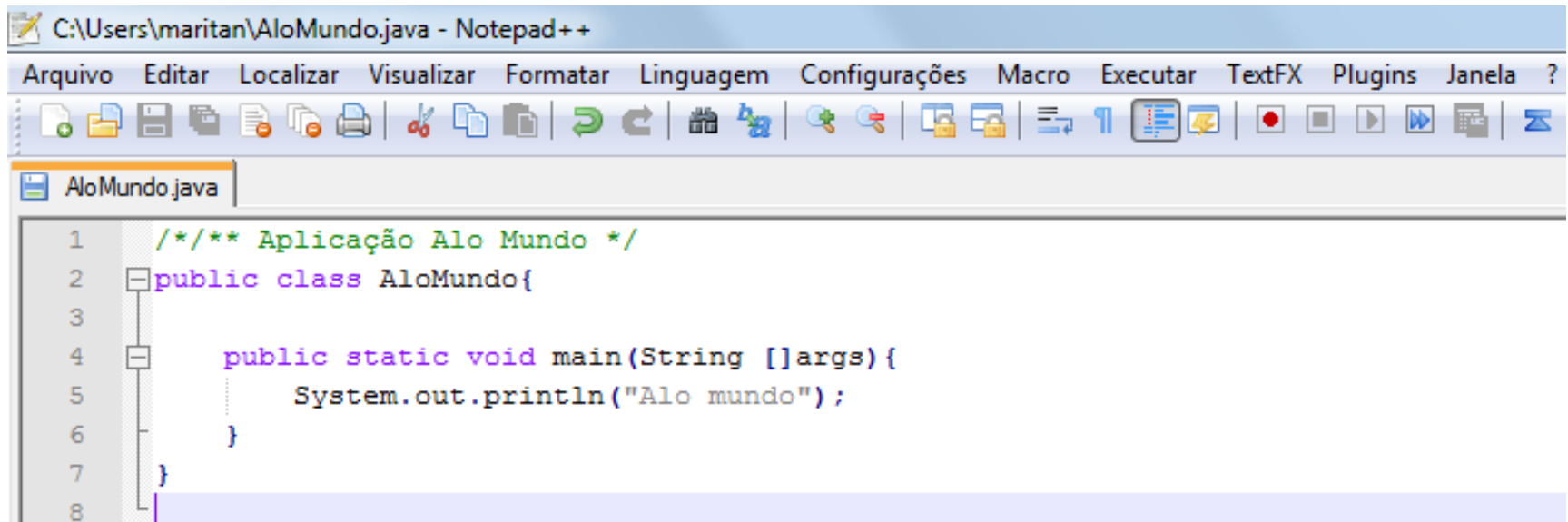
► Duas possibilidades:

1. “javac não é reconhecido como programa...” : **Problema na instalação**
2. Tela com instruções de help de Java: **Instalação OK!!!**

```
C:\Users\maritan>javac
Usage: javac <options> <source files>
where possible options include:
    -g                Generate all debugging info
    -g:none           Generate no debugging info
    -g:<lines,vars,source> Generate only some debugging info
    -nowarn           Generate no warnings
    -verbose          Output messages about what the compiler is doing
    -deprecation      Output source locations where deprecated
                        API is used
```

Escrever o programa

1. Escrever código do programa em um editor de texto;
 - ▶ Ex: Bloco de Notas, Notepad++, Vim, etc.
2. Salvar arquivo com a extensão `.java`
 - ▶ Nome arquivo = nome da classe (`AloMundo.java`)



The screenshot shows the Notepad++ editor with the file `AloMundo.java` open. The menu bar includes Arquivo, Editar, Localizar, Visualizar, Formatar, Linguagem, Configurações, Macro, Executar, TextFX, Plugins, and Janela. The toolbar contains various icons for file operations and editing. The code editor displays the following Java code:

```
1  /**/ ** Aplicação Alo Mundo */
2  public class AloMundo{
3
4      public static void main(String []args){
5          System.out.println("Alo mundo");
6      }
7  }
8
```

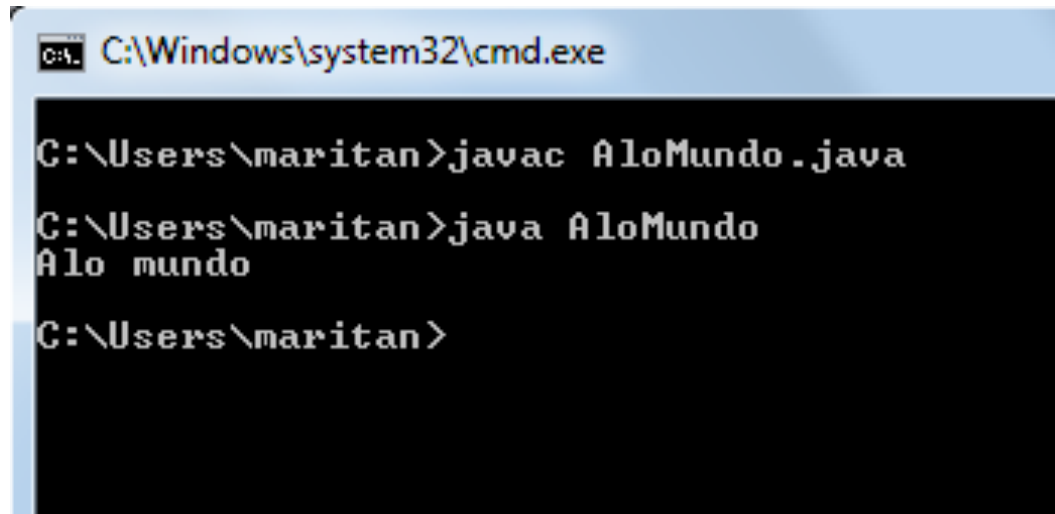

Compilar e Executar usando JDK

1. Compilar o programa (javac)

- ▶ **Comando:** `javac AloMundo.java`

2. Executar programa (java):

- ▶ **Comando:** `java AloMundo`



```
C:\Windows\system32\cmd.exe

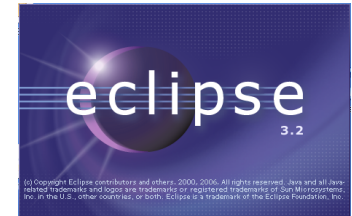
C:\Users\maritan>javac AloMundo.java

C:\Users\maritan>java AloMundo
Alo mundo

C:\Users\maritan>
```

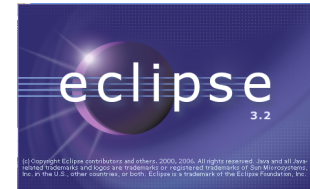
Ambientes de Desenvolvimento (IDEs)

- ▶ Outra alternativa para escrever, compilar e rodar programas Java;
- ▶ Possuem uma versão integrada de :
 - ▶ Editor de Texto;
 - ▶ Compilador e Interpretador Java;
 - ▶ Editor Gráfico;
 - ▶ Debugger;
- ▶ Facilita a construção e manipulação de programas
 - ▶ Rápida visualização de todos os arquivos do projeto;
 - ▶ Ferramentas de gerenciamento de trabalho coletivo, etc



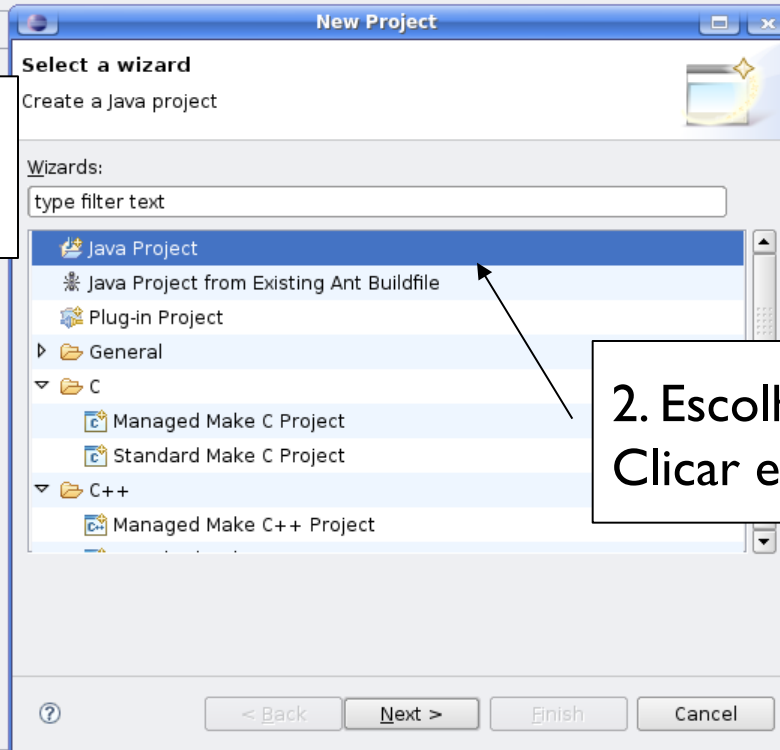
Eclipse

- ▶ No curso, usaremos a IDE Eclipse:
- ▶ Contém as seguintes ferramentas
 - ▶ **Editor de texto**: com destaque de sintaxe
 - ▶ **Compilador e interpretador** Java (e de outras linguagens);
 - ▶ **Editor gráfico**: criação e edição da interface gráfica;
 - ▶ **Debugger**: auxilia na correção de bugs;



Criando o Projeto no Eclipse...

1. Criar um Projeto Java
File → New → Project



2. Escolher Java Project
Clicar em Next

Criando o Projeto no Eclipse...

3. Nomear o Projeto...



4. Finalizar

Criando a Aplicação

**1. Clicar com botão direito...
New → Class**

3. Escrever o nome da classe

4. Finalizar

Java Class

The use of the default package is discouraged.

Source folder: Browse...

Package: Browse...

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ default ☐ private ☐ protected

☐ abstract ☐ final ☐ static

Superclass: Browse...

Buttons: Add... Remove

Stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments as configured in the [properties](#) of the current project?

☐ Generate comments

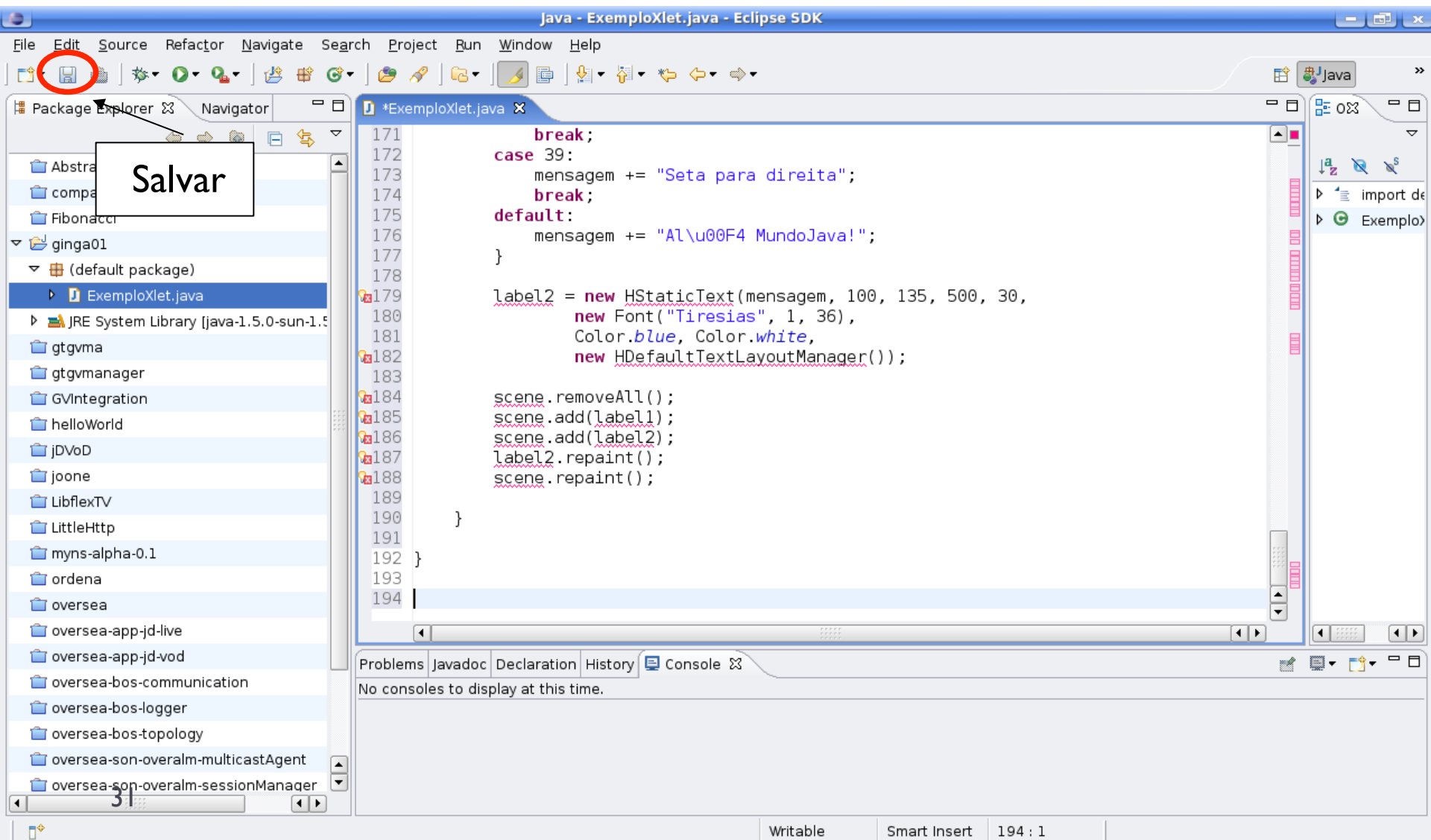
Buttons: Finish Cancel

Package Explorer:

- AbstractFactory
- comparaNumeros
- Fibonacci
- ginga01
- JRE System Library [java-1.5.0-sun-1.5.0]
- gtgvma
- gtqvmanager
- LittleHttp
- myns-alpha-0.1
- ordena
- oversea
- oversea-app-jd-live
- oversea-app-jd-vod
- oversea-bos-communication
- oversea-bos-logger
- oversea-bos-topology
- oversea-son-overalm-multicastAgent
- oversea-son-overalm-sessionManager
- oversea-son-smcp
- PortalStub

ginga01

Salvando...



Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Linguagem de Programação I

Introdução a Java

▶ Tiago Maritan

▶ tiago@ci.ufpb.br

