

# Programação Orientada a Objetos

Prof. Daniel Brandão

## Prof. Daniel Brandão

- Graduado em **Sistemas para Internet**
- Especialista em **Tecnologias Para Aplicações Web**
- Na **Informática** (desde 2004)
- Programador Web (desde 2006)
- Professor em **Informática** (desde 2011)
- Professor **Universitário** (desde 2013)

# Prof. Daniel Brandão

- Graduado em **Sistemas para Internet**
- Especialista em **Tecnologias Para Aplicações Web**
- Na Informática (desde 2004)
- Programador Web (desde 2006)
- Professor em Informática (desde 2011)
- Professor Universitário (desde 2013)
  
- **Contatos:**



DanielBrandao.com.br



Daniel85br



professor@danielbrandao.com.br

# Objetivos

- ▶ Embasamento teórico e Histórico da Linguagem
- ▶ Características da linguagem e ambiente de Programação
- ▶ Etapas da criação à execução de um programa em Java

# Perguntas

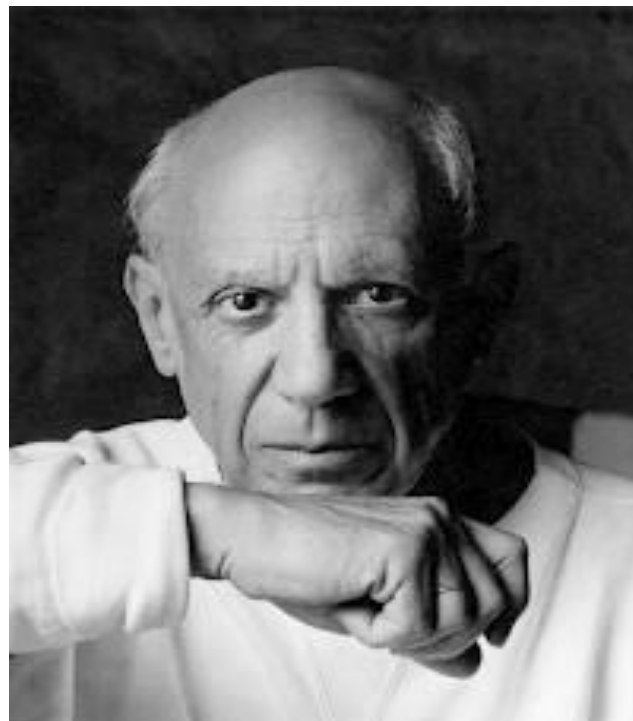
- ▶ Como surgiu a linguagem de programação Java ?
- ▶ Qual a origem do nome Java ?
- ▶ Qual foi seu primeiro nome ?
- ▶ Surgiu com que função ? Para que cenário de atuação ?
- ▶ O que é uma linguagem de alto nível ? Vantagens ?  
Desvantagens ?

# Perguntas

- ▶ Java é interpretada ou compilada ?
- ▶ O que são tradutores ?

“Computadores são inúteis, eles apenas dão respostas”

*Pablo Picasso*



## Mas qual é o problema ?

- ▶ Ponteiros ?
- ▶ Gerenciamento de memória ?
- ▶ Organização ?
- ▶ Falta de bibliotecas ?
- ▶ Reescrever parte do código se mudar o SO ?
- ▶ Custo financeiro de utilizar uma determinada tecnologia ?



# Linguagens de programação

## ▶ Tipos de linguagens de programação:

### ▶ Linguagem de Máquina

```
0010 0100 0000 0010 0000  
0001 0001 1010 0000 0011  
0000 0001
```

### ▶ Linguagem de Baixo Nível

```
- add ecx, ecx  
- sub eax, ecx
```

### ▶ Linguagem de Alto Nível

```
contador := 0;  
a := 3 * a;
```

# Linguagens de máquina

- São linguagens voltadas para a máquina
- São baseadas no código binário utilizado diretamente pelo computador
- As instruções variam de processador para processador
  - **Vantagens:** programas são executados com maior velocidade de processamento e ocupam menos espaço de memória
  - **Desvantagens:** pouca portabilidade (em geral, um programa para um processador não serve para outro)

## Linguagens de baixo nível

- As instruções são simplificações da linguagem de máquina que usam códigos mnemônicos;
- Usualmente recebem a denominação de Assembly;
- São linguagens intermediárias entre a linguagem de máquina e a linguagem de alto nível;
  - **Vantagens:** comandos com sintaxe mais inteligível que nas linguagens de máquina
  - **Desvantagens:** alguns mnemônicos ainda são de difícil compreensão

# Linguagens de Alto Nível

- ▶ Existem programas especiais (compiladores ou interpretadores) para traduzir o código para linguagem de máquina
- Os comandos apresentam um nível mais alto de abstração e próximos da linguagem humana
  - **Vantagens:** Um programa escrito para uma plataforma pode ser recompilado para outra plataforma, desde que exista o compilador adequado para esta outra plataforma.
  - **Desvantagens:** Em virtude da distância que existe entre linguagens de máquina e linguagens de alto nível, os compiladores podem gerar código de máquina que é pouco eficiente e ocupa mais memória do que deveria.

# Tradutores

- São programas conversores de linguagens:
- ✓ **Montadores** - traduzem um programa codificado em linguagem de baixo nível para linguagem de máquina. São chamados de Assembler;
- ✓ **Interpretadores** - transformam instrução por instrução de um programa em linguagem de máquina, cada instrução transformada é executada;
- ✓ **Compiladores** - transformam automaticamente a totalidade de um programa escrito em uma linguagem de alto nível para um programa que possa ser executado pela máquina (programa em linguagem de máquina);

# A Tecnologia Java

- ▶ O nome “Java” é utilizado para referir-se:
  - ▶ Uma linguagem de programação de alto nível e orientada a objetos
  - ▶ Ao ambiente que oferece suporte ao desenvolvimento e execução de aplicações escritas nessa linguagem
  - ▶ Uma biblioteca de classes que podem ser reaproveitadas durante desenvolvimento de aplicações
- ▶ Java foi anunciada pela Sun Microsystems em 1995
- ▶ Hoje é proprietária da Oracle Inc. desde 2009

# A Tecnologia Java - Histórico

- ▶ 1991 - Grupo de funcionários da Sun (**James Gosling**, Patrick Naughton e Mike Sheridan) criam o Projeto Green:
  - ▶ Projeto Green - Criação de uma plataforma de software para aparelhos eletrônicos
  - ▶ Desenvolvimento do Sistema Operacional GreenOS
  - ▶ Criação da linguagem de programação Oak para operar sob o GreenOS
  - ▶ O projeto não obteve sucesso! O mercado para aparelhos eletrônicos não cresceu tanto quanto a Sun esperava.

## A Tecnologia Java - Histórico

- ▶ 1993 - Direcionamento de aplicações Oak para a Internet, com a intenção de substituir as aplicações em CGI (*Common Gateway Interface*).
  - ▶ A CGI permite que um navegador execute programas residentes no servidor web, que podem processar dados recebidos de formulários de páginas HTML
- ▶ 1994 - Oak é rebatizado como Java (cidade de origem de um tipo de café).
- ▶ 1995 - A Sun anuncia o lançamento da tecnologia.



# Características da linguagem Java

## ► Compilada:

- Um programa em Java é compilado para o chamado bytecode, que é próximo as instruções de máquina, mas não de uma máquina real.
- O bytecode é um código de uma máquina virtual idealizada pelos criadores da linguagem, sendo interpretado e executado pela JVM (Máquina Virtual Java).
- Java pode ser mais rápida do que se fosse apenas interpretada. Um compilador just-in-time(JIT) pode produzir código de máquina para os bytecode enquanto eles são interpretados, isso elimina a necessidade de reinterpretá-los futuramente.

# Características da linguagem Java

- ▶ Portável (Independente de Plataforma):
  - ▶ O bytecode gerado pelo compilador pode ser transportado entre plataformas distintas que suportam Java (Solaris, Windows-NT, Windows-XP, Mac/Os, etc).
  - ▶ Esta portabilidade é importante para a criação de aplicações para a Internet.
  - ▶ O próprio compilador Java é escrito em Java, de modo que ele é portável para qualquer sistema que possua o interpretador de bytecodes.

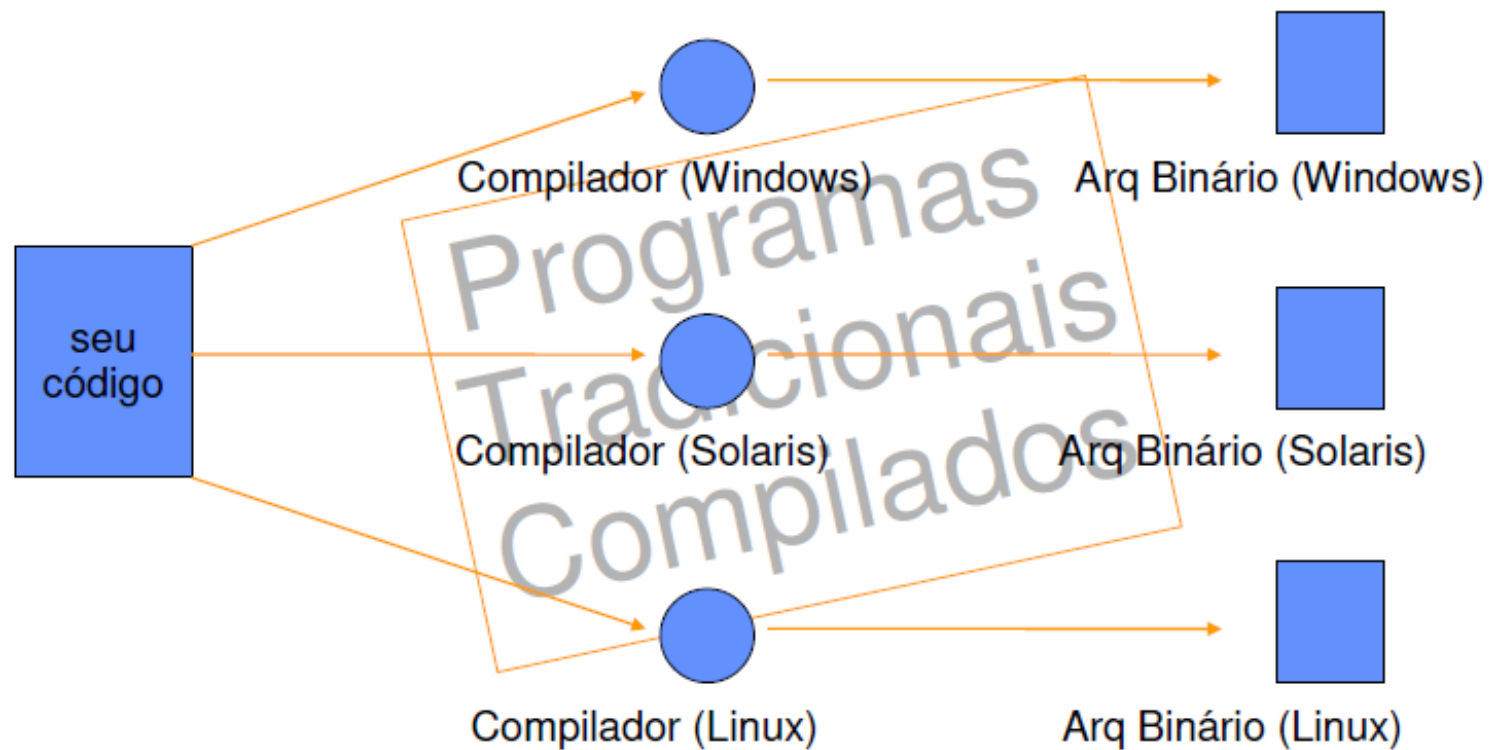
# Características da linguagem Java

## ► Programas Tradicionais Compilados



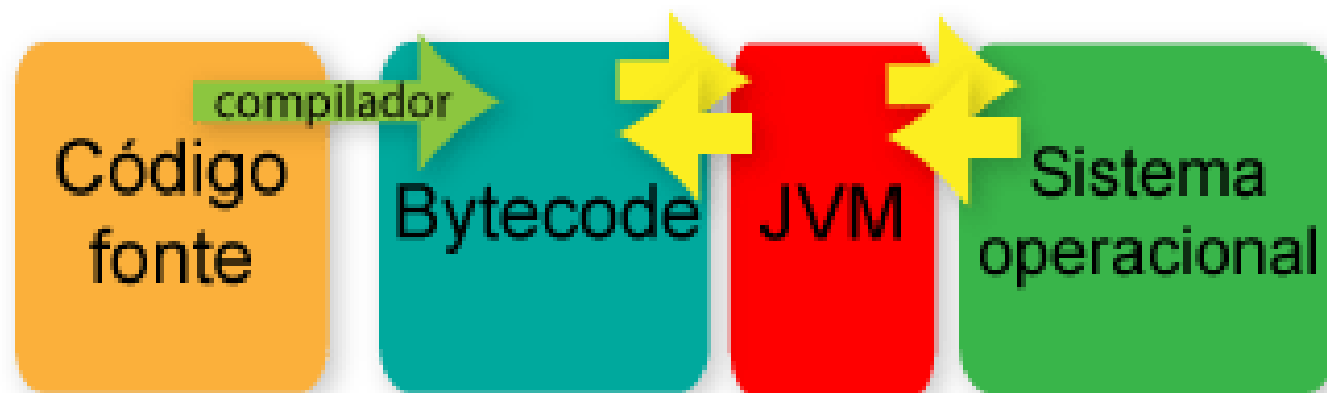
# Características da linguagem Java

## Programas Tradicionais Compilados



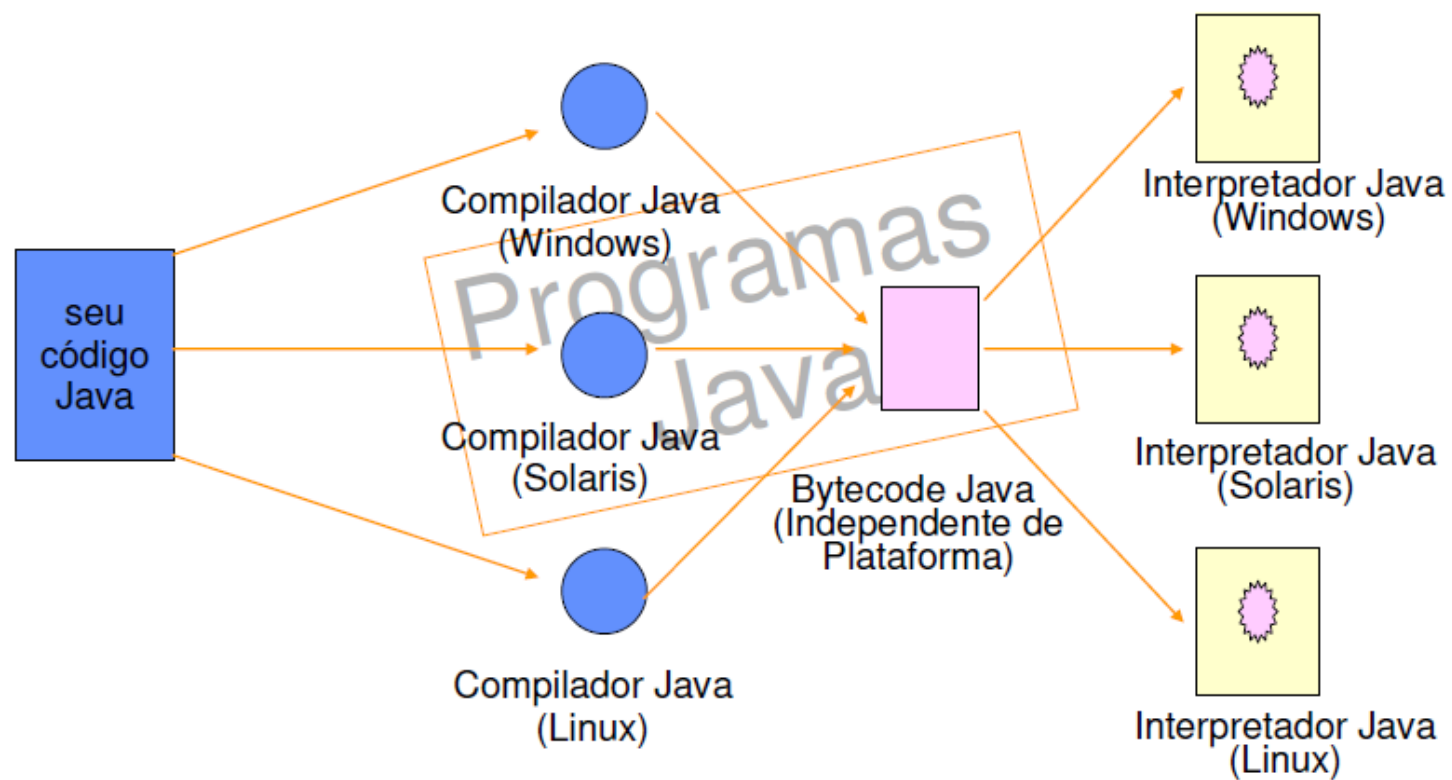
# Características da linguagem Java

## ► Programas Java



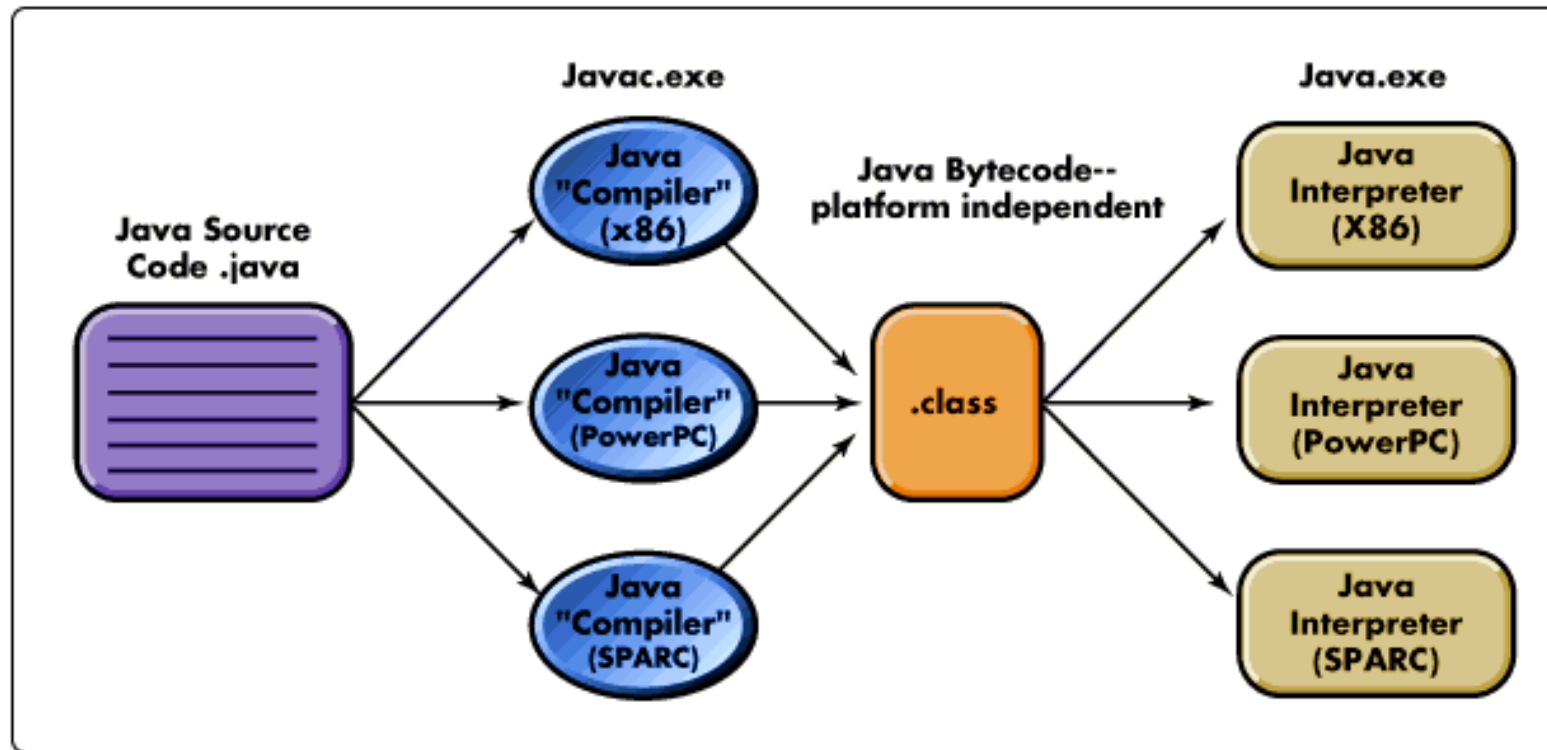
# Características da linguagem Java

## ► Programas Java



# Características da linguagem Java

## ► Programas Java



# Características da linguagem Java

- ▶ **Vantagens de Sistemas Multiplataforma:**
  - ▶ Não há necessidade de mudanças nos programas para funcionamento em diferentes plataformas
  - ▶ Apenas uma versão do programa é suficiente para distribuição multiplataforma.
    - ▶ Software para Internet



## Características da linguagem Java

- ▶ **Desvantagens de Sistemas Multiplataforma:**
  - ▶ Muitas vezes é necessário renunciar à algumas instruções particulares de um ambiente em específico
  - ▶ Interpretação pode ter desempenho inferior do que compilação
    - ▶ Em particular, a plataforma Java evoluiu bastante

# Características da linguagem Java

## ▶ Orientada a Objetos:

- ▶ Objetos e Classes
- ▶ Encapsulamento de dados e operações
- ▶ Polimorfismo
- ▶ Reutilização de código (herança)
- ▶ Facilidade de extensão e manutenção de código
- ▶ Provê extensa biblioteca de classes (API)
  - ▶ Applets (java.applet)
  - ▶ Definição de interfaces gráficas (java.awt)
  - ▶ Suporte a objetos distribuídos (java.rmi)

# Características da linguagem Java

## ▶ Segurança:

- ▶ A máquina virtual oferece um ambiente de execução seguro.
- ▶ Os *bytecodes* são verificados antes de serem executados.
- ▶ A coleta automática de lixo evita erros que os programadores cometem quando são obrigados a gerenciar diretamente a memória

# Características da linguagem Java

## ► Robustez:

- A presença de mecanismos de tratamento de exceções torna as aplicações mais robustas, permitindo que elas não sejam finalizadas indevidamente, mesmo quando rodando sob condições adversas.

# Características da linguagem Java

## ▶ Coleta de Lixo:

- ▶ Memória alocada em Java não precisa ser liberada pelo programador.
- ▶ Quando um objeto não tem mais referências apontando para ele, seus dados não podem mais ser usados e, dessa forma, a memória deve ser liberada.
- ▶ Sua função é gerenciar a utilização de memória, procurando áreas de memória que não estejam mais em uso para realizar a liberação das mesmas.

# Características da linguagem Java

## ▶ Eficiência:

- ▶ Java foi criada para ser utilizada em computadores com pouca capacidade de processamento.
- ▶ Java é mais eficiente que grande parte das linguagens de *scripting* existentes.
- ▶ Um compilador just-in-time(JIT).

## Ambiente de Programação

- ▶ **Java Standard Edition (JSE):** ferramentas e API's essenciais para qualquer aplicação Java, possibilita o desenvolvimento de aplicações
- ▶ **Java Enterprise Edition (JEE):** ferramentas e API's para o desenvolvimento de aplicações Web e distribuídas.
- ▶ **Java Micro Edition (JME):** ferramentas e API's para o desenvolvimento de aplicações para equipamentos eletrônicos (celulares, TV Interativa, etc.)

## Ambiente de Programação

- ▶ Ambiente Java Sun para Windows:
  - ▶ Java SE *Development* Kit (JDK)
  - ▶ Endereço: <http://Oreacle.com/javase>
  - ▶ A instalação é simples e intuitiva
  - ▶ Java Runtime Enviroment (JRE)
    - ▶ Tudo que é necessário para executar aplicações Java;
    - ▶ Este ambiente é um subconjunto do Java SE, vem sem o compilador e sem as ferramentas para desenvolvimento;



## Ambiente de Programação

<b>Sigla</b>	<b>Significado</b>
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine

## Ambiente de Programação

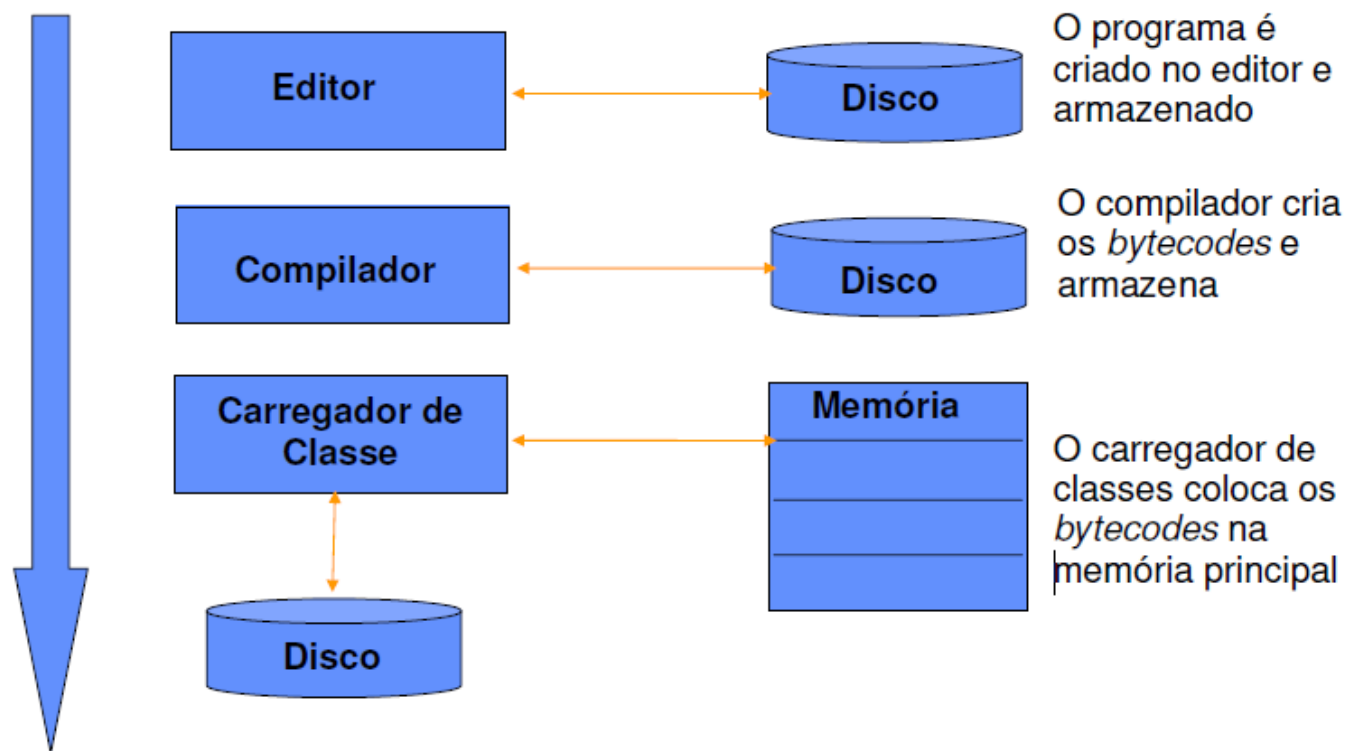
- ▶ Compilador Java: javac
  - ▶ Compila arquivos fonte Java com extensão .java
  - ▶ Gera arquivos de bytecodes com extensão .class
- ▶ Interpretador Java (Máquina Virtual): java[JVM]
  - ▶ Interpreta arquivos de bytecodes com extensão .class
- ▶ Documentador Java: javadoc
  - ▶ Aplicativo para geração automática de documentação em html.

## Ambiente de Programação

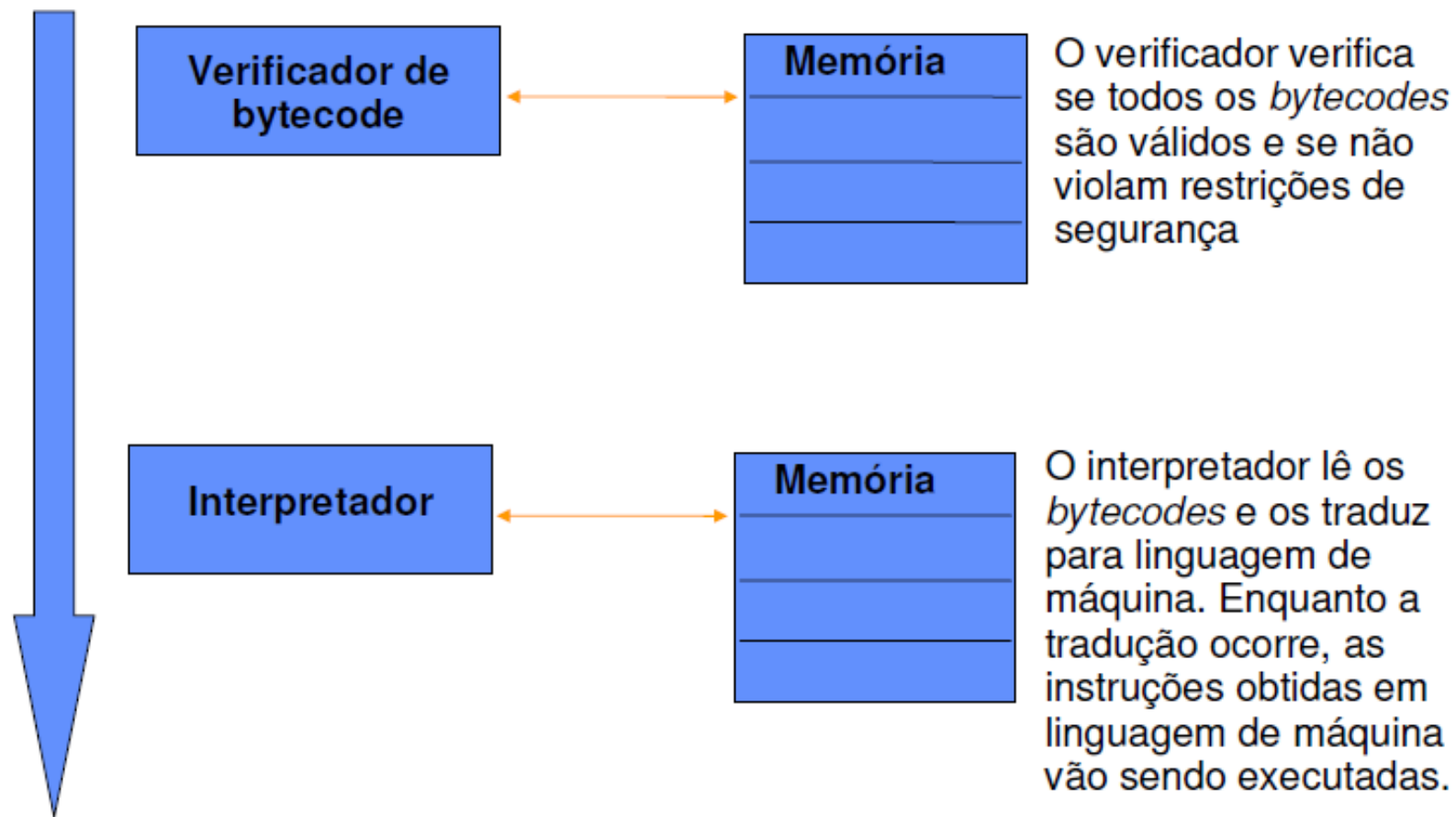
### ▶ Arquivos .jar:

- ▶ JAR é a sigla para Java ARchive, sendo o formato criado pela SUN, baseado no mesmo formato de compactação de arquivos ZIP, - Utilizado para distribuir aplicações, ou bibliotecas, através de um só arquivo.
- ▶ Para a criação de arquivos JAR, existe a ferramenta JAR, incluída no JSDK.
- ▶ Entre as vantagens: facilidade de transmissão via web, com possibilidade de assinatura digital, controle de versão do componente, além da independência de plataforma.

# Ambiente Java Típico



# Ambiente Java Típico

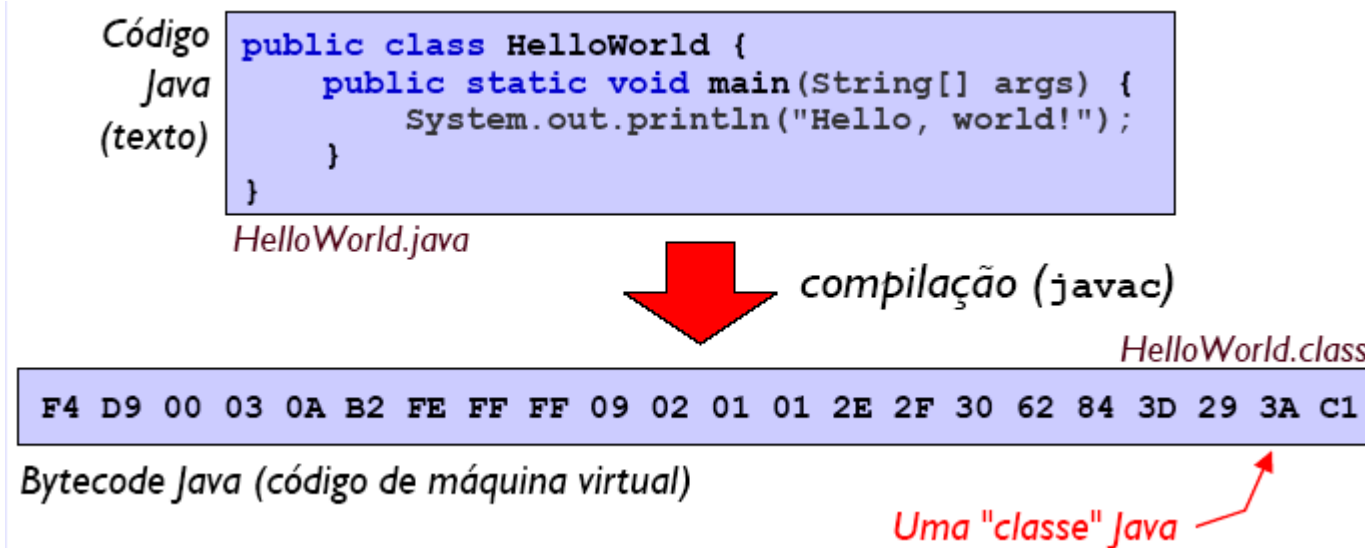


## Etapas - Compilação e Execução

- ▶ Edição - A aplicação é criada em um editor e armazenada no disco (extensão *.java*).
  - ▶ Cada programa é considerado uma classe
  - ▶ Nome da classe no programa tem que ser igual ao nome do arquivo físico *.java*
  - ▶ A linguagem é “*case-sensitive*”, *letras maiúsculas* diferentes de minúsculas.
- ▶ Compilação - O compilador cria *bytecodes* e os armazena em disco (extensão *.class*).

## Etapas - Compilação e Execução

- *Bytecodes: código que roda em qualquer máquina através da Máquina Virtual Java.*



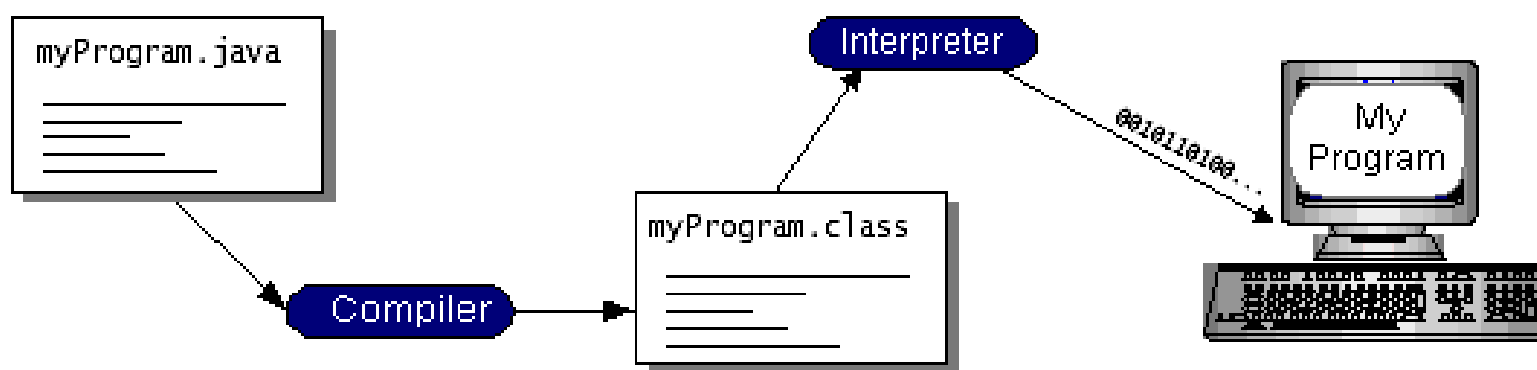
## Etapas - Compilação e Execução

- ▶ Carga - O carregador de classes (*classloader*)
  - ▶ carrega os *bytecodes* na memória principal.
- ▶ Verificação
  - ▶ O verificador de *bytecodes* verifica se os mesmos são válidos e não violam as restrições de segurança de Java.
- ▶ Execução
  - ▶ O interpretador lê os *bytecodes* e os traduz para uma linguagem na qual o computador possa entender.



## Etapas - Compilação e Execução

- ▶ Programas Java são compilados e interpretados
- ▶ javac - compilador
- ▶ java - interpretador

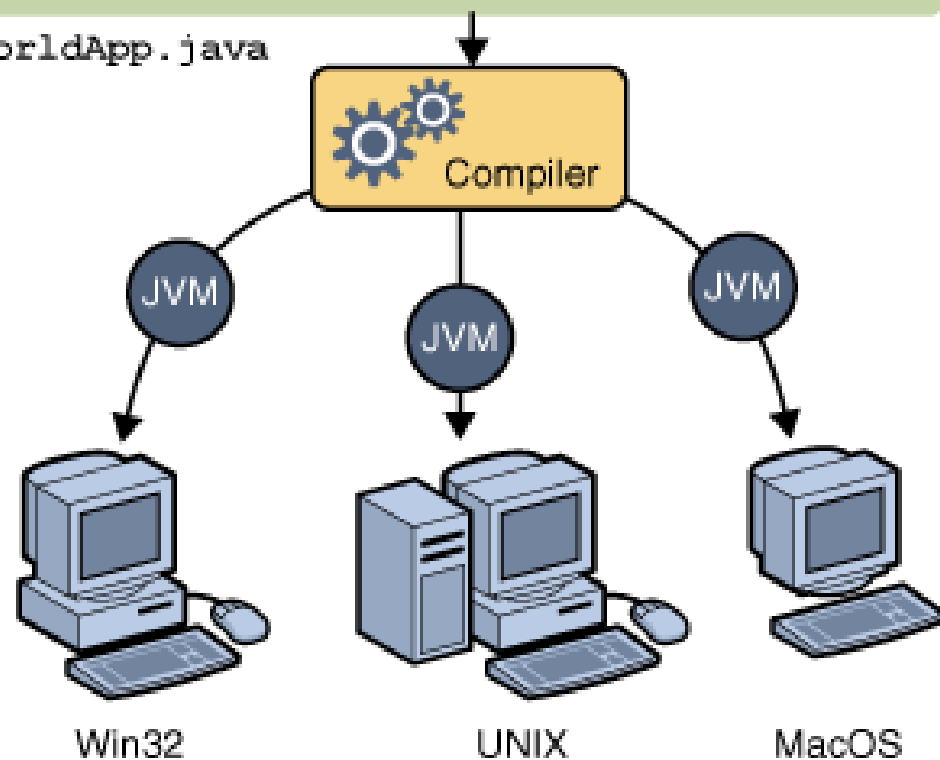


# Etapas - Compilação e Execução

Java Program

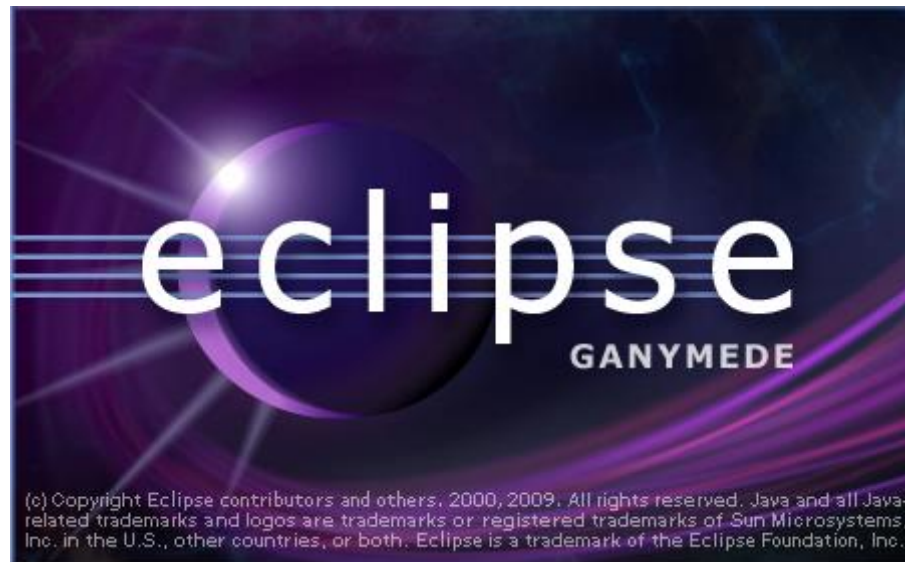
```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java

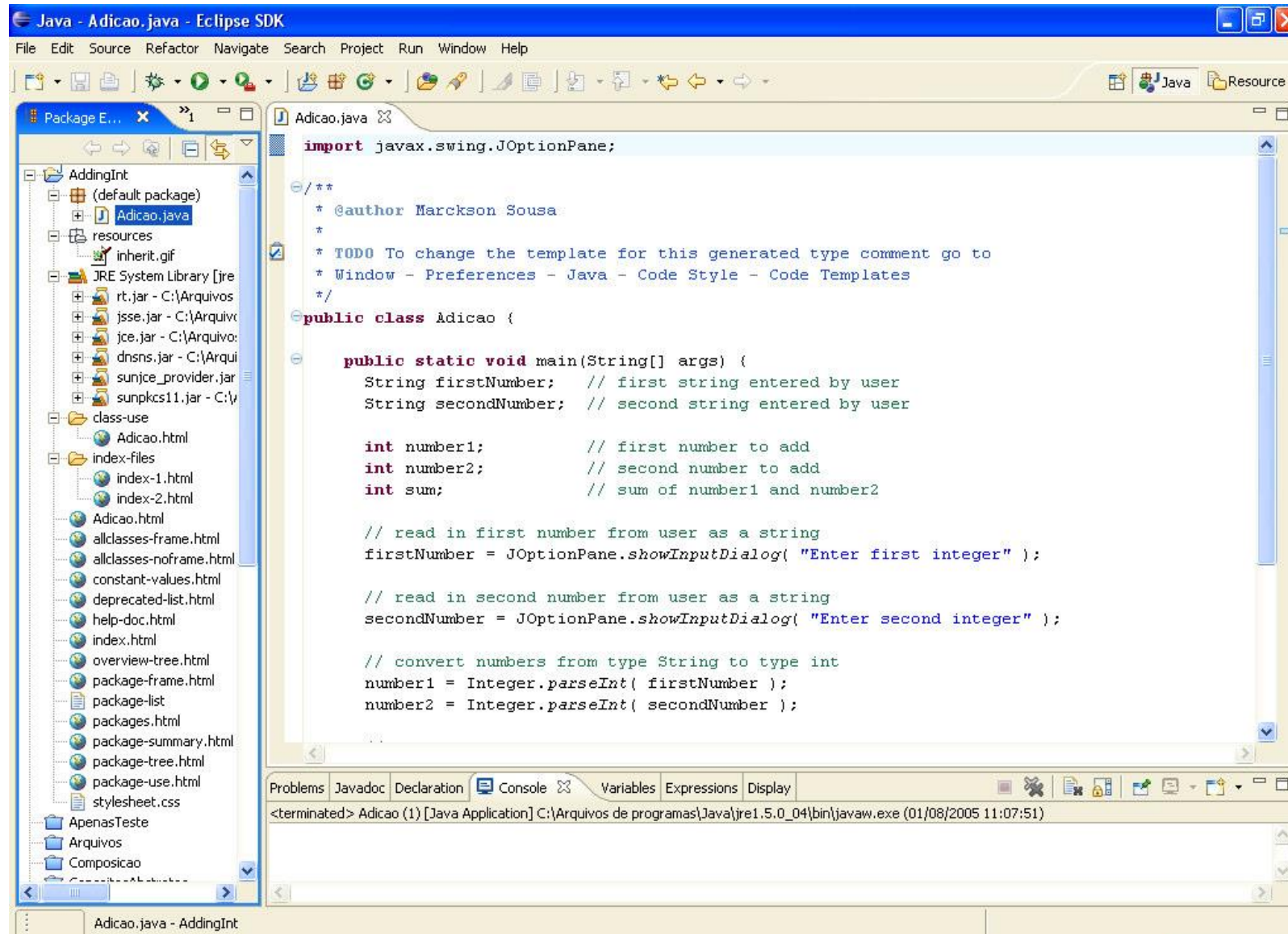


## Ambiente de Desenvolvimento

- ▶ **Eclipse:**
- ▶ Apoio de gigantes como IBM, HP, Oracle, Borland, Rational, etc. (<http://www.eclipse.org>)
- ▶ Facilidades de utilização do ambiente
- ▶ Plug-in para acréscimo de funcionalidades
- ▶ Facilita a criação e refatoramento de código Java

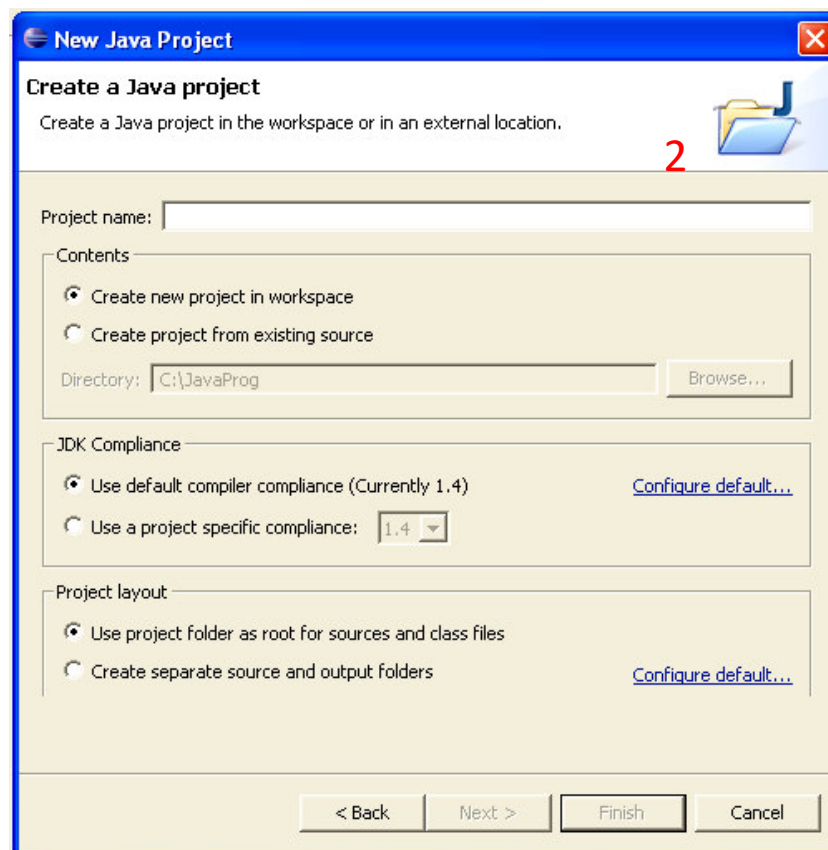
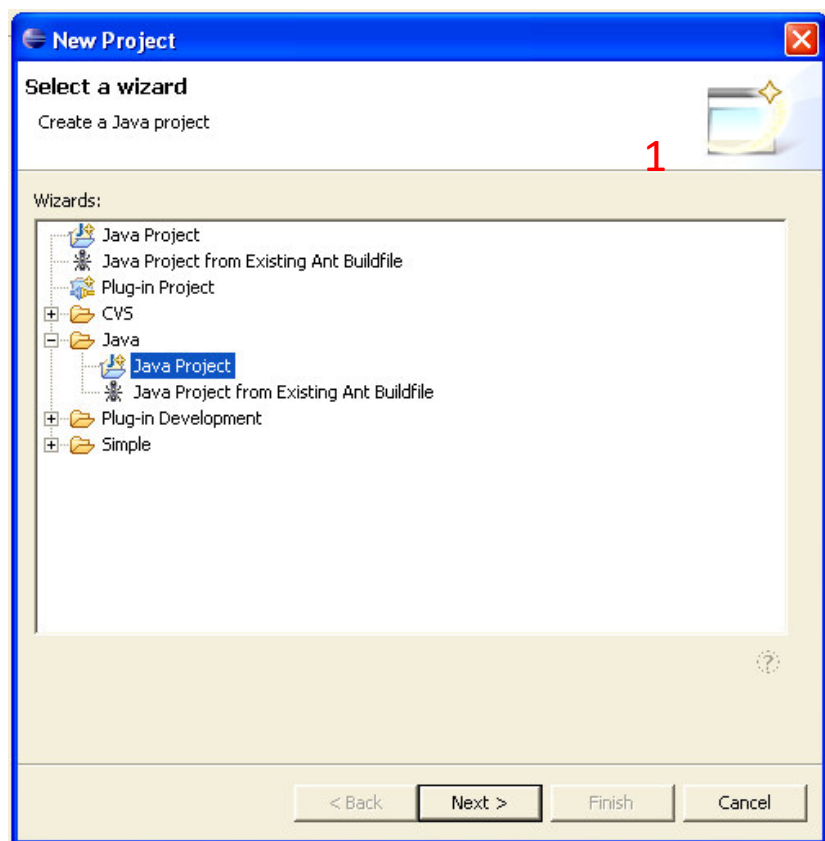


# Ambiente de Desenvolvimento



# Ambiente de Desenvolvimento

- Criação de um novo Projeto:  
File → New → Project



## Ambiente de Desenvolvimento

Onde é solicitado o nome do projeto digite:

- *Primeiro Programa*

## Etapas - Compilação e Execução

- ▶ Para criar uma Classe:
  - ▶ Clique com o botão direito sobre o projeto criado
  - ▶ Selecione a opção: New Class
    - ▶ Onde é solicitado o nome da classe digite:
      - ▶ *HelloWorld*

**New Java Class**

Create a new Java class.

Source folder: Primeiro Programa Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: |

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method 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

Finish Cancel

## Etapas - Compilação e Execução

- ▶ Deixe o código de sua classe conforme é mostrado a seguir:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         String nome = "Daniel";  
4         System.out.println("Hello, World! " + nome);  
5     }  
6 }
```



## Etapas - Compilação e Execução

- ▶ Para executar clique com o botão direito sobre o código da classe e selecione a opção:

**Run As → Java Application**

- ▶ Para gerar a documentação html de seu programa selecione no menu:

**Project → Generate Javadoc**

- ▶ Selecione o seu projeto e escolha o local onde a documentação será salva.
- ▶ Clique em finish.

## Explicando o Programa

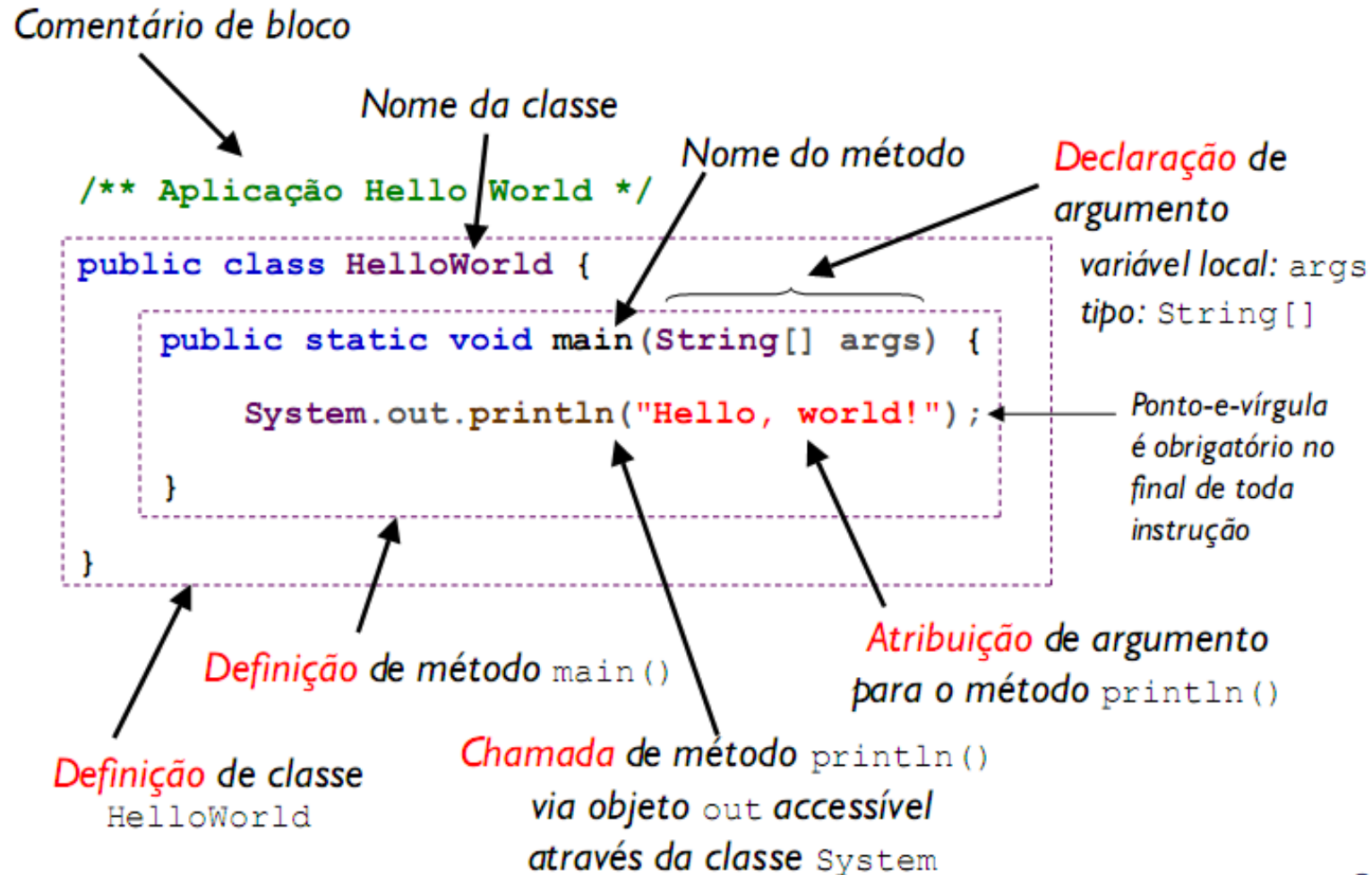
- ▶ Explicando alguns termos:
  - ▶ **import** `javax.swing.JOptionPane` - importa uma biblioteca da linguagem Java que permite criar caixas de diálogo.
- ▶ **public** é um especificador de visibilidade, indicando que este método é acessível externamente a esta classe.
- ▶ **class** é uma palavra reservada que marca o início da declaração de uma classe.
- ▶ **BemVindo** é o nome dado a classe.
- ▶ O "abre chaves" marca o início das declarações da classe que são os atributos e métodos.

## Explicando o Programa

- ▶ **static** é um outro especificador de visibilidade, indicando que o método deve ser compartilhado pelos objetos que serão criados a partir desta classe.
- ▶ **void** é uma palavra reservada que indica que o método *main não retorna nenhum valor*
- ▶ **main** corresponde a um método especial que indica para a máquina virtual Java o início da execução do programa.

# Explicando o Programa

## ► A Estrutura da classe: O método main



# Erros

- ▶ Durante o desenvolvimento, erros podem ocorrer em dois domínios: tempo de compilação e tempo de execução.
- ▶ É essencial aprender a identificar a causa do erro
  - ▶ LEIA a mensagem e identifique a linha onde ocorreu o erro
  - ▶ Corrija os erros na ordem em que eles parecem
  - ▶ Sempre recompile depois de corrigir os erros

## Exercício 1

- ▶ Abra o console de linha de comando e digite, compile e execute o primeiro programa em Java (Olá Mundo!)

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

## Exercício 2

- ▶ Abra o Eclipse e digite, compile e execute o segundo programa em Java (“Operações Matemática”)

```
int numero1 = <aqui você digita um número>;
```

```
int numero2 = <aqui você digita um outro número>;
```

```
// Crie variáveis para as quatro operações básicas matemáticas e  
imprima o resultado. Ex:
```

```
Int soma = numero1 + numero2;
```

```
Int subtrai = numero1 – numero2;
```

```
Int multiplica = numero1 * numero2;
```

```
Int divide = numero1 / numero2;
```

```
...
```

The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side of the image, creating a modern, dynamic feel. The rest of the background is a solid, very light blue-grey color.

# Dúvidas?