



Linguagem de Programação

JAVA

Aspectos Fundamentais

Profa. Joyce Miranda

Introdução



▶ Tópicos

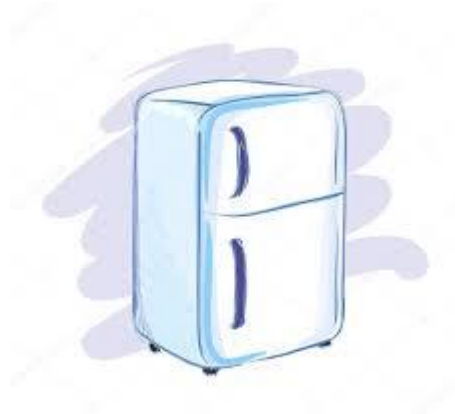
- ▶ Responder o que é Java;
- ▶ Principais características da linguagem;
- ▶ Compilar e executar um programa simples.

Introdução

► Breve Histórico

► Ideia Original

- **1992 - Sun Microsystems (Atual Oracle) - James Gosling**
- Interpretador multiplataforma para sistemas embarcados.
 - Facilitar a reescrita de software para diferentes dispositivos.
- Tentaram fechar contrato com grandes fabricantes de eletrônicos, mas não houve êxito.



Introdução

► Breve Histórico

► Grande Ideia

- Rodar pequenas aplicações dentro do navegador.
 - Diversidade grande de sistemas operacionais e navegadores



- Programar em uma única linguagem independente de plataforma.



▶ Características

▶ Multiplataforma

- ▶ Desktop, Web, *Mobile*, TV digital, Cartões inteligentes

▶ Linguagem segura e confiável

- ▶ Não trabalha com ponteiros
- ▶ Coleta automática de lixo
- ▶ Mecanismos para tratamento de exceções

▶ Orientada a objetos

- ▶ Encapsulamento
- ▶ Herança
- ▶ Polimorfismo



► Apresentação da Plataforma

SIGLA	SIGNIFICADO	UTILIZAÇÃO
JSE	<i>Java Standard Edition</i>	É a base da plataforma. Inclui o ambiente de execução e as bibliotecas comuns.
JEE	<i>Java Enterprise Edition</i>	Desenvolvimento de aplicações corporativas que incluem serviços de redes e de web.
JME	<i>Java Micro Edition</i>	Desenvolvimento de aplicações para dispositivos móveis e embarcados.



Introdução

▶ JSE – Java Standard Edition

- ▶ Ambiente completo para o desenvolvimento de aplicações desktop e para servidores.
- ▶ Base da plataforma JAVA.
- ▶ Pode ser obtido das seguintes formas:

JDK

*JAVA
Development Kit*

JRE

*Java Runtime
Environment*



Introdução

- ▶ **JSE – Java Standard Edition**
 - ▶ Compilação
 - ▶ Debugging
 - ▶ Geração de documentação (javadoc)
 - ▶ Empacotador de componentes (jar)
 - ▶ JRE, que contém a JVM (Java Virtual Machine).



► Resumo de definições

SIGLA	DEFINIÇÃO
JRE = Java Runtime Environment	Ambiente de execução Java, formado pela JVM e bibliotecas, tudo que você precisa para executar uma aplicação Java.
JDK = Java Development Kit	(Nós) Desenvolvedores, faremos o download do JDK do Java SE (Standard Edition).
JVM = Java Virtual Machine	Identifica o interpretador, esse download não existe.

Introdução

- ▶ **Criação de Programas em JAVA**
 - ▶ 1º Etapa – Definição do Código Fonte



Introdução

► Criação de Programas em JAVA

► 2º Etapa – Compilação

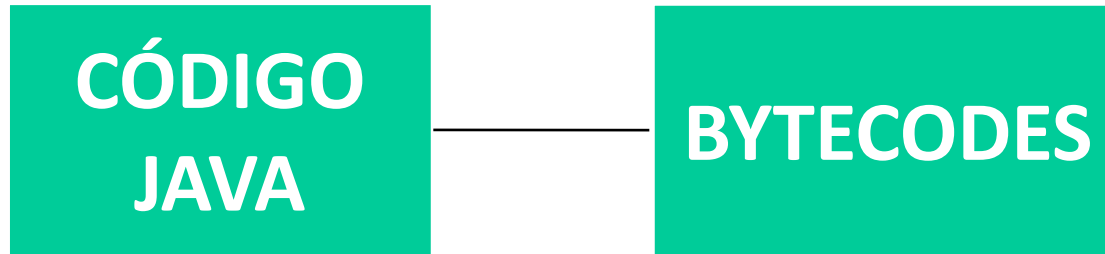


► Compilador: **javac**

Introdução

► Criação de Programas em JAVA

► 2º Etapa – Compilação



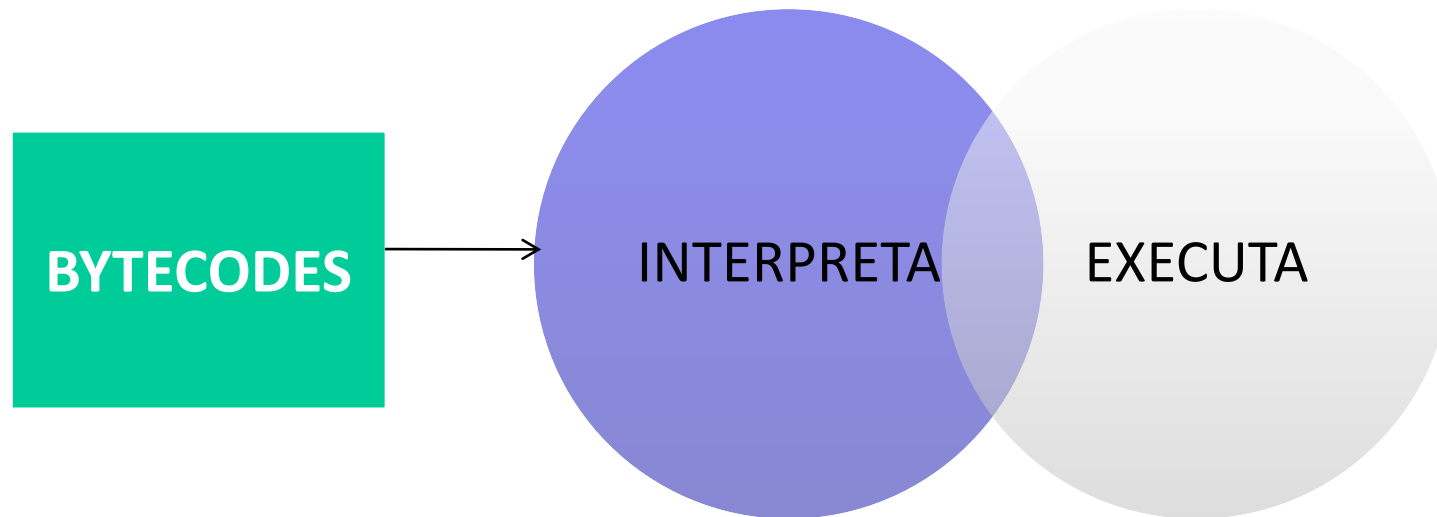
- *São os bytecodes que são independentes de plataforma.*

Introdução

► Criação de Programas em JAVA

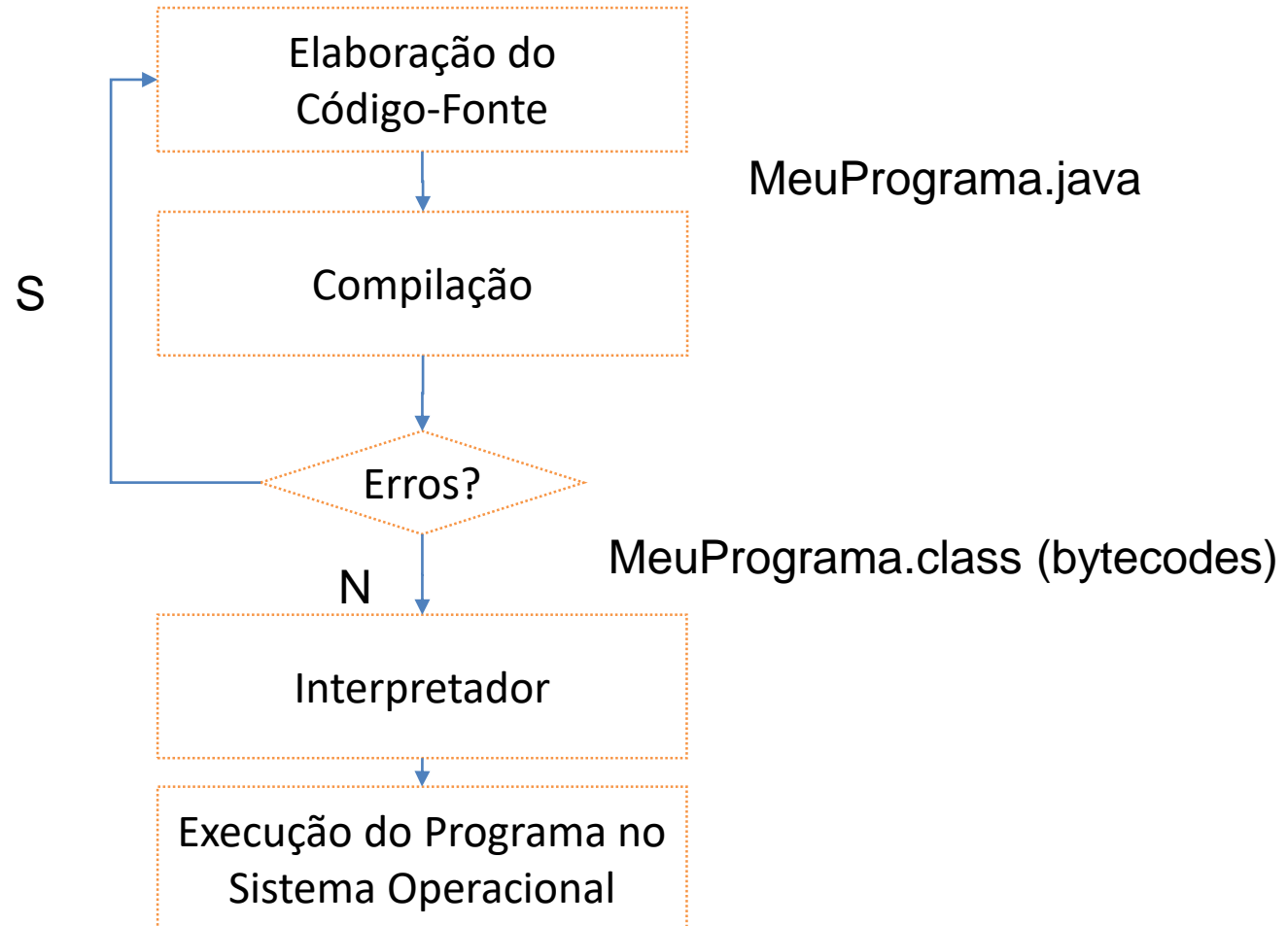
► 3º Etapa - Execução

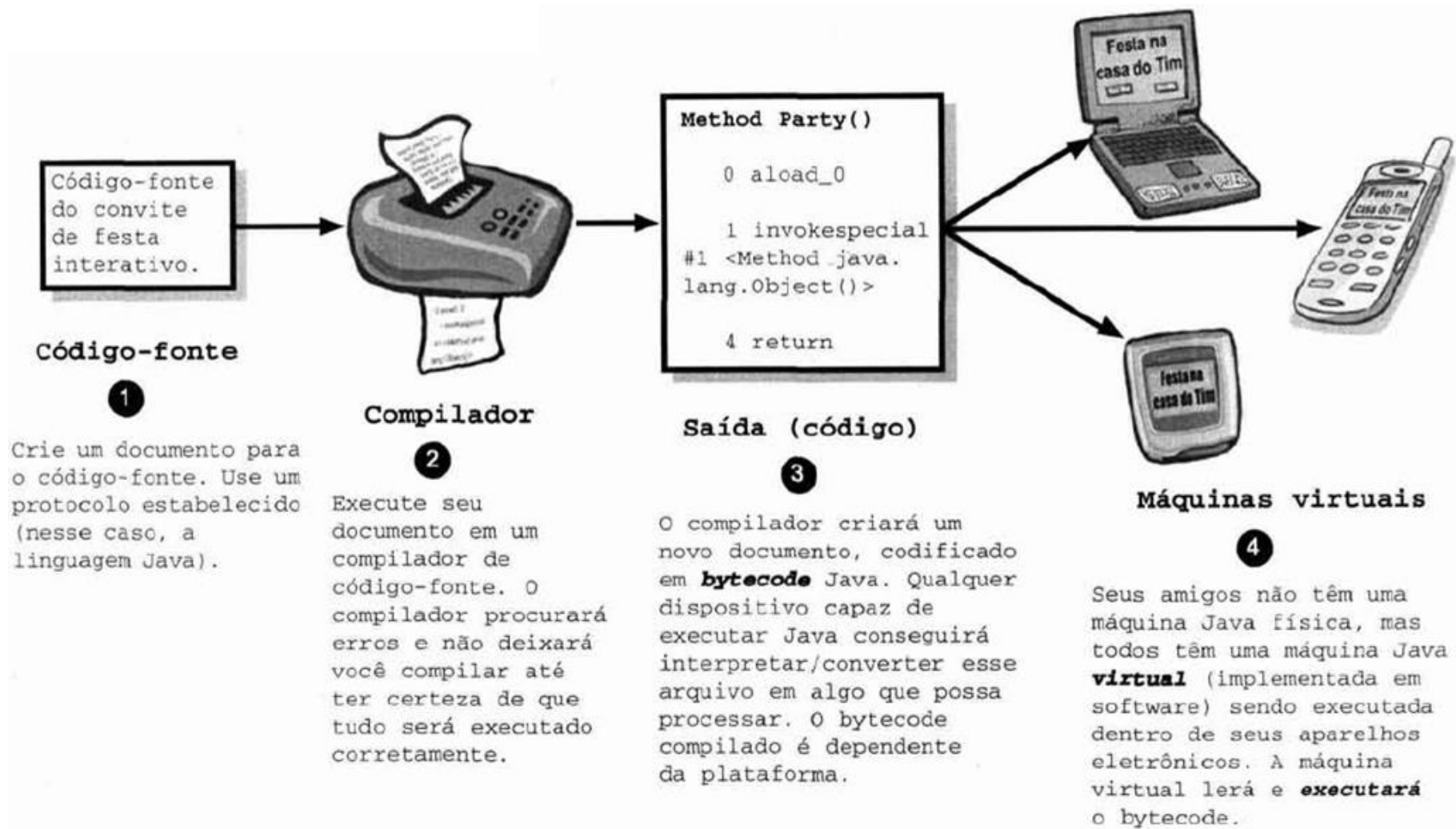
► Interpretador – (JVM)





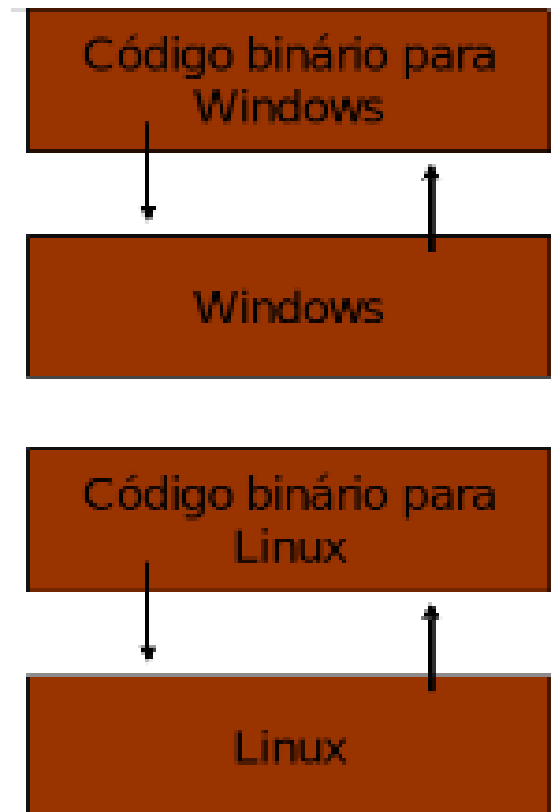
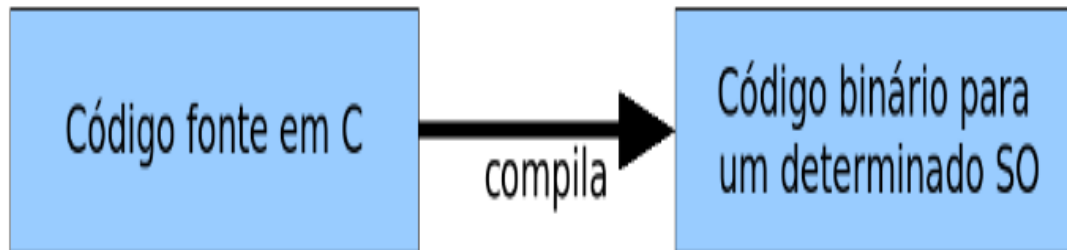
► Criação de Programas em JAVA





Introdução

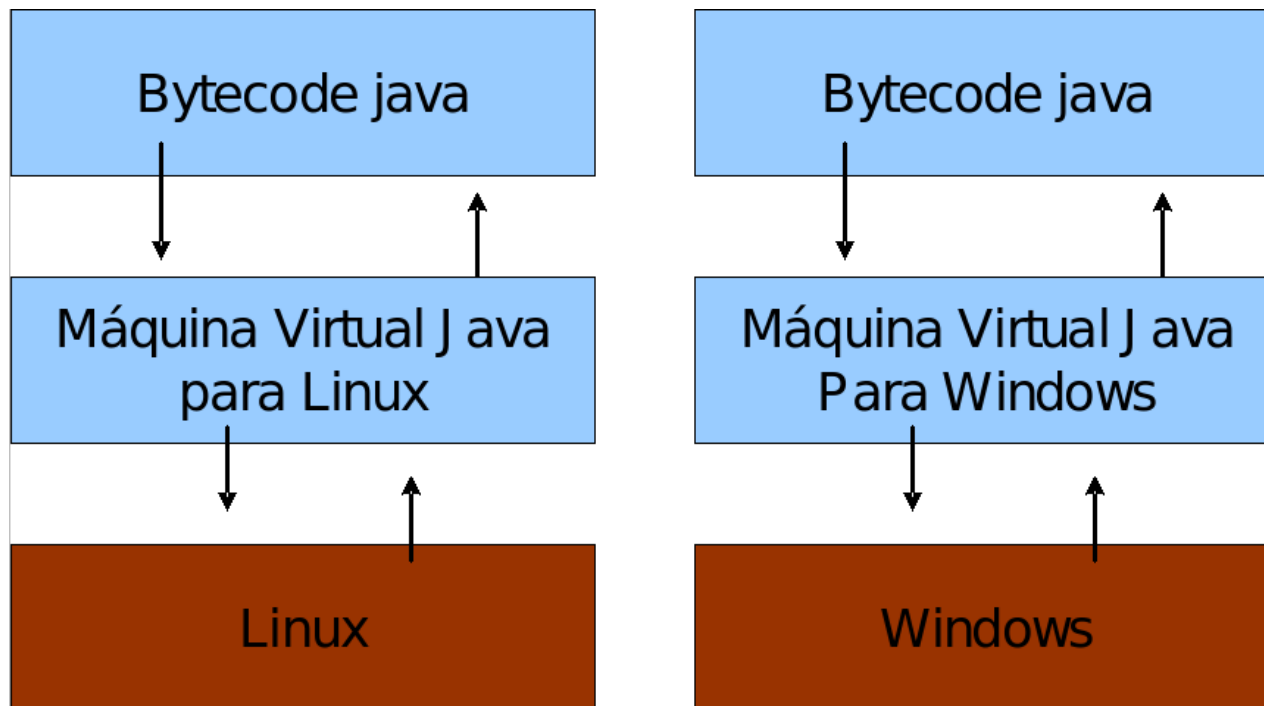
- ▶ Outras Linguagens Compiladas
 - ▶ O código resultante será executado pelo SO.



Introdução

► JAVA - Máquina Virtual

Write once, run anywhere..





```
import java.awt.*;
import java.awt.event.*;
class Party {
    public void buildInvite() {
        Frame f = new Frame();
        Label l = new
Label("Party at Tim's");
        Button b = new
Button("You bet");
        Button c = new
Button("Shoot me");
        Panel p = new Panel();
        p.add(l);
    } // mais código aqui...
}
```

Código-fonte

1

Digite seu código-fonte.
Salve como: **Party.java**

```
File Edit Window Help Plead
%javac Party.java
```

Compilador

2

Compile o arquivo **Party.java** executando o `javac` (o aplicativo do compilador). Se não houver erros, você terá um segundo documento chamado **Party.class**. O arquivo `Party.class` gerado pelo compilador é composto de **bytecodes**.

```
Method Party()
0 aload_0
1 invokespecial #1 <Method
java.lang.Object()>
4 return
Method void buildInvite()
0 new #2 <Class java.awt.Frame>
3 dup
4 invokespecial #3 <Method
java.awt.Frame()>
```

Saída (código)

3

Código compilado:
Party.class



Máquinas virtuais

4

Execute o programa iniciando a Java Virtual Machine (JVM) com o arquivo **Party.class**. A JVM converterá o **bytecode** em algo que a plataforma subjacente entenda e executará seu programa.



Introdução

- ▶ Vamos para o nosso primeiro código!

```
1 class MeuPrograma {  
2     public static void main(String[] args) {  
3         System.out.println("Minha primeira aplicação Java!!");  
4     }  
5 }
```

- ▶ Salve-o como **MeuPrograma.java** em algum diretório.
- ▶ **Obs: O nome definido na classe deve ser o mesmo nome definido no arquivo que será salvo.**



Introdução

- ▶ Vamos para o nosso primeiro código!
 - ▶ Compilando
 - ▶ Verifique o caminho de instalação do Java na sua máquina.

```
"C:\Program Files\Java\jdk1.8.0_25\bin\javac" MeuPrograma.java
```

- ▶ Caso não sejam apresentados erros, execute.

```
java MeuPrograma
```



Introdução

▶ Exercícios

- ▶ Modificando MeuPrograma

- ▶ **1) Altere seu programa para imprimir uma mensagem diferente.**

- ▶ **2) Altere seu programa para imprimir duas mensagens distintas.**

- ▶ **3) Sabendo que os caracteres `\n` representam uma quebra de linha, imprima duas linhas de texto usando uma única linha de código ***System.out.println*****

- ▶ **4) Altere o nome da classe no seu arquivo, compile e execute.**