



# LINGUAGEM DE PROGRAMAÇÃO JAVA

Cefet – Maracanã -RJ

BCC/TSI

Prof. Gustavo Guedes

E-mail: [gustavo.guedes@cefet-rj.br](mailto:gustavo.guedes@cefet-rj.br)

# O QUE É UM PROGRAMA JAVA?

---

- Programas Java consistem em partes chamadas **classes**.
- As classes incluem partes chamadas métodos que realizam tarefas e retornam informações ao concluir.
- A maioria dos programadores Java tira proveito das ricas coleções de classes existentes nas bibliotecas de classe Java, que também são conhecidas como APIs do Java.



# DOIS ASPECTOS

---

- Na realidade, existem dois aspectos para aprender no “mundo” do Java:
  - O primeiro é a própria linguagem Java, de modo que você possa programar suas próprias classes.
- O segundo diz respeito ao vasto número de classes nas extensas bibliotecas de classes Java.



# VISÃO GERAL

---

- No lançamento para o público como a versão 1.02, o Java seduziu os programadores com sua sintaxe amigável, recursos orientados a objetos, gerenciamento de memória e, o melhor de tudo, a promessa de portabilidade.
- A possibilidade de escrever uma vez/executar em qualquer local exerce uma atração muito forte.
- Seguidores devotados surgiram, enquanto os programadores combatiam os erros, limitações e, o fato de ela ser muito lenta.



# VISÃO GERAL

---

- Mas isso foi há muito tempo. Hoje em dia, você pode manipular o mais fácil, rápido e mais poderoso Java. Iremos estudar a versão 5.0.



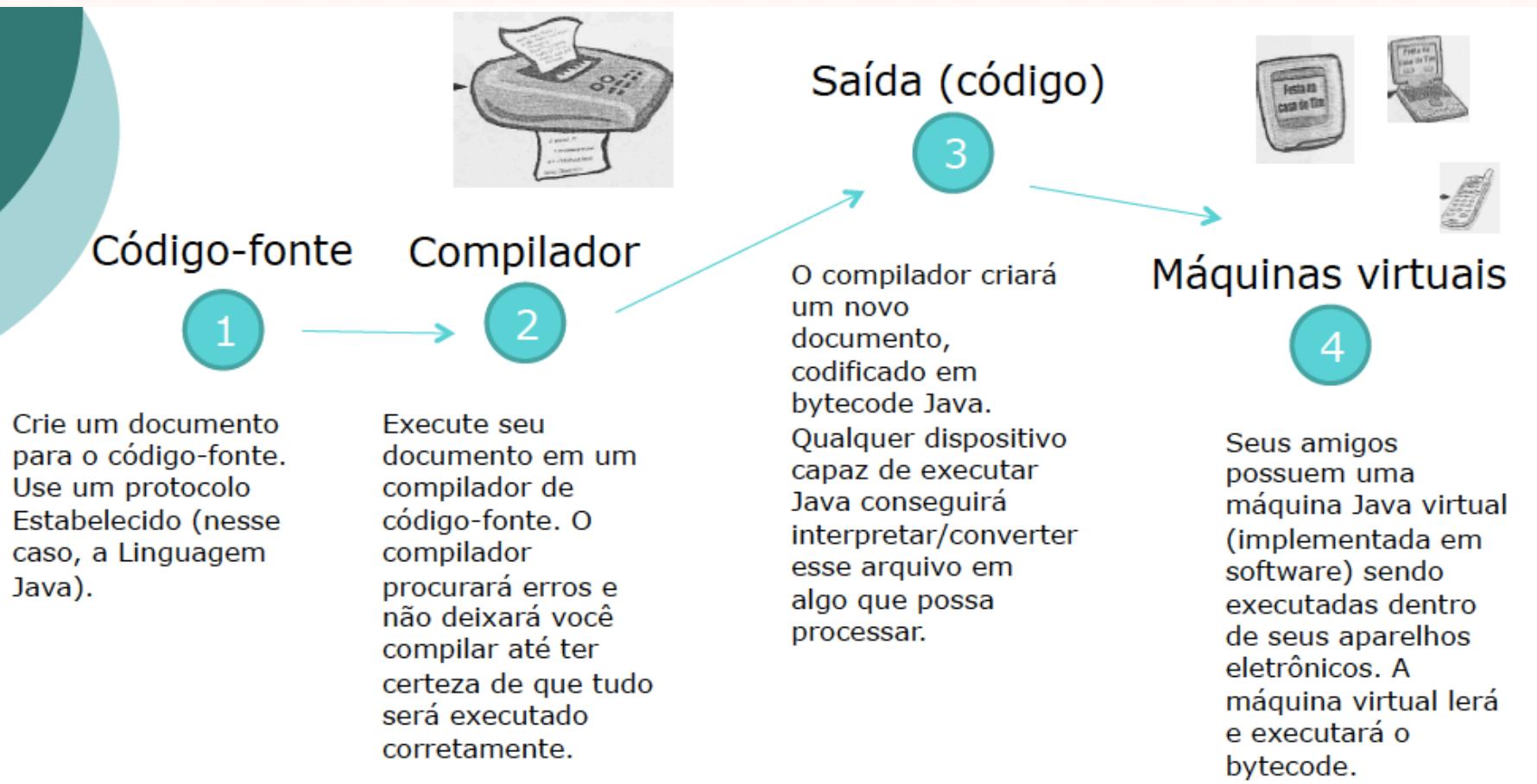
# MERCADO DE TRABALHO

---

- Java no mercado de trabalho.



# COMO O JAVA FUNCIONA



# PORTABILIDADE

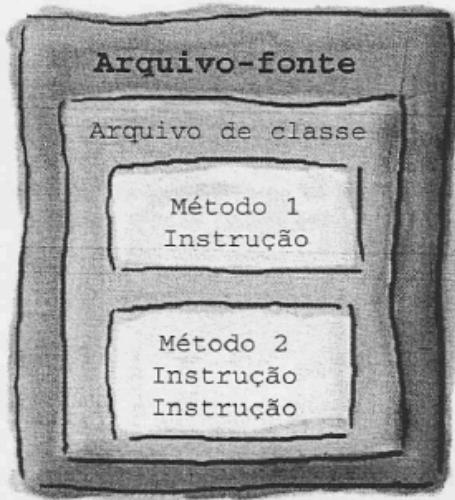
---

- Um programa Java é multiplataforma porque um mesmo Binário Java pode ser executado nas diferentes máquinas que implementam a JVM.



# ESTRUTURA DO CÓDIGO JAVA

## Estrutura do código em Java



### O que existe em um arquivo-FONTE?

Um arquivo de código-fonte (com a extensão `.java`) contém uma definição de **classe**. A classe representa uma *parte* de seu programa, embora um aplicativo muito pequeno possa precisar apenas de uma classe. A classe deve ficar dentro de uma par de chaves.

```
public class Dog{  
}  
} classe
```

### O que existe em uma CLASSE?

Uma classe tem um ou mais **métodos**. Na classe Dog, o método `bark` conterá instruções de como o cão deve latir. Seus métodos devem ser declarados *dentro* de uma classe (em outras palavras, dentro das chaves da classe).

```
public class Dog {  
    void bark( ) {  
    }  
}  
} méthodo
```

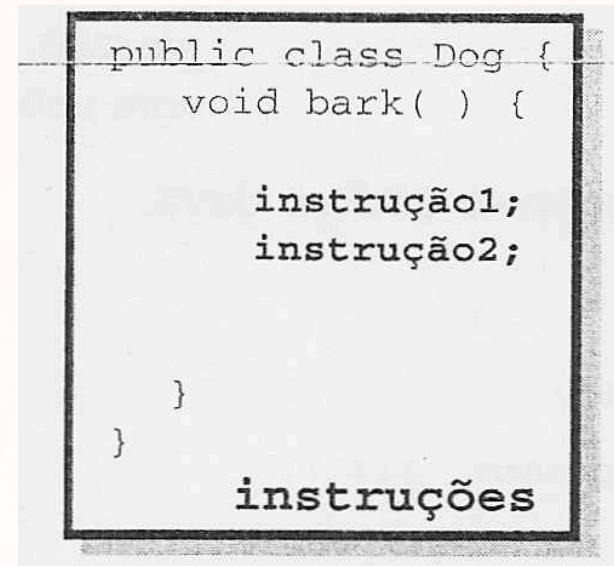
(Fonte: use a cabeça Java)



# MÉTODO

---

- Dentro das chaves de um método
  - são escritas instruções de como ele deve ser executado.
- Código de um método
  - basicamente um conjunto de instruções.



The diagram shows a rectangular frame containing Java code. The code defines a class named 'Dog' with a method 'bark()' that contains two statements: 'instrução1;' and 'instrução2;'. The entire block of code is labeled 'instruções' at the bottom right. The frame has a double border: a thin black one on the left and a thicker grey one on the right.

```
public class Dog {  
    void bark( ) {  
  
        instrução1;  
        instrução2;  
  
    }  
}  
  
instruções
```

(Fonte: use a cabeça Java)

# ANATOMIA DE UMA CLASSE

---

- Ao executar uma classe

- Método especial é procurado.
  - Deve ser escrito exatamente igual a figura abaixo:

```
public static void main (String[] args) {  
    // seu código entra aqui  
}
```

(Fonte: use a cabeça Java)



# ANATOMIA DE UMA CLASSE

---

- Toda classe Java que você irá executar precisa ter o método **main**. Esse é o método onde as instruções são iniciadas.



# ANATOMIA DE UMA CLASSE

por aplicativo).

pública para que todos possam acessá-la

essa é uma classe (duh)

o nome dessa classe

chave de abertura da classe

public class MyFirstApp {

(abordaremos isso depois)

o tipo de retorno. void significa que não há valor de retorno

o nome desse método

argumentos do método. Esse método deve receber uma matriz de strings, e a matriz se chamará 'args'

chave de abertura do método

public static void main (String[ ] args) {

System.out.print("I Rule!");

essa instrução significa exibir na saída padrão (o padrão é a linha de comando)

a string que você quer exibir

toda instrução DEVE terminar com ponto-e-vírgula!!

}

chave de fechamento do método main

}

chave de fechamento da classe MyFirstApp

Não se preocupe em memorizar nada agora. Este capítulo é apenas uma apresentação.

(Fonte: use a cabeça Java)

# EXERCÍCIO

---

- Abrir o notepad.
- Criar diretório padrão no c:\dev
- Criar um arquivo chamado MyFirstApp.java
- Digite o código abaixo e salve o arquivo.

```
public class MyFirstApp {  
    public static void main(String[] args) {  
        System.out.println("Meu primeiro programa em Java...");  
    }  
}
```



# PRIMEIRA CLASSE

---

- Não esqueça:
  - Toda classe Java que você irá executar precisa ter o método **main**.
- Considere “class <nome>“ o nome do “programa”.
- **public static void main(String args[])**
  - deve sempre ser declarado para que o programa seja executado, ou seja, toda classe Java que você irá executar precisa ter o método **main**.
- **System.out.println**
  - Por enquanto, considere uma função para exibir mensagens na tela.



# VARIÁVEIS

---

- Variáveis são alocações de memória nas quais podemos guardar dados.
- Cada variável deve possuir:
  - Nome
  - Tipo
- Valor da variável
  - opcional em alguns casos, veremos adiante.
- Toda vez que necessitar usar uma variável você precisa declará-la e só então poderá atribuir valores a mesma.



# CLASSE X OBJETO

---

- Uma **classe** é o **projeto** de um **objeto**.
- Ela informa ao Java como criar um objeto desse tipo específico. Cada objeto criado a partir dessa classe terá seus próprios valores para as variáveis de instância da classe.



(Fonte: use a cabeça Java)

# CLASSE X OBJETO

As coisas que um objeto *conhece* sobre ele são chamadas de **variáveis de instância**. Elas representam o estado de um objeto (os dados) e podem ter valores exclusivos para cada objeto desse tipo.

Considere **instância** como outra maneira de dizer **objeto**.

As coisas que um objeto *faz* são chamadas de **métodos**. Quando projetar uma classe, você pensará nos dados que um objeto terá que conhecer sobre si mesmo e também projetará os métodos que operarão sobre esses dados. É comum um objeto ter métodos que leiam ou gravem os valores das variáveis de instância. Por exemplo, os objetos Despertador têm uma variável de instância que armazena a hora de despertar e dois métodos que capturam e configuram essa hora.

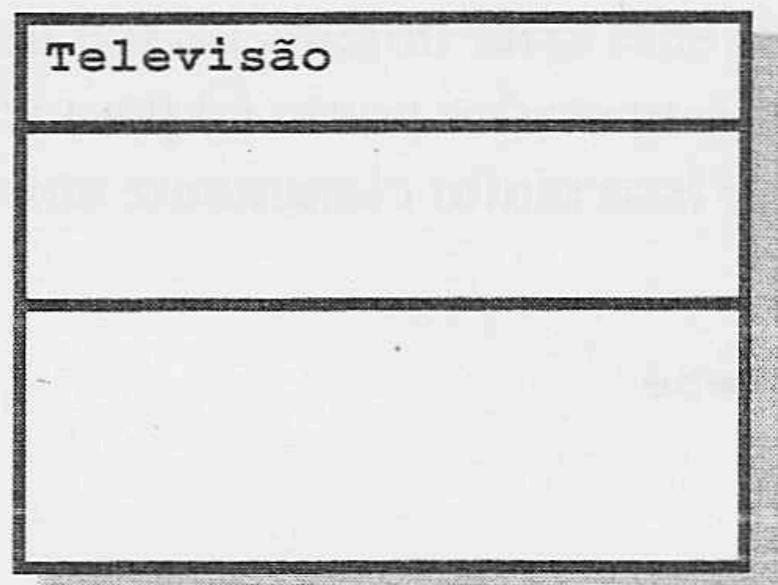
Portanto, os objetos têm variáveis de instância e métodos, mas essas variáveis de instância e métodos são projetadas como parte da classe.

(Fonte: use a cabeça Java)

# CLASSE X OBJETO

---

Preencha com o que um objeto televisão pode ter que saber e fazer.



variáveis de  
instância (estado)

métodos  
(comportamento)

(Fonte: use a cabeça Java)



# CRIANDO O PRIMEIRO OBJETO

---

- Nesse caso, você deverá criar duas classes. Uma para criar uma classe a partir da qual serão criados os objetos, por exemplo, **Carro** e outra para testar esse objeto.



# ECLIPSE

## 1 Crie sua classe

```
class Dog {  
    int size;  
    String breed; ← variáveis de instância  
    String name;  
  
    void bark() {  
        System.out.println("Ruff! Ruff!"); ← um método  
    }  
}
```



## 2 Crie uma classe testadora (TestDrive)

```
class DogTestDrive {  
    public static void main (String[] args) {  
        // o código de teste de Dog entra aqui  
    }  
}
```

Apenas um método main (forneceremos um código para ele na próxima etapa)

(Fonte: use a cabeça Java)

# VARIÁVEIS

---

- As declarações de variáveis consistem de um tipo e um nome de variável. Exemplo:
  - `int idade;`
  - `String nome;`
  - `boolean existe;`
- Em Java, os nomes de variáveis podem começar com uma letra, um sublinhado (`_`), ou um cífrão (`$`). Elas não podem começar com um número.
- Depois do primeiro caractere pode-se colocar qualquer letra ou número.



# VARIÁVEIS E TIPOS DE DADOS

---

- Toda variável deve possuir um tipo. Os tipos que uma variável pode assumir são:
  - Um dos 8 tipos básicos de dados
  - O nome de uma classe ou interface
  - Um “Array” de um dos dois tipos anteriores
- Veremos mais sobre o uso de arrays e classes Posteriormente.
- Os oito tipos básicos de dados são:
  - 4 tipos inteiros
  - 2 tipos de números de ponto-flutuante
  - Tipo caracter
  - Tipo booleano (verdadeiro / falso)



# DEBUG

---

- Debug no eclipse

