

# Aplicação mínima na linguagem Java

Prof. Alcides Calsavara

PUCPR

# Classes

---

Todo código em Java é organizado em classes.

Assim, uma aplicação Java consiste num conjunto de classes.

Cada classe possui um nome (um identificador)

e um corpo (circundado por chaves),

onde são definidos os seus atributos (campos) e métodos (funções).

No exemplo abaixo, "class" é uma palavra reservada

enquanto que "Teste" é um identificador (o nome da classe).

O corpo da classe é vazio neste caso.

```
class Teste  
{  
}
```

# Java: aplicação mínima

---



```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

# Java: aplicação mínima

---

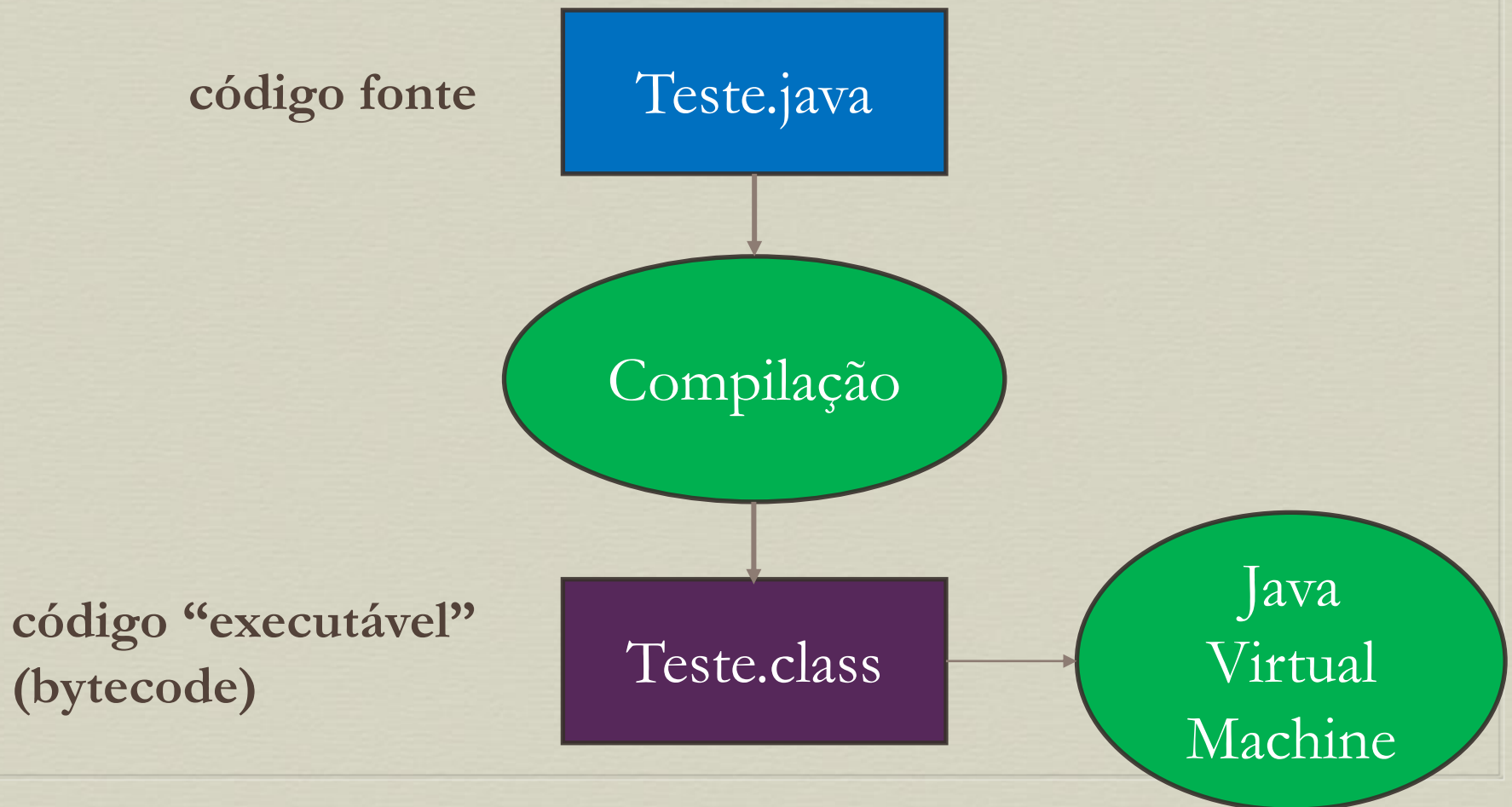
```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

Código Python equivalente:

```
print( "Objetos" )
```

# Arquivos fonte e executável

---





# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

Todo código Java é escrito em forma de classes.

Toda classe tem um nome e um escopo delimitado por um par de chaves.

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

A execução de um código Java inicia-se pelo método `main` contido em alguma classe.

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

Um método Java é semelhante a uma função Python.

```
def main( args )  
    print( "Objetos" )
```



# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

O método main sempre possui os qualificadores **public** e **static**.

O método main tem retorno **void**.

O método main tem um parâmetro do tipo `String[]`.

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

A implementação de um método é delimitada por um par de chaves.

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

O comando `System.out.println` imprime uma string na saída padrão (tela).

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

Todo comando em Java é terminado por ponto-e-vírgula.

# Java: aplicação mínima

---

```
class Teste {  
    public static void main( String[] args )  
    {  
        System.out.println( "Objetos" );  
    }  
}
```

A formatação de um código Java é livre, isto é, espaçamento (vertical ou horizontal) não possui significado semântico.



# Java: aplicação mínima

---

```
class Teste
{
    public static void main( String[] args )
        { System.out.println( "Objetos" ); }
}
```

A formatação de um código Java é livre, isto é, espaçamento (vertical ou horizontal) não possui significado semântico.

# Java: aplicação mínima

---



```
class Teste{  
public static void main( String[] args )  
{ System.out.println( "Objetos" ); } }
```

A formatação de um código Java é livre, isto é, espaçamento (vertical ou horizontal) não possui significado semântico.

# Comentários

---



// comentario de apenas uma linha (inicia-se por duas barras)

/\* comentario que  
se estende  
por diversas  
linhas  
(inicia-se com barra-asterisco e  
termina com asterisco-barra)

\* /

/\* assim tambem eh um comentario \*/