

Introdução a Algoritmos e Programação

IFRS Campus Erechim

Prof. Alexandre M. dos S. Adário

Fundamentos da Linguagem **JAVA**



Máquina computacional abstrata (imaginária)
Independente de S.O. e HW.



Funciona como o seu **NOVO
Hardware + Sistema Operacional**



Conjunto de instruções próprio.

Lê bytecodes compilados.

Derivado de um código Java ou não.

MeuPrograma.java



```
MeuPrograma.java
_____
_____
_____
_____
```

Compilador



```
Compilador
```

MeuPrograma.class



```
MeuPrograma.class
_____
_____
_____
_____
```

Interpretador



```
Interpretador
```

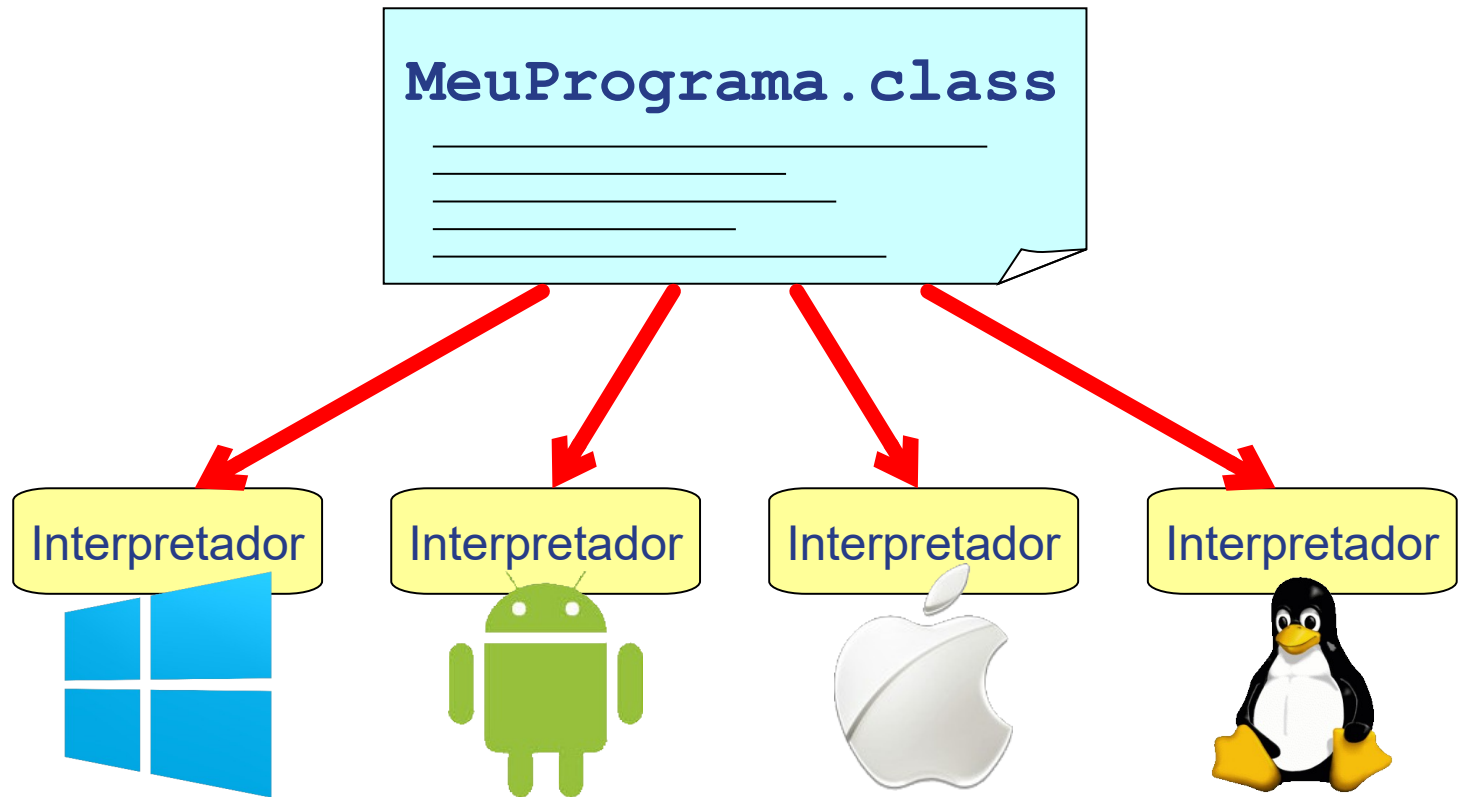
JVM



```
JVM
```



Independência de Plataforma



“Write once, Run anywhere”

Um Exemplo Inicial


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

```
1 class Greeting {  
2     void main() {  
3         String message = "Hello, World!";  
4         System.out.println( message );  
5     }  
6 }
```

A partir da versão 21 do
JDK, é possível
simplificar um pouco a
escrita do código básico.



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

Até a versão 20 do JDK,
essa era a única opção
para escrever um código
executável básico.



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

Java é uma linguagem
orientada a objetos.
Para que o programa execute,
é preciso ter um **OBJETO**
“ativo”.



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

Todo objeto tem um “modelo” para ser “construído”, a classe.
Todo objeto pertence a uma **CLASSE**



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

A “interação” com os objetos é feita com **MÉTODOS**. Se não há métodos, não conseguimos “conversar” com um objeto.



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

Um objeto/classe que será executado no sistema deve ter **obrigatoriamente** um método padrão denominado **main**.



```
1 class Greeting {  
2     void main() {  
3         String message = "Hello, World!";  
4         System.out.println( message );  
5     }  
6 }
```

Observe que, mesmo na
versão mais atual,
simplificada, existe um
método padrão **main**.




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

O método **main** precisa ser **public** (acessível ao mundo exterior) e não deve ter resultado de retorno (**void**).



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

O método **main** também deve ser **static** (acessível diretamente da classe, sem objeto) e ter parâmetros de entrada (**String[] args**)



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

Um método pode ter variáveis
declaradas no seu escopo.

message é uma variável, objeto da
classe **String** (cadeia de caracteres)



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

A variável **message** possui um valor inicial. Cadeias de caracteres (strings) sempre são definidos entre aspas duplas.



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

A classe **System**
representa o computador em
que o programa é executado.



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

Nessa classe, existe um objeto **System.out**, que representa a saída padrão (a tela do computador).



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

Para que seja impressa uma mensagem na tela, **invocamos** um método desse objeto, como se estivéssemos fazendo um pedido a ele.

