



Algoritmos e Programação I

Introdução a Linguagem de Programação Java

Prof. Alexandre Mignon

História do Java

- Criada por James Gosling da Sun Microsystems e lançada em 1996;
- Criada inicialmente para execução em televisores, eletrodomésticos, telefones, etc;
- Utilizou como ponto de partida o mesmo estilo de sintaxe do C/C++;
- Utiliza o conceito de WORA (*Write Once, Run Anywhere*);

Características do Java

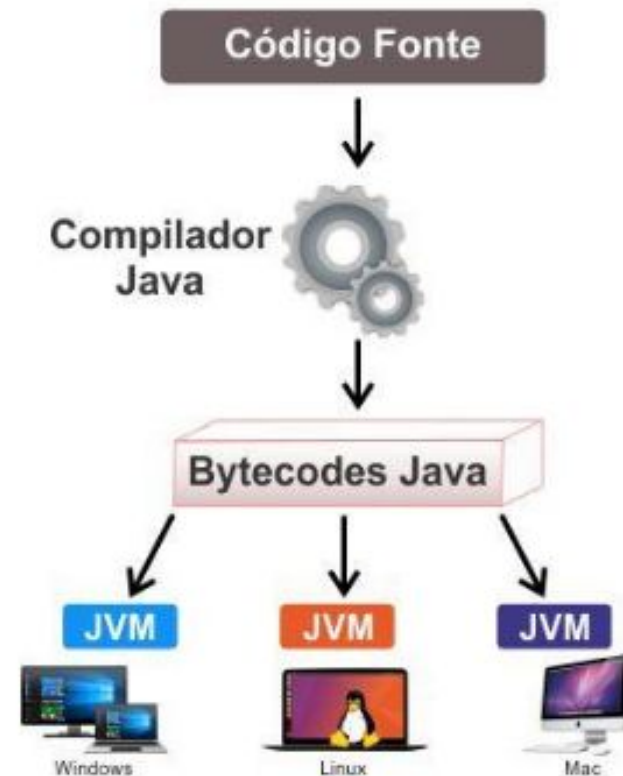
- Maior facilidade na programação;
- Escreva uma vez, execute em qualquer lugar;
- Portabilidade do código;
- Possibilidade de programas executarem mais de uma tarefa (multithreading);
- Programação centrada na rede;



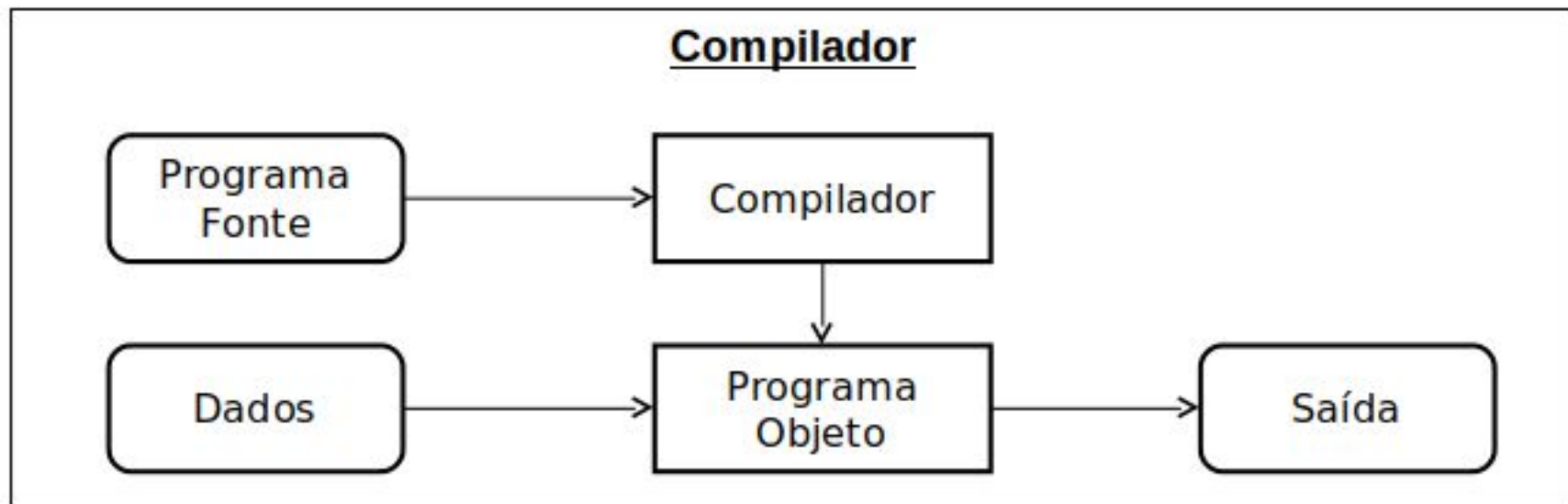
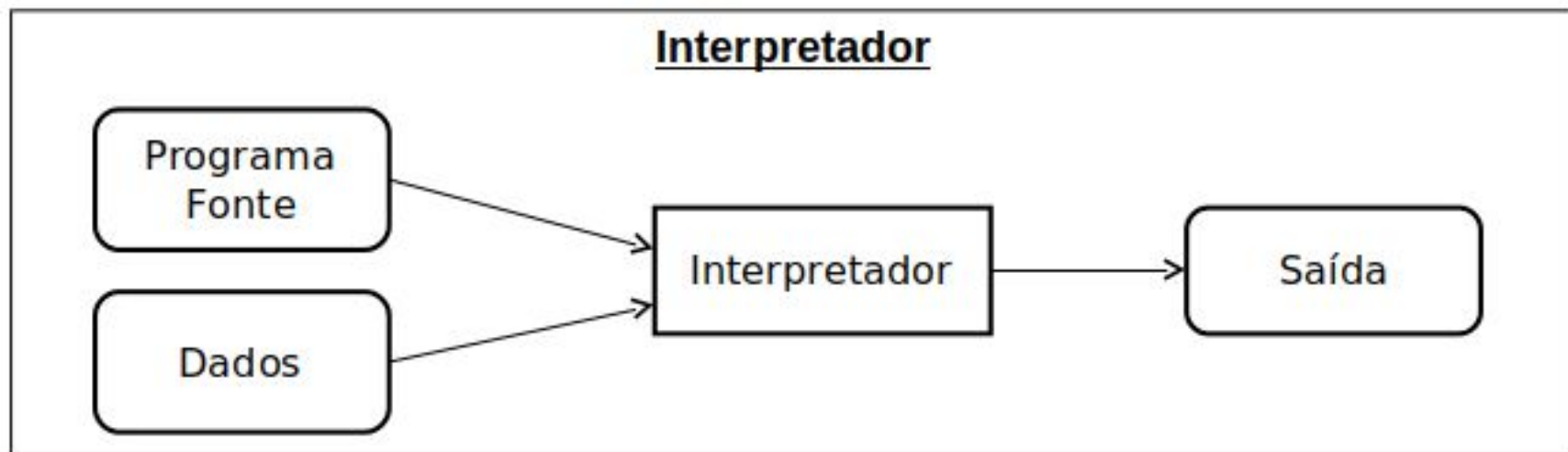
JVM - Java Virtual Machine

- Simula uma máquina real para o *bytecode*;
- Interpreta *bytecodes* (que são independentes de plataforma de hardware);
- Pode ser implementada tanto na forma de software como de hardware;
- Possui código compacto;
- Torna a linguagem JAVA portátil para diversas plataformas;

Write Once, Run Anywhere



Compilador x Interpretador



Exemplo - Ola Mundo

```
package exemplo;

/**
 * Meu primeiro programa Java.
 */
public class OlaMundo {

    public static void main(String[] args) {
        // Apresenta a mensagem Ola Mundo!
        System.out.println("Ola Mundo!");
    }
}
```

Pontos Importantes

- Palavra reservada *class* seguida pelo nome do algoritmo
 - ◆ O nome não deve conter espaços;
 - ◆ Deve ser iniciado por uma letra;
 - ◆ O caractere sublinhado (_) e o sinal \$ são considerados letras em classes, métodos e campos em Java;
 - ◆ Nome de classe não deve conter acentos e pode conter números;
- Caracteres de delimitação { } (início e fim)
- Caracteres maiúsculos e minúsculos são diferentes
 - ◆ Case sensitive
- Nome do arquivo deve ter o mesmo nome da classe
 - ◆ Extensão .java

Método *main*

- Método onde o programa inicia;
- Pode estar presente em qualquer classe;
- Os parâmetros de linha de comando são enviados para array `args[]`, do tipo `String`.

Comentários em Java

- Comentários servem apenas para explicar o código
 - ◆ São ignorados pelo compilador
- Para fazer comentários em Java, pode-se utilizar:
 - ◆ `//` para comentar até o final da linha
 - ◆ `/* */` para comentar o que estiver entre eles

```
/* comentario daqui,  
ate aqui */
```

```
// uma linha de comentário sobre a idade  
int idade;
```

Palavras Reservadas

abstract	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	extends	final	finally	float
for	goto	if	implements	import	instanceof
int	interface	long	native	new	package
private	protected	public	return	short	static
strictfp	super	switch	synchronized	this	throw
throws	transient	try	void	volatile	while
assert					

Tipos de Dados

Tipo	Descrição
boolean	Pode assumir os valores true ou false
char	Representa um caractere Unicode de 16 bits para armazenar dados alfanuméricos
byte	Inteiro de 8 bits. Pode assumir valores entre -2^7 e 2^7-1 (de -128 a 127)
short	Inteiro de 16 bits. Pode assumir valores entre -2^{15} e $2^{15}-1$ (de -32.768 a 32.767)
int	Inteiro de 32 bits. Pode assumir valores entre -2^{31} e $2^{31}-1$ (de -2.147.483.648 a 2.147.483.647)
long	Inteiro de 64 bits. Pode assumir valores entre -2^{63} e $2^{63}-1$
float	Número de ponto flutuante de 32 bits. Pode assumir valores entre $\pm 1.40129846432481707e-45$ e $\pm 3.40282346638528860e+38$
double	Número de ponto flutuante de 64 bits. Pode assumir valores entre $\pm 4.94065645841246544e-324$ e $\pm 1.79769313486231570e+308$

Tipo	Descrição
String	Cadeia de caracteres que usam 2 bytes por caractere. Strings podem ser vazias (zero caractere) e conter qualquer tipo de caractere.

Declaração de Variável

Tipo Variável

int

pesoVeiculo;

Nome Variável

Exemplos:

- float precoProduto;
- byte idadeAluno;
- char sexoFuncionario;
- String nome_Aluno;
- boolean maiorIdade;

Atribuição de Valores a Variáveis

Nome Variável

`pesoVeiculo = 1500;`

Valor Variável

Exemplos:

- `float precoProduto = 3.5;`
- `byte idadeAluno = 18;`
- `char sexoFuncionario = 'M';`
- `String nome_Aluno = "João Pedro";`
- `boolean maiorIdade = false;`

Operadores Aritméticos

Operador	Operação	Prioridade
*	Multiplicação	1
/	Divisão	1
%	Resto da Divisão (inteiros)	1
+	Adição	2
-	Subtração	2

- Parênteses permitem alterar a ordem de precedência dos operadores.
- Operadores de mesma prioridade, a avaliação é feita da esquerda para a direita.

Operadores de Incremento e Decremento

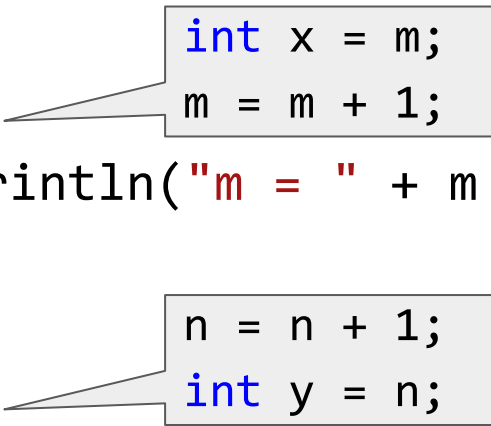
Operador	Operação	Exemplo	Equivalente a
++	Incremento	x++	x = x + 1
--	Decremento	x--	x = x - 1

→ Trabalha de dois modos:

- ◆ Pré-fixado;
- ◆ Pós- fixado;
- ◆ A diferença só existe quando utilizado em expressões ou atribuições.

Exemplo

```
public class Exemplo {  
    public static void main(String[] args) {  
        int m = 5;  
        int x = m++;  
        System.out.println("m = " + m + " x = " + x);  
  
        int n = 5;  
        int y = ++n;  
        System.out.println("n = " + n + " y = " + y);  
    }  
}
```



```
m = 6 x = 5  
n = 6 y = 6
```


Operadores Aritméticos de Atribuição

Operador	Operação	Exemplo	Equivalente a
<code>+=</code>	Adição e Atribuição	<code>x += 2</code>	<code>x = x + 2</code>
<code>-=</code>	Subtração e Atribuição	<code>x -= 2</code>	<code>x = x - 2</code>
<code>*=</code>	Multiplicação e Atribuição	<code>x *= 2</code>	<code>x = x * 2</code>
<code>/=</code>	Divisão e Atribuição	<code>x /= 2</code>	<code>x = x / 2</code>
<code>%=</code>	Módulo e Atribuição	<code>x %= 2</code>	<code>x = x % 2</code>

Operadores Relacionais

Operador	Operação
==	Igual a
!=	Diferente
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a

Operadores Lógicos

Operador	Operação	Prioridade
!	Negação	1
& &	Conjunção	2
	Disjunção	3

`(2 > 5) && (5 < 6) → falso`

`(2 > 3) || (5 < 6) → verdadeiro`

`!(2 > 3) → verdadeiro`

`!(2 < 5) || (15 / 3 == 5) → verdadeiro`

`!((2 < 5) || (15 / 3 == 5)) → falso`

Parênteses alteram a ordem
de prioridade dos
operadores

Conversão de Variáveis

Conversões

IMPLICITAS

Nenhuma sintaxe especial é necessária porque a conversão é de tipo seguro e nenhum dado será perdido. Exemplo:

- byte para int ou long para float → `byte b = 10; int i = b;`

Conversões

EXPLICITAS

As conversões explícitas exigem um operador cast. A conversão é necessária quando as informações podem ser perdidas na conversão ou quando a conversão pode não funcionar por outros motivos. Exemplo:

- Float para long ou int para byte → `int i = 10; byte b = (byte)i;`

Conversões

OUTRAS

O restante dos tipos de conversão disponíveis ou criados no programa. Exemplo:

- String para int → `String s = "123"; int i = Integer.parseInt(s);`

Conversão de Variáveis

PARA:	byte	short	char	int	long	float	double
DE:							
byte	----	<i>Impl.</i>	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
short	(byte)	----	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
char	(byte)	(short)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
int	(byte)	(short)	(char)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
long	(byte)	(short)	(char)	(int)	----	<i>Impl.</i>	<i>Impl.</i>
float	(byte)	(short)	(char)	(int)	(long)	----	<i>Impl.</i>
double	(byte)	(short)	(char)	(int)	(long)	(float)	----

Impl. Significa Implícita

Saída de Dados

→ Para apresentar dados na tela utiliza-se um dos comandos abaixo:

- ◆ `System.out.print()` - Apresenta a mensagem e coloca o cursor ao lado.
- ◆ `System.out.println()` - Apresenta a mensagem e colocar o cursor na linha de baixo.

```
System.out.println("Ola Mundo!");
```

```
System.out.print(x);
```

```
System.out.println("Idade = " + idade);
```

```
System.out.println("X = " + x + " Y = " + y);
```

Entrada de Dados - A Classe *Scanner*

→ Engloba diversos métodos para facilitar a leitura de dados pelo teclado.

```
package exemplo;
```

```
import java.util.Scanner;
```

1

```
public class ExemploScanner {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Digite seu nome: ");
```

```
        String nome = sc.next();
```

```
        System.out.println("Ola, " + nome + "!");
```

```
    }
```

```
}
```

2

3

A Classe *Scanner*

1. Importar o pacote `java.util.Scanner`;
2. Declarar um objeto do tipo `Scanner`;
3. Utilizar um dos métodos fornecidos pelo objeto para obter os dados do usuário.

A Classe Scanner

Método	Finalidade
next()	Aguarda uma entrada em formato String (primeira palavra)
nextLine()	Aguarda uma entrada em formato String (a linha inteira)
nextInt()	Aguarda uma entrada em formato Inteiro
nextByte()	Aguarda uma entrada em formato Inteiro
nextLong()	Aguarda uma entrada em formato Inteiro Longo
nextFloat()	Aguarda uma entrada em formato Número Fracionário
nextDouble()	Aguarda uma entrada em formato Número Fracionário

Exemplo

```
package exemplo;

import java.util.Scanner;

public class ExemploCirculo {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Raio: ");
        float raio = teclado.nextFloat();
        float area = 3.14f * (raio * raio);
        System.out.println("Area = " + area);
    }
}
```

Exercícios

1. Em uma empresa, há tabelas com o que foi gasto em cada mês. Para fechar o balanço do primeiro trimestre, é necessário somar o gasto total. Sabendo que, em janeiro, foram gastos 15000 reais, em fevereiro, 23000 reais e em março, 17000 reais, desenvolva um programa que calcule e apresente o gasto total no trimestre. Siga os seguintes passos:
 - a. Criar uma classe chamada `BalancoTrimestral` com um bloco *main*.
 - b. Dentro do *main*, declarar uma variável real denominada `gastosJaneiro` e inicializá-la com 15000;
 - c. Declarar as variáveis `gastosFevereiro` e `gastosMarco`, inicializando-as com 23000 e 17000, respectivamente;
 - d. Declarar uma variável denominada `gastosTrimestre` e inicializá-la com a soma das outras três variáveis;
 - e. Imprimir o valor da variável `gastosTrimestre`.

Exercícios

2. Adicionar código (sem alterar as linhas que já existem) na classe criada anteriormente para imprimir a média mensal de gasto, criando uma variável `mediaMensal` junto com uma mensagem. Para isso, concatenar a `String` com o valor, usando:

```
"Valor da média mensal = " + mediaMensal
```

Exercícios

3. Criar um programa que leia o valor correspondente ao salário mensal de um trabalhador e também o valor do percentual de reajuste a ser atribuído ao salário. Apresente na tela o valor do novo salário.
4. Criar um programa que receba uma determinada hora (hora e minutos separados) e apresente na tela a hora em minutos.
5. Uma conta telefônica é composta dos seguintes custos:
 - assinatura: R\$ 17,90
 - chamada local: R\$ 0,04 por minuto
 - chamada para celular: R\$ 0,20 por minuto
 - interurbanos

Criar um programa que calcule o valor total a ser pago sabendo será fornecida a quantidade de minutos locais e para celular. O valor gasto em interurbanos foi de R\$ 34,29. Apresentar na tela os custos e o valor total a ser pago.