



## **Curso de Engenharia da Computação**

*Paradigmas de Programação (60h/a)*

# ***Introdução à Linguagem Java usando Exemplos***

## **Aula 03**

**Prof. Lenardo Chaves e Silva, D.Sc.** 

*lenardo@ufersa.edu.br*

segunda-feira, 27 de novembro de 2017



**Universidade Federal Rural do Semi-Árido (UFERSA)**

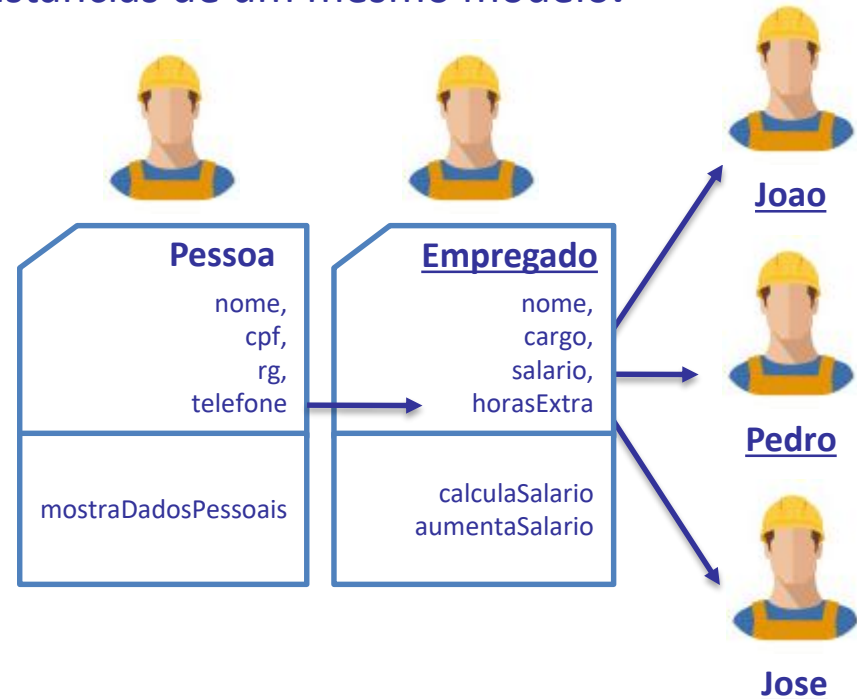
- Ambiente de Desenvolvimento Java Típico;
- Tipos Primitivos e seus Limites de Representação;
- Conversão entre Tipos;
- Operadores Java;
- Série de exemplos de programas em Java.
- Exercícios de Fixação;
- Referências.

## O que vimos antes...

**Modelos:** representações simplificadas (i.e., abstrações) de entidades do mundo real (e.g., itens, pessoas, animais, tarefas, conceitos, etc.), independentes de computadores.

☛ **Reuso:** Representar diferentes instâncias de um mesmo modelo:

- $\neq$  dados e  $\equiv$  operações.



## Alguns Conceitos Fundamentais:

- ☛ **Abstração** → ocultar detalhes da Implementação (complexidade);
- ☛ **Encapsulamento** → restringir o acesso direto aos Dados;
- ☛ **Classes:** equivalem aos modelos de entidades do mundo real;
- ☛ **Objetos:** tipo de dado específico derivado a partir de uma classe;
- ☛ **Atributos:** dados contidos na classe ← encapsulados;
- ☛ **Métodos:** operações pertencentes à classe p/ manipular os seus atributos.

## O que queremos entender?

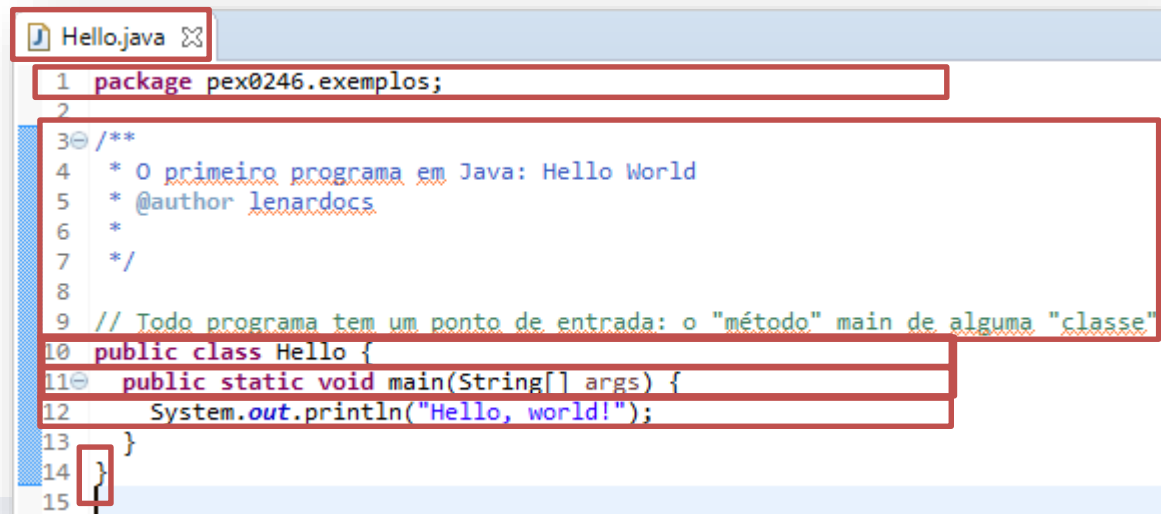
- ☛ Introdução à Java sem nos preocupar (à priori) com POO;
- ☛ Java é + 1 Linguagem de Programação:
  - ☐ Sintaxe  $\neq$  de outras LPs  $\leftrightarrow$  Linguagem C (Imperativa).
- ☛ Como Java lida com...?
  - ☐ Conceitos chave: tipos básicos, constantes, decisões, laços, etc.
  - ☐ Compilação e Execução de um programa;
  - ☐ Ambiente de Desenvolvimento.



## Ambiente de Desenvolvimento Java Típico:

☛ **Primeiro Programa:** Como seria um *HelloWorld* em Java?

- ❑ Linha 1: `package pex0246.exemplos` → pode conter vários programas.
- ❑ **Árvore de diretórios:** `pex0246/exemplos/Hello.java`
- ❑ Nome do Programa (Hello) ↔ Nome do Arquivo (e.g., `Hello.java`);
- ❑ **Java** é “*case-sensitive*”: reconhece diferença de caixa.

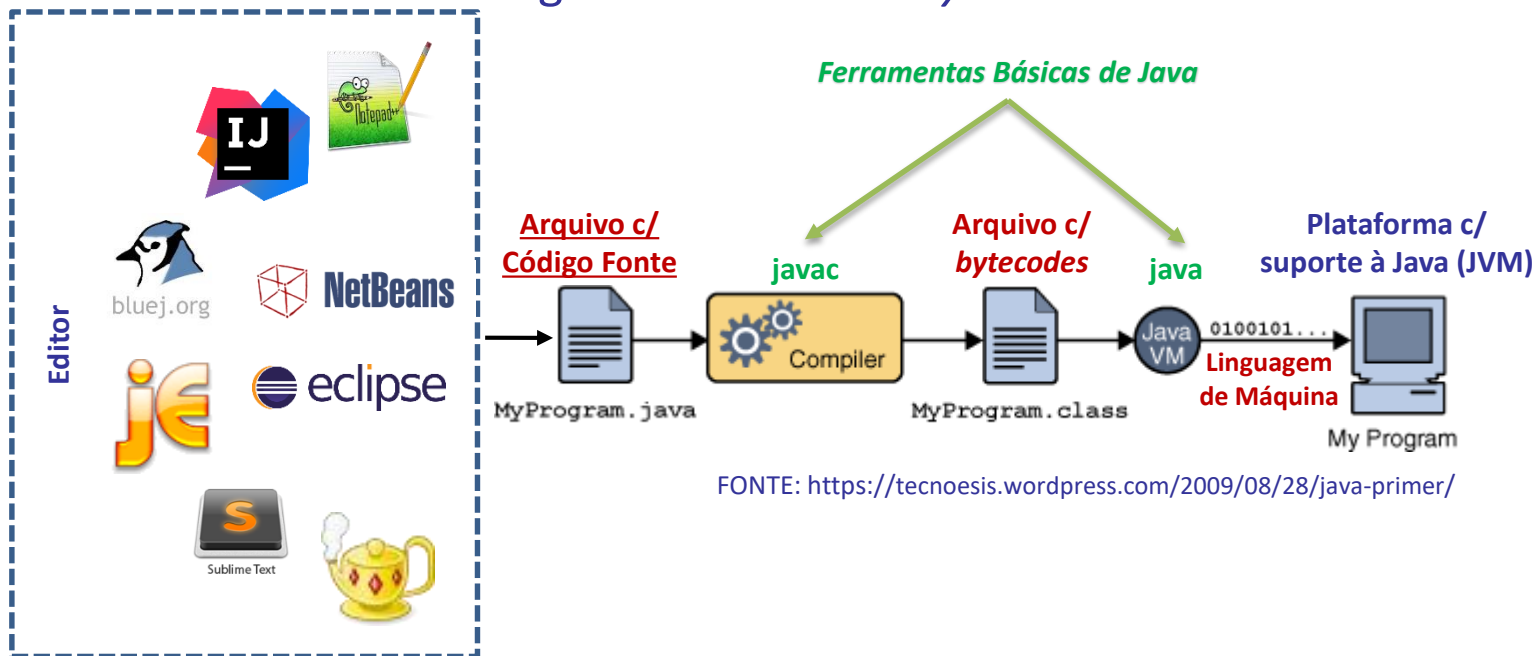


```
1 package pex0246.exemplos;
2
3 /**
4  * O primeiro programa em Java: Hello World
5  * @author lenardocs
6  *
7  */
8
9 // Todo programa tem um ponto de entrada: o "método" main de alguma "classe"
10 public class Hello {
11     public static void main(String[] args) {
12         System.out.println("Hello, world!");
13     }
14 }
15
```

## Ambiente de Desenvolvimento Java Típico:

### ➡ Processo de Compilação:

- ❑ Converter código fonte Java em *bytecodes*<sup>1</sup>.

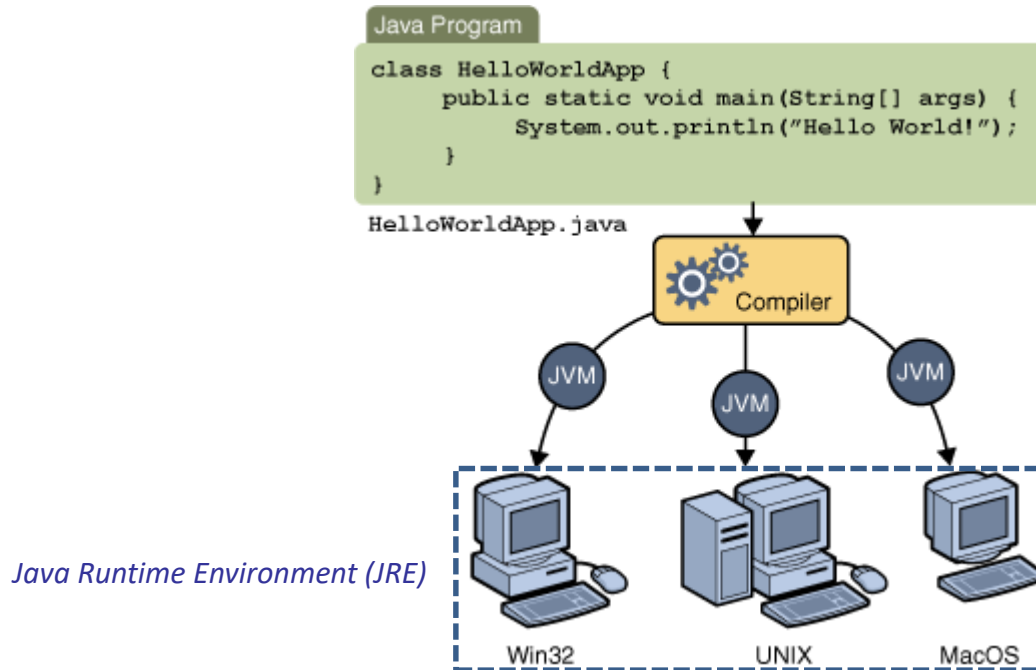


<sup>1</sup> Conjunto de instruções independentes de hardware particular.

## Ambiente de Desenvolvimento Java Típico:

### ☛ Processo de Interpretação ← JVM:

☐ PORTABILIDADE ➔ “Escreva uma vez, execute em qualquer lugar!”



FONTE: <https://tecnoesis.wordpress.com/2009/08/28/java-primer/>



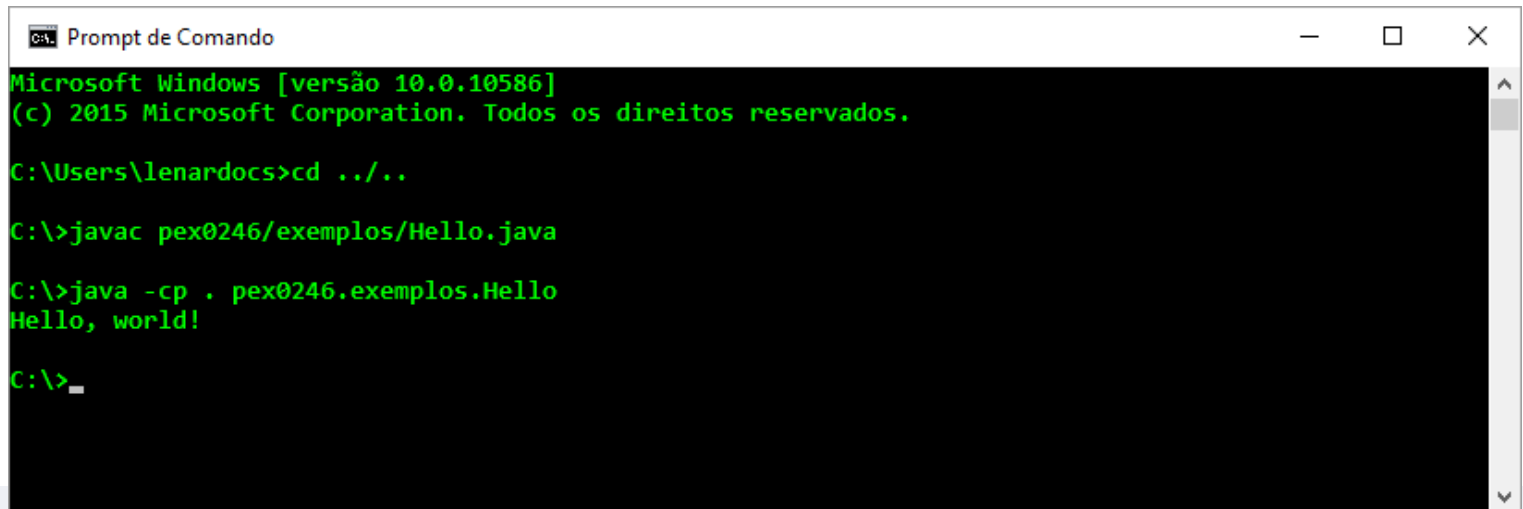
## Ambiente de Desenvolvimento Java Típico:

### ☛ Processo de Execução (Compilação + Interpretação):

❑ Comando p/ Compilar: `javac NomePrograma.java`

❑ Comando p/ Executar: `java -cp . NomeProgramaCompleto`

- `-cp` indica a partir de onde na árvore de diretórios do SO o programa **NomeProgramaCompleto** será procurado.



```
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\Users\lenardocs>cd ../../

C:\>javac pex0246/exemplos/Hello.java

C:\>java -cp . pex0246.exemplos.Hello
Hello, world!

C:\>_
```

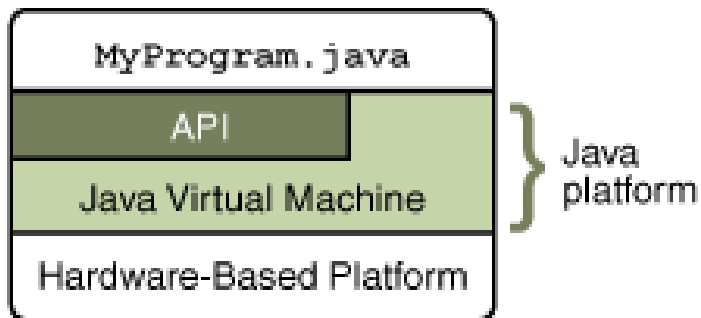
## Ambiente de Desenvolvimento Java Típico:

### ☛ Processo de Execução ← (Compilação + Interpretação):

#### ☐ Plataforma Java:

- Apenas **Software** → roda sobre outras plataformas de **Hardware**.
- **Componentes:**
  1. **JVM** (*Java Virtual Machine*) → base para plataforma Java.
  2. **Java API** (*Application Programming Interface*) → coleção de componentes de software prontos p/ reuso → recursos úteis:

- Bibliotecas de Classes ≡ Pacotes;
- Interfaces.



## Ambiente de Desenvolvimento Java Típico:

### ☛ Processo de Execução (Compilação + Interpretação):

#### Erros:

- **Sintaxe/Compilação:** código que viola as regras da linguagem Java e são detectados durante a etapa de compilação.

**Ex.:** testing.java:14: `;' expected.

```
System.out.println("Input has " + count + " chars.")
```

^

1 error

- **Execução:**

- **Fatais:** causam o término imediato dos programas (insucesso).
- **Não Fatais:** permitem os programas terminarem a execução e produzem resultados incorretos.

**Ex.:** Divisão por zero,

**Obs.:** Problemas comuns e [soluções](#).

## Segundo Programa: Mín. e Máx. dentre 3 inteiros (Solução 1):

```
MinMax1.java
1 package pex0246.exemplos;
2
3 * O segundo programa em Java: Ler 3 números inteiros da entrada e
10
11 import java.util.Scanner;
12
13 public class MinMax1 {
14
15     public static void main(String[] args) {
16
17         Scanner sc = new Scanner(System.in);
18
19         int n1, n2, n3;
20
21         System.out.print("Entre com o primeiro inteiro: ");
22         n1 = sc.nextInt();
23
24         System.out.print("Entre com o segundo inteiro: ");
25         n2 = sc.nextInt();
26
27         System.out.print("Entre com o terceiro inteiro: ");
28         n3 = sc.nextInt();
29
30         if (n1 > n2) {
31             if (n1 > n3) {
32                 if (n2 < n3) {
33                     System.out.println("O menor numero eh: " + n2);
34                 } else {
35                     System.out.println("O menor numero eh: " + n3);
36                 }
37                 System.out.println("O maior numero eh: " + n1);
38             }
39             if (n1 < n2) {
40                 System.out.println("O menor numero eh: " + n1);
41             } else {
42                 System.out.println("O menor numero eh: " + n2);
43             }
44             System.out.println("O maior numero eh: " + n3);
45         }
46     } else {
47         if (n2 > n3) {
48             if (n1 < n3) {
49                 System.out.println("O menor numero eh: " + n1);
50             } else {
51                 System.out.println("O menor numero eh: " + n3);
52             }
53             System.out.println("O maior numero eh: " + n2);
54         } else {
55             if (n1 < n2) {
56                 System.out.println("O menor numero eh: " + n1);
57             } else {
58                 System.out.println("O menor numero eh: " + n2);
59             }
60             System.out.println("O maior numero eh: " + n3);
61         }
62     }
63 }
64 }
65 }
```

Sobrecarga do operador '+'

```

38
39
40     } else {
41         if (n1 < n2) {
42             System.out.println("O menor numero eh: " + n1);
43         } else {
44             System.out.println("O menor numero eh: " + n2);
45         }
46         System.out.println("O maior numero eh: " + n3);
47     }
48 } else {
49     if (n2 > n3) {
50         if (n1 < n3) {
51             System.out.println("O menor numero eh: " + n1);
52         } else {
53             System.out.println("O menor numero eh: " + n3);
54         }
55         System.out.println("O maior numero eh: " + n2);
56     } else {
57         if (n1 < n2) {
58             System.out.println("O menor numero eh: " + n1);
59         } else {
60             System.out.println("O menor numero eh: " + n2);
61         }
62         System.out.println("O maior numero eh: " + n3);
63     }
64 }
65 }
```

### Saída do Programa:

```

Entre com o primeiro inteiro: 3
Entre com o segundo inteiro: 2
Entre com o terceiro inteiro: 4
O menor numero eh: 2
O maior numero eh: 4
```

## Segundo Programa: Mín. e Máx. dentre 3 inteiros (Solução 2):

```
MinMax2.java
1 package pex0246.exemplos;
2
3 * 0 segundo programa em Java v2
4
5
6 import java.util.Scanner;
7
8 public class MinMax2 {
9
10 public static void main(String[] args) {
11
12 Scanner sc = new Scanner(System.in);
13
14 int n1, n2, n3;
15
16 System.out.print("Entre com o primeiro inteiro: ");
17 n1 = sc.nextInt();
18
19 System.out.print("Entre com o segundo inteiro: ");
20 n2 = sc.nextInt();
21
22 System.out.print("Entre com o terceiro inteiro: ");
23 n3 = sc.nextInt();
24
25 int mínimo, máximo;
26
27 if (n1 > n2) {
28     if (n1 > n3) {
29         if (n2 < n3) {
30             mínimo = n2;
```

```
31         } else {
32             mínimo = n3;
33         }
34     } else {
35         máximo = n1;
36     } else {
37         if (n1 < n2) {
38             mínimo = n1;
39         } else {
40             mínimo = n2;
41         }
42     }
43     máximo = n3;
44 }
45 } else {
46     if (n2 > n3) {
47         if (n1 < n3) {
48             mínimo = n1;
49         } else {
50             mínimo = n3;
51         }
52     }
53     máximo = n2;
54 } else {
55     if (n1 < n2) {
56         mínimo = n1;
57     } else {
58         mínimo = n3;
59     }
60     máximo = n3;
61 }
62 }
63 }
64 System.out.println("O menor numero eh: " + mínimo);
65 System.out.println("O maior numero eh: " + máximo);
66 }
67 }
```

### Saída do Programa:

```
Entre com o primeiro inteiro: 3
Entre com o segundo inteiro: 2
Entre com o terceiro inteiro: 4
O menor numero eh: 2
O maior numero eh: 4
```

### Perguntas sobre este programa:

1. Você achou o programa "bem escrito"?
  2. É fácil de entender?
  3. É fácil trocar as mensagens de saída por outras?
  4. É fácil assegurar-se de que não há bugs?
  5. É fácil estender para 4 números lidos na entrada?
- Uma das versões deste programa tem um bug: ache-o!

## Segundo Programa: Mín. e Máx. dentre 3 inteiros (Solução 3):

```

1 package pex0246.exemplos;
2
3 * O segundo programa em Java v3
4
5 import java.util.Scanner;
6
7 public class MinMax3 {
8
9     public static void main(String[] args) {
10
11         Scanner sc = new Scanner(System.in);
12
13         int num;
14         int mínimo = Integer.MAX_VALUE;
15         int máximo = Integer.MIN_VALUE;
16
17         System.out.print("Entre com o primeiro inteiro: ");
18         num = sc.nextInt();
19
20         if (num < mínimo) {
21             mínimo = num;
22         }
23
24         if (num > máximo) {
25             máximo = num;
26         }
27
28         System.out.print("Entre com o segundo inteiro: ");
29         num = sc.nextInt();
30
31         if (num < mínimo) {
32             mínimo = num;
33         }
34
35         if (num > máximo) {
36             máximo = num;
37         }
38
39         System.out.print("Entre com o terceiro inteiro: ");
40         num = sc.nextInt();
41
42         if (num < mínimo) {
43             mínimo = num;
44         }
45
46         if (num > máximo) {
47             máximo = num;
48         }
49
50         System.out.println("O menor numero eh: " + mínimo);
51         System.out.println("O maior numero eh: " + máximo);
52     }
53 }

```

— É muito mais simples, não é?  
• Por quê?

### Saída do Programa:

```

Entre com o primeiro inteiro: 3
Entre com o segundo inteiro: 2
Entre com o terceiro inteiro: 4
O menor numero eh: 2
O maior numero eh: 4

```

## Segundo Programa: Mín. e Máx. dentre 3 inteiros (Solução 4):

```
MinMax4.java
1 package pex0246.exemplos;
2
3 * O segundo programa em Java v4
4
5 import java.util.Scanner;
6
7 public class MinMax4 {
8
9     public static void main(String[] args) {
10
11         final int NÚMEROS_A_LER = 3;
12
13         Scanner sc = new Scanner(System.in);
14
15         int mínimo = Integer.MAX_VALUE;
16         int máximo = Integer.MIN_VALUE;
17
18         for (int i = 0; i < NÚMEROS_A_LER; i++) {
19             System.out.print("Entre com o próximo inteiro:");
20             int num = sc.nextInt(); // local da declaração
21             i=i+1;
22
23             if (num < mínimo) {
24                 mínimo = num;
25             }
26
27             if (num > máximo) {
28                 máximo = num;
29             }
30
31             System.out.println("O menor numero eh: " + mínimo);
32             System.out.println("O maior numero eh: " + máximo);
33         }
34     }
35 }
```

### Saída do Programa:

```
Entre com o primeiro inteiro: 3
Entre com o segundo inteiro: 2
Entre com o terceiro inteiro: 4
O menor numero eh: 2
O maior numero eh: 4
```

## Segundo Programa: Mín. e Máx. dentre 3 inteiros (Solução 5):

```

1 package pex0246.exemplos;
2
3 * 0 segundo programa em Java v5
4
5 import java.util.Scanner;
6
7 public class MinMax5 {
8
9     public static void main(String[] args) {
10
11         final int NÚMEROS_A_LER = 3;
12
13         Scanner sc = new Scanner(System.in);
14
15         int mínimo = Integer.MAX_VALUE;
16         int máximo = Integer.MIN_VALUE;
17
18         for (int i = 0; i < NÚMEROS_A_LER; i++) {
19             int num = recebeProximoInteiro(sc);
20             mínimo = menorNumeroEntre(mínimo, num);
21             máximo = maiorNumeroEntre(máximo, num);
22         }
23
24         System.out.println("O menor numero eh: " + mínimo);
25         System.out.println("O maior numero eh: " + máximo);
26     }
27 }

```

```

33 private static int maiorNumeroEntre(int numero1, int numero2) {
34     return numero2 > numero1 ? numero2 : numero1; // Operador ternário
35 }

```

```

37 private static int menorNumeroEntre(int numero1, int numero2) {
38     return numero2 < numero1 ? numero2 : numero1;
39 }

```

```

41 private static int recebeProximoInteiro(Scanner sc) {
42     System.out.print("Entre com o proximo inteiro:");
43     int num = sc.nextInt();
44     return num;
45 }

```

```

private static int maiorNumeroEntre(int numero1, int numero2) {
    if (numero2 > numero1) { // Equivale a:
        return numero2;
    }
    return numero1;
}

```

### Saída do Programa:

```

Entre com o primeiro inteiro: 3
Entre com o segundo inteiro: 2
Entre com o terceiro inteiro: 4
O menor numero eh: 2
O maior numero eh: 4

```



## Tipos Primitivos e seus Limites de Representação:

Tipo Primitivo (Portáveis)	Tamanho	Mínimo	Máximo	Exemplos de Constantes
boolean	1 bit	-	-	true, false
char	16 bits	Unicode 0	Unicode 65.535	'a' (letra a)
byte	8 bits	-128	+127	97, -23 0x65 (hexadecimal)
short	16 bits	$-2^{15}$ (-32.768)	$+2^{15}-1$ (32.767)	17569, -21875
int	32 bits	$-2^{31}$ (uns -2 bi)	$+2^{31}-1$ (uns 2 bi)	1876345240, -2000000000
long	64 bits	$-2^{63}$ (uns -9 quintilhões)	$+2^{63}-1$ (uns 9 quintilhões)	123981723971982318273L, -12381726387613678688L, 97L, -23L, 0L (Observe o 'L' final)
float	32 bits	aprox $-10^{38}$ (6-7 dígitos significativos)	aprox $+10^{38}$ (6-7 dígitos significativos)	-3.4F 45.78E+23F (Observe o 'F' final)
double	64 bits	aprox $-10^{308}$ (15 dígitos significativos)	aprox $+10^{308}$ (15 dígitos significativos)	-3.4 45.78E+23

le void

-

-

-

indica ausência de tipo

## Conversão entre Tipos:

☛ **Implícitas:** operandos idênticos.

☐ Promoção → valor de um tipo tratado como de um tipo similar.

☐ Padrão:

```
int total = 0; // soma de graus
int grauContador = 0; // número de graus de entrada
double media; // valor do grau
//...
media = total / grauContador;
```

☛ **Explícita:** operandos diferentes.

☐ Operador de **cast** unário → cópia temporária de um operando.

☐ Notação: (tipoPrimitivo) variável.

```
double x = 8.89;
int n = (int)x; // n terá valor 8
```

## Operadores Java:

- **Operadores Binários (operações aritméticas básicas):** Soma (+), Subtração (-), multiplicação (\*), divisão (/) e Resto da Divisão (%).

- ☐ **Exemplos de Expressões:**  $3 + 5$ ;  $b \% 2$ .

- **Versões Unárias dos operadores:** mais (+) e menos (-).

- ☐ **Exemplos de Expressões:**  $-7$  e  $+5$ .

- **Operadores Binários seguidos de:** igual (=).

- ☐ **Exemplo de expressão:**

$soma += nota * peso$ ; // equivalente a  $soma = soma + nota * peso$ ;

# Introdução à Linguagem Java

## Operadores Java:

### ➤ Outros operadores:

- ❑ Operador soma (+) → sobrecarregado para concatenar *strings*.

- Exemplo de expressão:

```
String welcome = "Hello, ";  
System.out.println(welcome + "world!");
```

- ❑ Auto-incremento (++) e Auto-decremento (--).

- Exemplo de expressão:

```
int númeroDeAlunos;  
  
númeroDeAlunos++; // equivalente a númeroDeAlunos = númeroDeAlunos + 1  
númeroDeAlunos--; // equivalente a númeroDeAlunos = númeroDeAlunos - 1  
  
if(númeroDeAlunos-- > 0) // equivalente a testar númeroDeAlunos e depois decrementar  
if(--númeroDeAlunos > 0) // equivalente a decrementar e depois testar númeroDeAlunos
```

## Operadores Java:

### ☛ Relacionais:

- ☐ Menor (<), Menor ou Igual (<=), Maior (>), Maior ou Igual (>=), Igual (==) e Diferente (!=).

### ☛ Lógicos:

- ☐ AND (&&), OR (||) e NOT (!).

## Exemplos de expressões:

```
if (numeroDeAlunos > MAX_ALUNOS || numeroDeProfessores > MAX_PROFS) {...}
```

```
If (ano % 4 == 0 && ano % 100 != 0 || ano % 400 == 0) {...} // ano bissexto
```

## Operadores Java:

☛ **Bits e Deslocamento:** pesquisem e testem. **O que acontece?**

☐ **&, &=, |, |=, ^, ^=, ~**

Ex.: `int x = 13 >> 2; int c = ~20;`

☛ **Ternário:**


☐ Operação **Condicional** sem usar *if-else*

**Exemplos de expressão:**

```
// a linha seguinte
média = númeroDeNotas == 0 ? 0.0 : soma/númeroDeNotas;

// é equivalente às linhas seguintes
if(númeroDeNotas == 0) {
    média = 0.0;
} else {
    média = soma/númeroDeNotas;
}
```

## Precedências dos Operadores Java:

Operadores	Precedência
[] . ()	Mais alta
! ~ ++ -- + (unário) - (unário) (cast) new	
* / %	
+ -	
<< >> >>>	
< <= > >= instanceof	
== !=	
&	
^	
&&	
? : (ternário)	
= += -= *= /= %= &=  = ^= <<= >>= >>>=	
	Mais baixa

**Obs.:** Na dúvida, use parênteses!

## Terceiro Programa: Planejamento de Aposentadora.

```
Aposentadoria.java ✕
1 package pex0246.exemplos;
2
3 * Planejamento de aposentadoria.
4
5 import java.util.Scanner;
6
7 public class Aposentadoria {
8
9     public static void main(String[] args) {
10
11         final double QUANTIA_MINIMA_TOTAL = 10000;
12         final double MENOR_CONTRIBUICAO = 200;
13
14         System.out.print("Quanto dinheiro voce quer para se aposentar? ");
15         double alvo = recebeDoubleMaiorQue(QUANTIA_MINIMA_TOTAL);
16
17         System.out.print("Quanto dinheiro voce vai contribuir todo ano?");
18         double contribuicaoAnual = recebeDoubleMaiorQue(MENOR_CONTRIBUICAO);
19
20         System.out.print("Taxa de juros (ex.: digite 0,075 para 7,5%): ");
21         double juros = recebePercentual();
22
23         System.out.println("Voce pode se aposentar em "
24             + computaAnosDeContribuicao(alvo, contribuicaoAnual, juros)
25             + " anos.");
26     }
27 }
28
29
30
31 private static int computaAnosDeContribuicao(double alvo,
32                                             double contribuicao,
33                                             double juros) {
34     int anos = 0;
35     double saldo = 0;
36
37     while (saldo < alvo) {
38         saldo = (saldo + contribuicao) * (1 + juros);
39         anos++;
40     }
41     return anos;
42 }
43 }
```



## Terceiro Programa: Planejamento de Aposentadora.

```

44 private static double recebePercentual() {
45     Scanner sc = new Scanner(System.in);
46
47     if (!sc.hasNextDouble()) {
48         sc.next();
49         System.out.println("Voce deve digitar um valor real entre 0 e 1.");
50         return recebePercentual();
51     }
52
53     double valor = sc.nextDouble();
54
55     if (valor > 1 || valor < 0) {
56         System.out.println(" Voce deve digitar um valor real entre 0 e 1.");
57         return recebePercentual();
58     }
59     return valor;
60 }
61

```

### Saída do Programa:

Quanto dinheiro voce quer para se aposentar?  
 500.000,00  
 Quanto dinheiro voce vai contribuir todo ano?  
 300,00  
 Taxa de juros (ex.: digite 0,075 para 7,5%):  
 0,075  
 Voce pode se aposentar em 66 anos.

```

62 private static double recebeDoubleMaiorQue(double minimo) {
63
64     Scanner sc = new Scanner(System.in);
65
66     if (!sc.hasNextDouble()) {
67         sc.next();
68         System.out.println("Voce deve digitar um valor real maior que "
69             + minimo);
70         return recebeDoubleMaiorQue(minimo);
71     }
72     double valor = sc.nextDouble();
73
74     if (valor < minimo) {
75         System.out.println("Voce deve digitar um valor real maior que "
76             + minimo);
77         return recebeDoubleMaiorQue(minimo);
78     }
79     return valor;
80 }
81

```

## Quarto Programa: Ler 10 inteiros e imprime-os em ordem inversa v1.

☛ **Arrays** são sempre indexados a partir de zero.

```
Inverte1.java ✕
1 package pex0246.exemplos;
2
3 * Ler 10 números inteiros da entrada, imprimir em ordem inversa.
4
5
6
7
8
9 import java.util.Scanner;
10
11 public class Inverte1 {
12
13     public static void main(String[] args) {
14
15         Scanner sc = new Scanner(System.in);
16
17         // criação do array de 10 inteiros
18         int[] números = new int[10];
19
20         for (int i = 0; i < 10; i++) {
21             System.out.print("Entre com o proximo inteiro: ");
22             números[i] = sc.nextInt();
23         }
24
25         for (int i = 10 - 1; i >= 0; i--) {
26             System.out.print(números[i] + " ");
27         }
28     }
29 }
```

Será que este programa pode ser melhorado?

### Saída do Programa:

```
Entre com o proximo inteiro: 10
Entre com o proximo inteiro: 9
Entre com o proximo inteiro: 8
Entre com o proximo inteiro: 7
Entre com o proximo inteiro: 6
Entre com o proximo inteiro: 5
Entre com o proximo inteiro: 4
Entre com o proximo inteiro: 3
Entre com o proximo inteiro: 2
Entre com o proximo inteiro: 1
1 2 3 4 5 6 7 8 9 10
```

## Quarto Programa: Ler 10 inteiros e imprime-os em ordem inversa v2.

☛ *números.length* é a qtd. de elementos no *array* números.

```

Inverte2.java
4
40 * Ler 10 números inteiros da entrada, imprimir em ordem inversa v2
8
9 import java.util.Scanner;
10
11 public class Inverte2 {
12
13     public static void main(String[] args) {
14
15         final int NÚMEROS_A_LER = 10;
16
17         Scanner sc = new Scanner(System.in);
18
19         // criação do array de 10 inteiros
20         int[] números = new int[NÚMEROS_A_LER];
21
22         for (int i = 0; i < números.length; i++) {
23             System.out.print("Entre com o proximo inteiro: ");
24             números[i] = sc.nextInt();
25         }
26
27         for (int i = números.length - 1; i >= 0; i--) {
28             System.out.println(números[i]);
29         }
30     }
31 }
    
```

### Saída do Programa:

```

Entre com o proximo inteiro: 10
Entre com o proximo inteiro: 9
Entre com o proximo inteiro: 8
Entre com o proximo inteiro: 7
Entre com o proximo inteiro: 6
Entre com o proximo inteiro: 5
Entre com o proximo inteiro: 4
Entre com o proximo inteiro: 3
Entre com o proximo inteiro: 2
Entre com o proximo inteiro: 1
1
2
3
4
5
6
7
8
9
10
    
```

## Quinto Programa: Exibe argumentos de linha de comando.

☛ *args* é um *array* normal composto de *Strings*.

```
EcoArgumentos.java ✕
1 package pex0246.exemplos;
2
3
4 * Ecoar argumentos da linha de comando.
5
6
7
8 public class EcoArgumentos {
9
10     public static void main(String[] args) {
11
12         for (int i = 0; i < args.length; i++) {
13             System.out.print(args[i] + " ");
14         }
15
16         System.out.println();
17
18         for (int i = 0; i < args.length; i++) {
19             System.out.println(args[i]);
20         }
21     }
22 }
```

### Execução do Programa:

```
java EcoArgumentos Prof. Lenardo Chaves e Silva
```

### Saída do Programa:

```
Prof. Lenardo Chaves e Silva
Prof.
Lenardo
Chaves
e
Silva
```

## Sexto Programa: Mostra uma forma de inicializar *arrays*.

➤ Use *System.err* em vez de *System.out* para imprimir erros.

➤ *System.exit(0)* é usado para terminar o programa “de boa”.

➤ *System.exit(1)* é usado para terminar o programa “de mal”.

### Execução do Programa:

```
java Dia 3
```

### Saída do Programa:

```
Terca
```

Obs.: Mais informações sobre E/S em Java [aqui](#).

```

1 package pex0246.exemplos;
2
3 * Imprime o dia da semana correspondente ao argumento de linha de comando.
4
5 public class Dia {
6
7     public static void main(String[] args) {
8
9         final int DIAS_NA_SEMANA = 7;
10        final String[] diasDaSemana = { "", "Domingo", "Segunda",
11            "Terca", "Quarta", "Quinta", "Sexta", "Sabado" };
12
13        if (args.length != 1) {
14            System.err.println("Sintaxe: Dia numero");
15            System.exit(1);
16        }
17
18        int dia = Integer.parseInt(args[0]);
19
20        if (dia < 1 || dia > DIAS_NA_SEMANA) {
21            System.err.println("O dia da semana deve estar entre 1 e " +
22                DIAS_NA_SEMANA);
23            System.exit(1);
24        }
25
26        System.out.println(diasDaSemana[dia]);
27    }
28 }

```

# Exercícios de Fixação - Java



1. Quando você compila um programa escrito na linguagem de programação Java, o compilador converte o arquivo fonte legível por nós humanos em código independente de plataforma que uma JVM pode entender. Como este código independente de plataforma é denominado?
2. Qual das seguintes opções não apresenta um comentário válido:
  - a. `/** comentário */`
  - b. `/* comentário */`
  - c. `/* comentário`
  - d. `// comentário`
3. Qual é a primeira coisa que você deve verificar se você ver o seguinte erro em tempo de execução?

**Exceção no segmento `java.lang.NoClassDefFoundError`  
"principal": HelloWorldApp.java.**

4. Qual é a assinatura correta do método **main**?
5. Ao declarar o método **main**, que modificador deve vir em primeiro lugar, **public** ou **static**?
6. Que parâmetros o método **main** define?

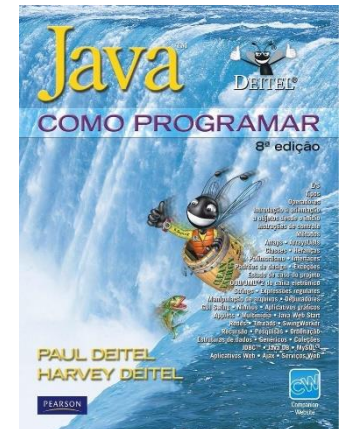
**Extra:** O programa **HelloWorld2.java** tem um erro. Corrija o erro para que o programa compile e seja executado com sucesso. Qual era o erro?

```
1 package pex0246.exemplos;
2
3 public class Hello2 {
4
5     /**
6      * The HelloWorldApp class implements an application that
7      * simply prints "Hello World!" to standard output.
8      */
9     public static void main(String[] args) {
10         System.out.println("Hello World!"); // Display the string.
11     }
12 }
```

# Referências



- ☞ DEITEL, P.; DEITEL, H. Java: como programar. 6/8. ed. São Paulo: Pearson, 2010. 1176p. Capítulos 2.
- ☞ ORACLE. About the Java Technology. Disponível em: <<http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>>. Acesso em: Jul. 2016.
- ☞ ORACLE. Lesson: Common Problems (and Their Solutions). <<http://docs.oracle.com/javase/tutorial/getStarted/problems/index.html>>. Acesso em: Jul. 2016.
- ☞ ORACLE. Lesson: Basic I/O. Disponível em: <<http://docs.oracle.com/javase/tutorial/essential/io/>>. Acesso em: Jul. 2016.
- ☞ ORACLE. Scanning and Formatting. Disponível em: <<http://docs.oracle.com/javase/tutorial/essential/io/scanfor.html>>. Acesso em: Jul. 2016.





# Próxima Aula...

## ➤ Aula 04:

☐ **POO:** Criação de Classes em Java.



Importante! *Você* é responsável por aprender detalhes adicionais da linguagem Java. O que faremos em aula não é uma cobertura completa da linguagem.