

# Java: aspectos básicos

Laboratório de Programação (5COP011)

Prof. Bruno Bogaz Zarpelão

Departamento de Computação - 2015



UNIVERSIDADE  
ESTADUAL DE LONDRINA

# Objetivo

- Aprender estruturas básicas da linguagem Java:
  - compilação usando linha de comando;
  - método main;
  - declaração de variáveis;
  - tipos primitivos;
  - operadores;
  - estruturas de controle;

# Compilação e execução em Java

- Os programas em Java são executados por uma máquina virtual, a JVM (Java Virtual Machine).
- Para entender melhor esse processo, vamos compilar e executar um programa Java fora do IDE.

# Hello World em Java

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

# Hello World em Java

- Compilando o código fonte:
  - `javac <Caminho do arquivo>/OlaMundo.java`
- Será gerado um arquivo `OlaMundo.class`
- Executando o programa:
  - `java OlaMundo`

# Método `main`

- O `main` é um método especial que serve como ponto de partida da execução do programa em Java.
- O `main` precisa ser `public`, `static`, `void` e ter um vetor de `Strings` como parâmetro.

# Método main

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

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

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

```
public static void (String[] params)
```

# Parâmetros do método `main`

- Os parâmetros do método `main` são passados na linha de comando da execução do programa:

```
class Programa{  
    public static void main (String args[]){  
        for (int i = 0; i<args.length; i++){  
            System.out.println(args[i]);  
        }  
    }  
}
```

```
java Programa Parametro1 Parametro2 Parametro3
```



# Variáveis

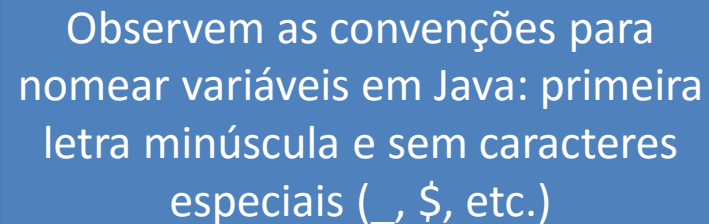
- Variáveis podem guardar dados de vários tipos: números, textos, booleanos, referências a objetos, etc.
- Declaração de variável em Java:
  - Definir nome, tipo e escopo.

# Variáveis

```
int numeroDaConta;
```

```
int numeroDaConta = 0;
```

```
double precoDoProduto;
```



Observem as convenções para nomear variáveis em Java: primeira letra minúscula e sem caracteres especiais (\_, \$, etc.)

# Variáveis

- Se declararmos duas variáveis com o mesmo nome em um mesmo bloco (escopo), o código não compilará.
- A variável deve ser inicializada antes de sua primeira inicialização.

# Tipos primitivos

Tipo	Descrição	Tamanho
byte	Valor inteiro entre -128 e 127 (inclusivo)	1 byte
short	Valor inteiro entre -32.768 e 32.767 (inclusivo)	2 bytes
int	Valor inteiro entre -2.147.483.648 e 2.147.483.647 (inclusivo)	4 bytes
long	Valor inteiro entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807 (inclusivo)	8 bytes

# Tipos primitivos

Tipo	Descrição	Tamanho
float	Valor com ponto flutuante entre $1,40129846432481707 \times 10^{-45}$ e $3,40282346638528860 \times 10^{38}$ (positivo ou negativo)	4 bytes
double	Valor com ponto flutuante entre $4,94065645841246544 \times 10^{-324}$ e $1,79769313486231570 \times 10^{308}$ (positivo ou negativo)	8 bytes
boolean	true ou false	1 bit
char	Um único caractere Unicode de 16 bits.	2 bytes

# Tipos primitivos

- Para armazenar texto, deve-se recorrer ao “tipo” String.
- Importante: String não é um tipo primitivo!

# Operadores aritméticos

- Soma: +
- Subtração: –
- Multiplicação: \*
- Divisão: /
- Resto da divisão: %

# Operadores de atribuição

- Simples: =
- Incremental: +=
- Decremental: -=
- Multiplicativa: \*=
- Divisória: /=
- Modular: %=



# Operadores de atribuição

```
int valor = 1; //valor == 1
valor += 2; //valor == 3
valor -= 1; //valor == 2
valor *= 2; //valor == 4
valor /= 2; //valor == 2
valor %= 2; //valor == 0
```



```
int valor = 1; //valor == 1
valor = valor + 2; //valor == 3
valor = valor - 1; //valor == 2
valor = valor*2; //valor == 4
valor = valor/2; //valor == 2
valor = valor%2; //valor == 0
```

# Operadores relacionais

- Igualdade: ==
- Diferença: !=
- Menor: <
- Menor ou igual: <=
- Maior: >
- Maior ou igual: >=

# Operadores lógicos

- “E” lógico: &&
- “OU” lógico: ||

# Se...então

```
if (preco < 0) {  
    System.out.println("O preço do produto não pode ser negativo");  
} else {  
    System.out.println("Produto cadastrado com sucesso");  
}
```

# Enquanto...faça

```
int contador = 0;
while(contador < 100){
    System.out.println("Bom dia!");
    contador++;
}
```

# Para...

```
for (int contador = 0; contador < 100; contador++) {  
    System.out.println("Bom dia!");  
}
```

# Ler *input* do teclado

- Utilizar a classe Scanner do pacote java.util:

```
public static void main(String[] args)
{
    int    a;
    float  b;
    String c;
    Scanner sc = new Scanner(System.in);
    System.out.println("Inteiro:");
    a = sc.nextInt();
    System.out.println("Float");
    b = sc.nextFloat();
    System.out.println("String");
    c = sc.next();
    System.out.println("Mostrando inputs:");
    System.out.println("\tInteiro: " + a);
    System.out.println("\tFloat:   " + b);
    System.out.println("\tString:  " + c);
}
```