

LÓGICA DE PROGRAMAÇÃO COM JAVA

Revisitando os conceitos, agora em outra linguagem

A estrutura de um programa Java

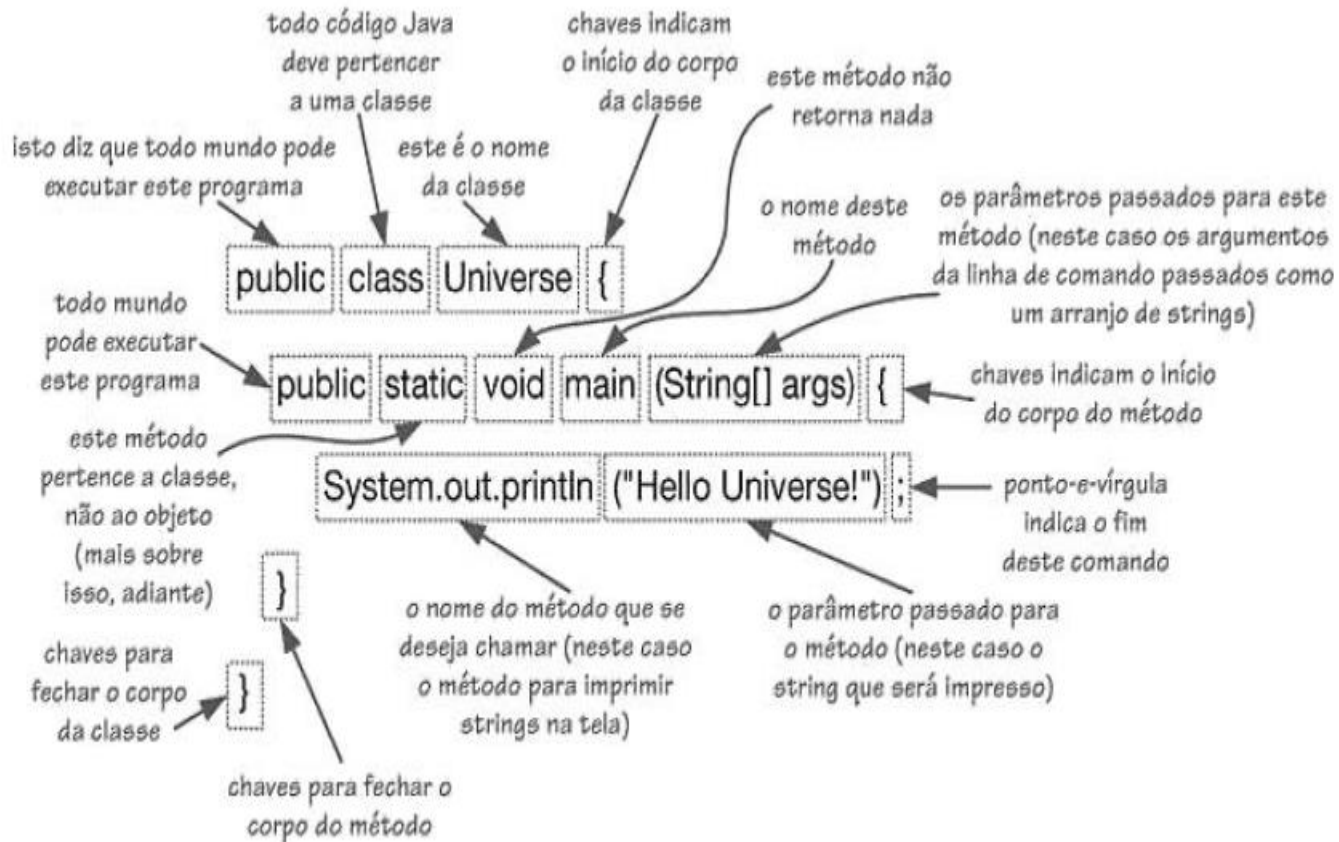


Figura 1.1 O programa "Hello Universe!"

Declaração de variáveis

- Dentro de um bloco, podemos declarar variáveis e usá-las. Em Java, toda variável tem um tipo que não pode ser mudado, uma vez que declarado:

tipoDaVariavel nomeDaVariavel;

- Por exemplo, é possível ter uma variável idade que guarda um número inteiro:

int idade;

- Com isso, você declara a variável idade, que passa a existir a partir daquela linha. Ela é do tipo int, que guarda um número inteiro. A partir daí, você pode usá-la, primeiramente atribuindo valores. A linha a seguir é a tradução de: "idade **deve valer quinze**".

idade = 15;

Declaração de variáveis

```
class TesteIdade {  
  
    public static void main(String[] args) {  
        // imprime a idade  
        int idade = 20;  
        System.out.println(idade);  
  
        // gera uma idade no ano seguinte  
        int idadeNoAnoQueVem;  
        idadeNoAnoQueVem = idade + 1;  
  
        // imprime a idade  
        System.out.println(idadeNoAnoQueVem);  
    }  
}
```

TIPO	TAMANHO
boolean	1 bit
byte	1 byte
short	2 bytes
char	2 bytes
int	4 bytes
float	4 bytes
long	8 bytes
double	8 bytes

Os tipos primitivos de variáveis

Declaração de variáveis

Tipos não primitivos

TIPOS NÃO PRIMITIVOS

Logo conheceremos outros tipos (não primitivos) da linguagem Java, mas desde já repare que um texto, assim como qualquer outro valor, também tem um tipo! Esse tipo é conhecido por `String`.

Você poderia, no lugar de fazer:

```
System.out.println("Eu sou uma String");
```

Declarar da seguinte forma:

```
String texto = "Eu sou uma String";  
System.out.println(texto);|
```

Declaração de variáveis

```
public class CalculadoraDeEstoque {  
  
    public static void main(String[] args) {  
  
        double livroJava8 = 59.90;  
        double livroTDD = 59.90;  
  
        double soma = livroJava8 + livroTDD;  
  
        System.out.println("O total em estoque é "+ soma);  
    }  
}
```

Qual será a saída deste programa?

Declaração de variáveis - Casting

```
public class CalculadoraDeEstoque {  
  
    public static void main(String[] args) {  
  
        int livroJava8 = 59.90;  
        int livroTDD = 59.90;  
  
        int soma = livroJava8 + livroTDD;  
  
        System.out.println("O total em estoque é "+ soma);  
    }  
}
```

Troque o tipo **double** do código anterior para **int** e execute o programa.

Rodou?

Declaração de variáveis - Casting

```
public class CalculadoraDeEstoque {  
  
    public static void main(String[] args) {  
  
        int livroJava8 = 59.90;  
        int livroTDD = 59.90;  
  
        int soma = livroJava8 + livroTDD;  
  
        System.out.println("O total em estoque é " + soma);  
    }  
}
```

Há algo de errado nesse trecho de código?

Type mismatch: cannot convert from double to int

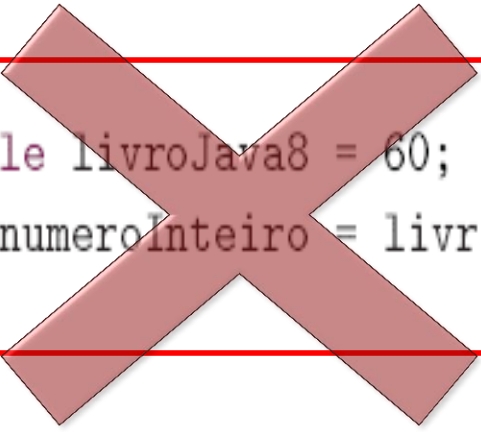
Declaração de variáveis - Casting

```
double livroJava8 = 60;  
int numeroInteiro = livroJava8;
```

Nem esse código compilará!

Declaração de variáveis - Casting

- Casting é a ação de forçar a conversão de tipos de variáveis:
 - Ex: A variável foi declarada como Double e por alguma necessidade precisa ser reconhecida como Int



```
double livroJava8 = 60;  
int numeroInteiro = livroJava8;
```

```
double livroJava8 = 60;  
int numeroInteiro = (int) livroJava8;
```

Agora sim o código será compilado

Operadores básicos

Operador	Função
+	soma
-	subtração
*	multiplicação
/	divisão
%	modulo

Estrutura condicional: o comando IF

```
if (condicaoBooleana) {  
    codigo;  
}
```

Uma **condição booleana** é qualquer expressão que retorne true ou false. Para isso, você pode usar os operadores <, >, <=, >= e outros. Um exemplo:

```
int idade = 15;  
if (idade < 18) {  
    System.out.println("Não pode entrar");  
}
```

Estrutura condicional: o comando IF

Além disso, você pode usar a cláusula `else` para indicar o comportamento que deve ser executado no caso da expressão booleana ser falsa:

```
int idade = 15;
if (idade < 18) {
    System.out.println("Não pode entrar");
} else {
    System.out.println("Pode entrar");
}
```

Estrutura condicional: o comando IF

Você pode concatenar expressões booleanas através dos operadores lógicos "E" e "OU". O "E" é representado pelo `&&` e o "OU" é representado pelo `||`.

Um exemplo seria verificar se ele tem menos de 18 anos e se ele não é amigo do dono:

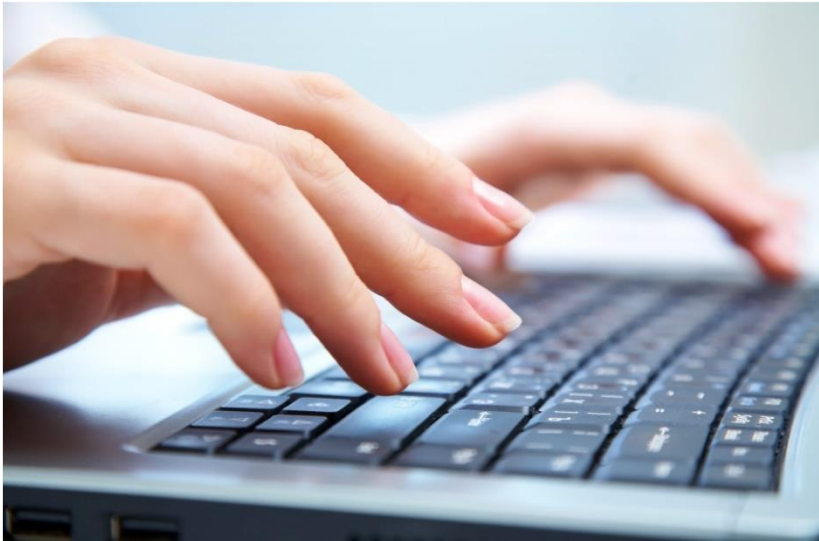
```
int idade = 15;
boolean amigoDoDono = true;
if (idade < 18 && amigoDoDono == false) {
    System.out.println("Não pode entrar");
}
else {
    System.out.println("Pode entrar");
}
```

Estrutura condicional: o comando IF

Esse código poderia ficar ainda mais legível, utilizando-se o operador de negação, o `!`. Esse operador transforma o resultado de uma expressão booleana de `false` para `true` e vice versa.

```
1 int idade = 15;
2 boolean amigoDoDono = true;
3 if (idade < 18 && !amigoDoDono) {
4     System.out.println("Não pode entrar");
5 }
6 else {
7     System.out.println("Pode entrar");
8 }
```

Estrutura condicional: o comando IF – Vamos Digitar!



```
//Usando a API do Java
import java.util.Scanner;

public class ProgramaParOuImpar {

    public static void main (String args[]){
        //Variavel para receber o dado do usuario
        int numeroEntrada;

        //Instanciando a classe Scanner para receber uma entrada de usuario
        Scanner entrada = new Scanner(System.in);

        //Mensagem para o usuario
        System.out.println("Digite um número para saber se é par ou ímpar:");
        //Fazendo com a que variavel numeroEntrada receba o valor digitado
        numeroEntrada = entrada.nextInt();

        //Bloco condicional para checar se e par ou impar
        if (numeroEntrada % 2 == 0) {
            System.out.println("O seu número é Par!");
        } else {
            System.out.println("O seu número é Ímpar!");
        }

        entrada.close();
    }
}
```


Exemplo 1

```
public class VerificacaoCompra {

    //Declaração da função/método principal: main
    public static void main(String[] args) {

        //Declarando duas variáveis
        double valorCompra1;
        double valorCompra2;

        //Atribuindo valor nas variáveis
        valorCompra1 = 89.99;
        valorCompra2 = 70.50;

        //Declarando outra variável e inicializando em seguida
        double somaCompra = valorCompra1 + valorCompra2;

        //Decisão a ser tomada para mensagem na tela (desvio condicional)
        if(somaCompra<=100){
            //Mensagem na tela concatenando texto
            System.out.println("O valor R$" +somaCompra + " está abaixo do mínimo. Compre mais produtos.");
        }else{
            System.out.println("O valor R$" +somaCompra + " está liberado para pagamento.");
        }
    }
} //fim do método main
} //fim da classe
```

```
//Usando a API do Java
import java.util.Scanner;
```

```
public class ProgramaParOuImpar {
```

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

```
        //Variavel para receber o dado do usuario
```

```
        int numeroEntrada;
```

```
        //Instanciando a classe Scanner para receber uma entrada de usuario
```

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

```
        //Mensagem para o usuario
```

```
        System.out.println("Digite um número para saber se é par ou ímpar:");
```

```
        //Fazendo com a que variavel numeroEntrada receba o valor digitado
```

```
        numeroEntrada = entrada.nextInt();
```

```
        //Bloco condicional para checar se e par ou impar
```

```
        if (numeroEntrada % 2 == 0) {
```

```
            System.out.println("O seu número é Par!");
```

```
        } else {
```

```
            System.out.println("O seu número é Ímpar!");
```

```
        }
```

```
        entrada.close();
```

```
    }
```

```
}
```

Exemplo 2

Mudando de assunto rapidinho

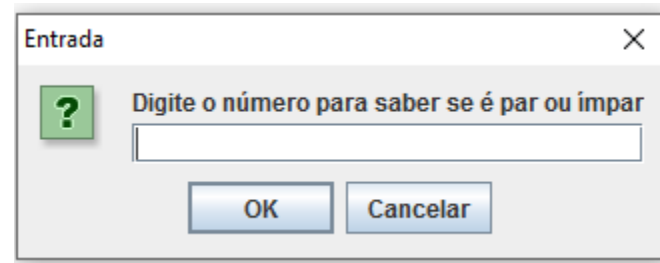
```
/*Importando componentes SWING da API do Java  
para exibir caixas de texto*/  
import javax.swing.JOptionPane;
```

- Comando import: fazer uso das classes da API do Java ou outros pacotes do programa.
- JOptionPane → classe (da API do Java) que permite a criação de caixas de dialogo para a entrada de dados ou exibição.
- Documentação oficial:
<http://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>

Mudando de assunto rapidinho

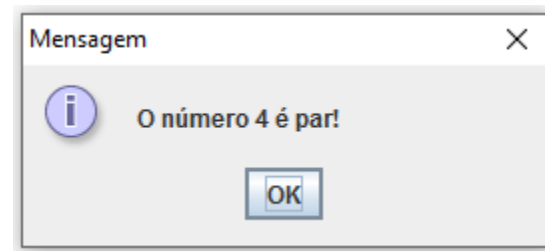
```
String entradaValor;  
//Lendo o valor do teclado  
entradaValor = JOptionPane.showInputDialog("Digite o número para saber se é par ou ímpar");
```

Comando (método) de JOptionPane para exibir uma caixa de mensagem solicitando a digitação de um número.



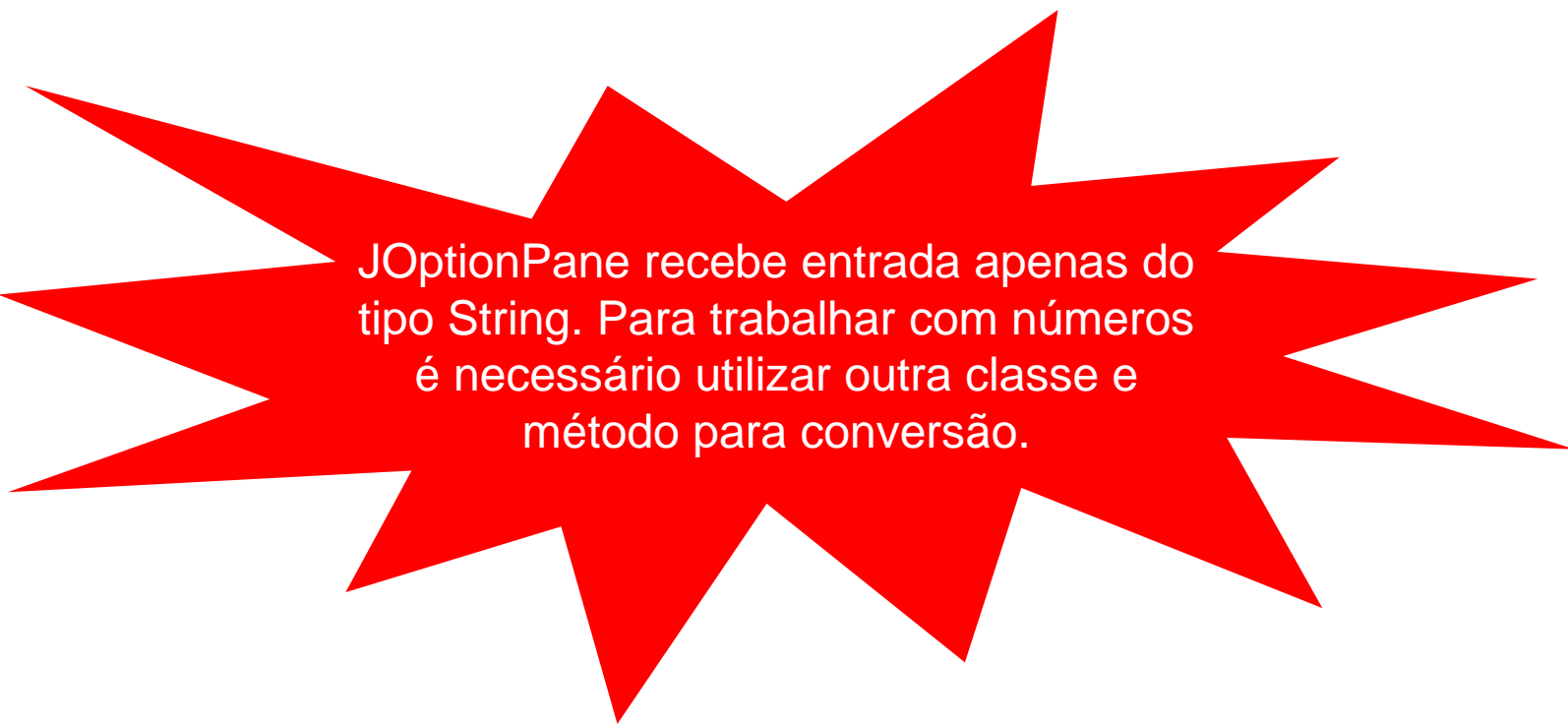
```
JOptionPane.showMessageDialog(null, "O número " + numero + " é par!");
```

Comando (método) de JOptionPane para exibir uma caixa de mensagem simples com texto informativo.



Mudando de assunto rapidinho

```
String entradaValor;  
//Lendo o valor do teclado  
entradaValor = JOptionPane.showInputDialog("Digite o número para saber se é par ou ímpar");
```



JOptionPane recebe entrada apenas do tipo String. Para trabalhar com números é necessário utilizar outra classe e método para conversão.

Mudando de assunto rapidinho

```
String entradaValor;  
//Lendo o valor do teclado  
entradaValor = JOptionPane.showInputDialog("Digite o número para saber se é par ou ímpar");  
  
/*Porém todo valor lido por JOptionPane é  
 * do tipo String. É necessário converter  
 * para número inteiro.*/  
int numero = Integer.parseInt(entradaValor);
```

Para tipo Double: `Double.parseDouble(valor)`

Para tipo Float: `Float.parseFloat(valor)`

Exemplo 2 – outra forma de entrada de dados

```
//Usando a API Java JOptionPane
import javax.swing.JOptionPane;

public class ProgramaParOuImpar {

    public static void main(String[] args) {
        /*Declarando variável para receber entrada
        *do teclado*/
        String entradaValor;
        //Lendo o valor do teclado
        entradaValor = JOptionPane.showInputDialog("Digite o número para saber se é par ou ímpar");

        /*Porém todo valor lido por JOptionPane é
        * do tipo String. É necessário converter
        * para número inteiro.*/
        int numero = Integer.parseInt(entradaValor);

        //Verificando se é par ou ímpar
        if (numero % 2 == 0){
            JOptionPane.showMessageDialog(null, "O número " +numero + " é par!");
        }else{
            JOptionPane.showMessageDialog(null, "O número " +numero + " é ímpar!");
        }
    }
}

//fim método main
//fim da classe
```

```
public class CalculadoraDeEstoque {
```

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

```
        double livroJava8 = 59.90;
```

```
        double livroTDD = 59.90;
```

```
        double soma = livroJava8 + livroTDD;
```

```
        System.out.println("O total em estoque é "+ soma);
```

```
        if (soma < 150) {
```

```
            System.out.println("Seu estoque está muito baixo!");
```

```
        } else if (soma >= 2000) {
```

```
            System.out.println("Seu estoque está muito alto!");
```

```
        } else {
```


```
            System.out.println("Seu estoque está bom");
```

```
        }
```

```
    }
```

```
}
```

Exemplo 3



**Desvios
condicionais
aninhados**

Estrutura de repetição: o comando While

```
public class ImprimeNumero {  
  
    public static void main(String[] args) {  
  
        //Declarando a variável  
        int numeroInteiro = 0;  
  
        //Imprimindo na tela de 0 a 100  
        while(numeroInteiro < 101){  
            System.out.println(numeroInteiro);  
            /*Contador que vai incrementar em 1  
             * o valor de numeroInteiro*/  
            numeroInteiro = numeroInteiro + 1;  
        }  
    }  
} //fim do método main  
} //fim da classe
```

Condição para inicialização do comando while. Se verdadeiro o trecho entre { } será executado

Imprescindível o uso de uma variável de controle, normalmente chamada de contador

Estrutura de repetição: o comando FOR

- Outra forma de realizar loops controlados, mas com uma estrutura mais compacta

```
for (inicialização; condição; atualização) {  
    Bloco de comando  
}
```

Estrutura de repetição: o comando FOR

```
public class ImprimeNumeroFor {  
  
    public static void main(String[] args) {  
  
        //Declarando a variável  
        int numeroInteiro;  
  
        for(numeroInteiro = 0; numeroInteiro < 101; numeroInteiro++){  
            System.out.println(numeroInteiro);  
        }  
    } //fim do método main  
} //fim da classe
```

Desbravando Java e Orientação a Objetos

Um guia para o iniciante da linguagem



Casa do Código

RODRIGO TURINI

PETER JANDL JUNIOR

3ª Edição
revisada e ampliada

JAVA

GUIA DO PROGRAMADOR

ATUALIZADO PARA JAVA 8

novatec

Programação Orientada a Objetos com JAVA

David J. Barnes · Michael Kölling

Uma introdução prática usando o BLUEJ



Prefácio de James Gosling

PEARSON
Prentice Hall



Pense Java - Guia de Aprendizagem

Tradução da 2ª Edição
Cobre Java 5.0

Use a Cabeça!

Java

Aprenda como os segmentos (threads) podem mudar sua vida



Evite erros embaraçosos de OO

Rote a cabeça para funcionar com os 42 quebra-cabeças em Java



Faça os conceitos de Java se incorporarem à sua mente



Crie GUIs atraentes e úteis

ATA

Kathy Sierra & Bert Bates



**KEEP
CALM
AND
HAVE
A BREAK**