Decode Team

Instituto Politécnico de tomar

Portugol IDE

Equivalências de estruturas

Contents

[Java 2](#_Toc356908038)

[Variáveis 2](#_Toc356908039)

[Tipos de variáveis 2](#_Toc356908040)

[Definição de variáveis 2](#_Toc356908041)

[Estruturas input/output 3](#_Toc356908042)

[Input – Ler 3](#_Toc356908043)

[Output – Escrever 4](#_Toc356908044)

[Estrutura *Início* 4](#_Toc356908045)

[Estrutura *Fim* 5](#_Toc356908046)

[Estruturas de Decisão 5](#_Toc356908047)

[Condição if 5](#_Toc356908048)

[Condição if else 6](#_Toc356908049)

[Condição while 6](#_Toc356908050)

[Condição do while 7](#_Toc356908051)

[Funções 7](#_Toc356908052)

[Definir funções 7](#_Toc356908053)

[Chamada de funções 7](#_Toc356908054)

[Operadores 8](#_Toc356908055)

[Aritméticos 8](#_Toc356908056)

[Lógicos 8](#_Toc356908057)

[Relacionais 9](#_Toc356908058)

Portugol

Equivalências de código às estruturas de fluxograma

# Java

Algumas notas sobre *Java*:

* É case sensitive.
* Usa o ponto e virgula (;) para terminar uma linha de código.
* As funções podem ser definidas antes ou depois no main.

## Variáveis

### Tipos de variáveis

* Int
* Double
* String
* Char
* Float
* Long
* Short
* Byte
* Boolean

### Definição de variáveis

variável <- 0

* Int
  + Int variavel=0;
* Double
  + Double variavel =5.3;
* String
  + char[] variavel = {'C', 'H', 'A', 'R'};
  + String variavel\_1 = new String (variavel);
  + **Nota**: String é um array de caracteres
* Char
  + Char variavel =’X’;
* Float
  + Float variavel =100.9F
  + **Nota**: O tipo de dados float termina com F
* Long
  + Long variavel =1000000L
  + **Nota**: O tipo de dados long termina com L
* Short
  + Short variavel =100;
* Byte
  + Byte variavel =10;
* Boolean
  + Boolean variavel =false;

## Estruturas input/output

### Input – Ler

variavel

Para ler do teclado, primeiro é necessário é fazer o import da biblioteca java.util.Scanner

import java.util.Scanner;

Depois é necessário criar um objecto do tipo *Scanner*:

Scanner scanner = new Scanner(System.in);

De seguida podemos fazer a leitura propriamente dita.

Para ler uma string:

variavel=scanner.nextLine();

Para ler um inteiro:

variavel=scanner.nextInt();

Para ler um double:

variavel=scanner.nextDouble();

### Output – Escrever

mensagem

Para escrever no ecrã existem duas formas:

Após escrever mensagem permanece na mesma linha

System.out.print("mensagem");

Após escrever mensagem permanece na mesma linha

System.out.println("mensagem ");

Para escrever no ecrã a mensagem, juntamente com o valor de uma variável:

System.out.print("Inserir texto aqui" + variável);

## Estrutura *Início*

Início

Início de uma classe:

public class exemplo {

Início do *main*

public static void main(String[] args) {

## Estrutura *Fim*

Fim

}

## Estruturas de Decisão

### Condição if

true

condição

If (*condição*) {

Instruções

}

**Nota**: A condição deve estar dentro de parêntesis

### Condição if else

condição

false

true

If (*condição*) {

Instruções

}else {

Instruções

}

**Nota**: A condição deve estar dentro de parêntesis

### Condição while

condição

false

instruções

true

While (*condição*){

Instruções

}

**Nota**: A condição deve estar dentro de parêntesis

### Condição do while

instruções

true

condição

false

do{

Instruções

}while(*condição*);

**Nota**: A condição deve estar dentro de parêntesis

## Funções

### Definir funções

Definir função *Exemplo*:

public static void Exemplo() {

}

### Chamada de funções

Chamada da função *Exemplo*:

Exemplo();

As funções podem também ser de vários tipos e receber parâmetros.

Temos como exemplo uma função do tipo inteiro que recebe o parâmetro *k*:

public static int fact(int k) {

instruções;

}

## Operadores

### Aritméticos

* Adição
  + +
* Subtração
  + -
* Divisão
  + /
* Multiplicação
  + \*
* Resto da divisão inteira
  + %
* Potenciação
* Concatenação de texto
  + +

### Lógicos

* Disjunção
  + &&
* Conjunção
  + ||
* Conjunção Exclusiva
* Negação
  + !

### Relacionais

* Igual
  + ==
* Diferente
  + !=
* Maior
  + >
* Maior ou igual
  + >=
* Menor
  + <
* Menor ou igual
  + <=