Decode Team

Instituto Politécnico de tomar

Portugol IDE

Equivalências de estruturas

Índice

[Java 3](#_Toc356914900)

[Variáveis 3](#_Toc356914901)

[Tipos de variáveis 3](#_Toc356914902)

[Definição e atribuição de variáveis 3](#_Toc356914903)

[Estruturas input/output 4](#_Toc356914904)

[Input – Ler 4](#_Toc356914905)

[Output – Escrever 5](#_Toc356914906)

[Estrutura *Início* 5](#_Toc356914907)

[Estrutura *Fim* 6](#_Toc356914908)

[Estruturas de Decisão 6](#_Toc356914909)

[Condição if 6](#_Toc356914910)

[Condição if else 7](#_Toc356914911)

[Condição while 7](#_Toc356914912)

[Condição do while 8](#_Toc356914913)

[Funções 8](#_Toc356914914)

[Definir funções 8](#_Toc356914915)

[Chamada de funções 9](#_Toc356914916)

[Operadores 9](#_Toc356914917)

[Aritméticos 9](#_Toc356914918)

[Lógicos 10](#_Toc356914919)

[Relacionais 10](#_Toc356914920)

[C 11](#_Toc356914921)

[Variáveis 11](#_Toc356914922)

[Tipos de variáveis 11](#_Toc356914923)

[Definição e atribuição de variáveis 11](#_Toc356914924)

[Estruturas input/output 11](#_Toc356914925)

[Input – Ler 12](#_Toc356914926)

[Output – Escrever 12](#_Toc356914927)

[Estrutura *Início* 13](#_Toc356914928)

[Estrutura *Fim* 13](#_Toc356914929)

[Estruturas de Decisão 13](#_Toc356914930)

[Condição if 13](#_Toc356914931)

[Condição if else 14](#_Toc356914932)

[Condição while 14](#_Toc356914933)

[Condição do while 15](#_Toc356914934)

[Funções 15](#_Toc356914935)

[Definir funções 15](#_Toc356914936)

[Chamada de funções 16](#_Toc356914937)

[Operadores 16](#_Toc356914938)

[Aritméticos 16](#_Toc356914939)

[Lógicos 16](#_Toc356914940)

[Relacionais 17](#_Toc356914941)

# 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 e atribuição de variáveis

variável <- valor

* Int
  + Int variavel=valor;
  + **Nota**: *valor* é um número inteiro. Ex: 1
* Double
  + Double variavel =valor;
  + **Nota**: *valor* é um número decimal. Ex: 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’;
  + **Nota**: *X* é um caracter e deve estar dentro de pelicas.
* Float
  + Float variavel =valorF;
  + **Nota**: *valor* é um decimal.
  + O tipo de dados float termina com F ou f
* Long
  + Long variavel =valorL;
  + **Nota**: *valor* é um número inteiro.
  + O tipo de dados long termina com L ou l
* Boolean
  + Boolean variavel =false;
  + **Nota:** Este tipo de dados pode assumer o valor *true* ou *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

Para fechar uma classe ou o main:

}

## 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* sem parâmetros de entrada:

public static void Exemplo() {

}

Definir função *Exemplo* do tipo int com um parâmetro de entrada:

**Nota 1:** as funções podem ter ou não tipos.

**Nota 2:** O tipo dos parâmetros de entrada devem ser definidos.

public static int Exemplo(int k) {

}

### Chamada de funções

Chamada da função *Exemplo*:

Exemplo();

## 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
  + <=

# C

Algumas notas sobre *C*:

* É case sensitive.

## Variáveis

### Tipos de variáveis

Tipos de Variáveis:

* Inteiros:
  + short int
  + int
  + long int
* Decimais:
  + float
  + short double
  + double
  + long double
* Texto
  + char

### Definição e atribuição de variáveis

Variável <- valor

* Tipo int
  + int variavel=valor;
  + **Nota:** *valor* deve ser um número inteiro. Ex: 0
* Tipo char
  + char variavel=;
  + **Nota:** *valor* deve ser
* Tipo float
  + float variavel=valorf;
  + **Nota:** *valor* deve ser um número decimal. Ex: 0.0

## Estruturas input/output

Deve ter se em atenção a chamada da biblioteca stdio.h quando se usa input/output:

#include <stdio.h>

### Input – Ler

variavel

scanf (“%tipo\_var”,&variavel);

exemplo: scanf (“%d”, &i);

* Tipo *d*:
  + short int
  + int
* Tipo *f*:
  + long int
  + float
* Tipo *if*:
  + short double
  + double
* Tipo *c*:
  + long double
  + char

### Output – Escrever

mensagem

Printf(“%tipo\_var”, variavel);

exemplo: Printf (“%d”, &variavel);

* Tipo *d*:
  + short int
  + int
* Tipo *f*:
  + long int
  + float
* Tipo *if*:
  + short double
  + double
* Tipo *c*:
  + long double
  + char

## Estrutura *Início*

Início

Início de uma classe:

Início do *main*

void main()

{

## Estrutura *Fim*

Fim

Para fechar uma classe ou o main:

}

## 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

true

instruções

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* sem parâmetros de entrada:

int exemplo() {

}

Definir função *exemplo* com um parâmetro de entrada:

int exemplo(int k){

}

Definir função *exemplo* com dois parâmetros de entrada:

int exemplo(int k, int j){

}

### Chamada de funções

Chamada da função *exemplo* com parâmetro *k*:

exemplo(k);

## 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
  + <=