**Anotações – Java 2024**

Auxilio: Curso mais didático e completo de Java e OO, UML, JDBC, JavaFX, Spring Boot, JPA, Hibernate, MySQL, MongoDB e muito mais.

Plataforma: Udemy

Professor: Nelio Alves

Algoritmo

Sequência finita de instruções para se resolver um problema.

Exemplo:

Problema: lavar roupa suja

Algoritmo:

1. Colocar a roupa em um recipiente;
2. Colocar um pouco de sabão e amaciante;
3. Encher de água;
4. Mexer tudo até dissolver todo o sabão;
5. Deixar de molho por vinte minutos;
6. Esfregar a roupa;
7. Enxaguar;
8. Torcer.

Automação

Consiste em utilizar máquina(s) para executar o procedimento desejado de forma automática ou semiautomática.

Computador

Hardware – parte física (a máquina em si)

Software – parte lógica (programas)

* Sistema operacional (Windows, Linux, iOS);
* Aplicativos (apps de escritório, app de câmera, navegador web);
* Jogos;
* Utilitários (Antivírus, compactador de arquivos);
* Outros.

Programa > Algoritmo

Programas de computador **são algoritmos** executados pelo computador (em linhas gerais).

Conclusão: o computador é uma máquina de **automatiza** a execução de **algoritmos**.

Qualquer algoritmo? Não. Apenas algoritmos computacionais:

* Processamento de dados;
* Cálculos.

Linguagem de programação, léxica, sintática.

Linguagem de programação:

É um conjunto de regras **léxicas** (ortografia) e **sintáticas** (gramática) para se escrever programas.

Léxica

Diz respeito à correção das palavras “isoladas” (ortografia).

Exemplo (Português):

* cachorro – correto
* caxorro – errado

Exemplo (Linguagem de programação):

* main – correto
* maim – errado

Sintática

Diz respeito à correção das **sentenças** (gramática).

Exemplo (Português):

* O cachorro está com fome - certo
* A cachorro está com fome – errado

Exemplo (Linguagem de programação):

* x = 2 + y; - certo
* x = + 2 y; - errado

Exemplo de um programa:

Suponha um programa que solicita do usuário dois números e depois mostra na média aritmética deles:

*Digite o primeiro numero: 3*

*Digite o segundo numero: 6*

*Media = 4.5*

Solução em linguagem C

*#include <stdio.h>*

*int main() {*

*double x, y, media;*

*printf(“Digite o primeiro numero: “);*

*scanf(“%1f”, &x);*

*printf(“Digite o segundo numero: “);*

*scanf(“%1f”, &y);*

*media = (x + y) / 2.0;*

*printf(“Media = %.1f\n”, media);*

*return 0 ;*

*}*

Solução em linguagem C++

*#include <iostream>*

*using namespace std;*

*int main() {*

*double x, y, media;*

*cout << “Digite o primeiro numero: “;*

*cout >> x;*

*cin >> y;*

*media = (x + y) / 2.0;*

*cout << “Media = “ << media << endl;*

*return 0;*

*}*

Solução em linguagem C#

*using System;*

*namespace programa {*

*class Program {*

*static void Main(string[] args) {*

*double x, y, media;*

*Console.Write(“Digite o primeiro numero: “);*

*x = double.Parse(Console.ReadLine());*

*y = double.Parse(Console.ReadLine());*

*media = (x + y) / 2.0;*

*Console.WriteLine(“Media = “ + media);*

*}*

*}*

*}*

Solução em linguagem Java

*import java.util.Scanner;*

*public class Main {*

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

*Scanner sc = new Scanner(System.in);*

*double x, y, media;*

*System.out.print(“Digite o primeiro numero: “);*

*x = sc.nextDouble();*

*System.out.print(“Digite o segundo numero: “);*

*y = sc.nextDouble();*

*media = (x + y) / 2.0;*

*System.out.println(“Media = “ + media);*

*sc.close();*

*}*

*}*

**IDE – Ambiente Integrado de Desenvolvimento**

É um conjunto de softwares utilizado para a construção de programas.

Exemplos:

* C/C++ : Code Blocks
* Java: Eclipse, NetBeans
* C#: Microsoft Visual Studio

Funcionalidades de uma IDE

* Edição de código fonte (endentação, autocompletar, destaque de palavras, etc.);
* Depuração e testes;
* Construção do produto final (build);
* Sugestão de modelos (templates);
* Auxiliar em várias tarefas do seu projeto;
* Etc.

Compilação e interpretação – Código fonte e objeto Máquina virtual

Código fonte: é aquele escrito pelo programador em linguagem de programação

Compilação

Código fonte -> **compilador: compilação** | *análise léxica + análise sintática* -> Código objeto -> **Gerador de código: construção (build)** | *geração de código* -> Código executável -> **Execução**

**Exemplos de linguagem que tipicamente usam essa abordagem: C, C++**

Interpretação

Código fonte -> **Interpretador: interpretação** | *análise léxica + análise sintática + geração de código SOB DEMANDA* -> **Execução**

Abordagem híbrida

Código fonte **-> Compilador: precompilação** | *análise léxica + análise sintática* -> Bytecode -> **Máquina virtual: interpretação** | *geração de código SOB DEMANDA* -> Execução

**Exemplos de linguagem que tipicamente usam essa abordagem: Java (JVM), C# (Microsoft .NET Framework)**

**Vantagens**

**Compilação:**

* **velocidade do programa**
* **auxílio do compilador antes da execução**

**Interpretação:**

* **flexibilidade de manutenção do aplicativo em produção**
* **expressividade da linguagem**
* **código fonte não precisa ser recompilado para rodar em plataformas diferentes**

**Abordagem híbrida:**

* velocidade do programa
* **auxílio do compilador antes da execução**
* flexibilidade de manutenção do aplicativo em produção
* **código fonte não precisa ser recompilado para rodar em plataformas diferentes**

**C/C++**

Código fonte

Código executável (específico para o sistema operacional)

Sistema Operacional

Hardware

**PHP, Python, JavaScript**

Código fonte

Interpretador (específico para o sistema operacional)

Sistema Operacional

Hardware

**Java, C#**

Código fonte

Bytecode (código precompilado)

Máquina virtual (específica para o sistema operacional)

Sistema Operacional

Hardware

**Java – contextualização**

O que é Java?

Linguagem de programação (regras sintáticas)

Plataforma de desenvolvimento e execução

* Bibliotecas (API)
* Ambientes de execução

Histórico

Problemas resolvidos e motivo do seu sucesso:

* Ponteiros / gerenciamento de memória
* Portabilidade falha: reescrever parte do código ao mudar de SO
* Utilização em dispositivos diversos
* Custo

Criada pela Sun Microsystems no meio da década de 1990

Adquirida pela Oracle Corporation em 2010.

**Aspectos notáveis**

* Código compilado para bytecode e executado em máquina virtual (JVM)
* Portável, segura, robusta
* Roda em vários tipos de dispositivos
* Domina o mercado corporativo desde o fim do século 20
* Padrão Android por muitos anos

**Edições**

* Java ME – Java Micro Edition – dispositivos embarcados e móveis – IoT
* <http://www.oracle.com/technetwork/java/javame>
* Java SE – Java Standard Edition – core – desktop e servidores
* <http://www.oracle.com/technetwork/java/javase>
* Java EE – Java Enterprise Edition – aplicações corporativas
* <http://www.oracle.com/technetwork/java/javaee>

**Plataforma Java SE**

**JVM – Java Virtual Machine**

* Máquina virtual do Java – necessário para executar sistemas Java

**Compilação e interpretação**

* Linguagens **compiladas**: C, C++
* Linguagens **interpretadas**: PHP, JavaScript
* Linguagens **pré-compiladas + máquina virtual**: Java, C#

**Estrutura de uma aplicação Java**

Uma aplicação é composta por classes.

package = agrupamento LÓGICO de classes relacionadas.

Módulo (Java 9+) = Agrupamento lógico de pacotes relacionados.

Runtime = Agrupamento físico.

Aplicação = Agrupamento de módulo relacionados.

**Expressões aritméticas**

Expressão – resultado -> valor numérico

4 + 5 resultado -> 9

**Operadores aritméticos**

*C, C++, Java, C#*

**Operador Significado**

+ adição

- subtração

\* multiplicação

/ divisão

% resto da divisão (“mod”)

**Precedência**: 1º lugar: \* / % 2º lugar: + -

Exemplos de expressões aritméticas

2 \* 6 / 3 Resultado = 4

3 + 2 \* 4 Resultado = 11

(3 + 2) \* 4 Resultado = 20

60 / (3 + 2) \* 4 Resultado = 48

60 / ((3 + 2) \* 4) Resultado = 3

Exemplos com operador “mod”

14 % 3 Resultado = 2

19 % 5 Resultado = 4

**Variáveis - Tipos básicos em Java**

Um programa de computador em execução lida com dados.

Como esses dados são armazenados?

* Em **variáveis**!

**Variáveis**

Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.

Computador > Memória RAM > x = 3; salario = 5000.0; nome = “Maria”

**Declaração de variáveis**

**Sintaxe:**

<tipo> <nome> = <valor inicial>;

Valor inicial é opcional.

**Exemplos:**

int idade = 25;

double altura = 1.68;

char sexp = ‘F’;

*Memória RAM:*

idade = 25

altura = 1.68

sexo = F

**Uma variável possui:**

* Nome (ou identificador)
* Tipo
* Valor
* Endereço

Tipos primitivos em Java



Um bit pode armazenar 2 valores possíveis (0 ou 1)

Cada bit = 2 possibilidades

8 bits:

2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 = 28 = 256 possibilidades

**Nomes de variáveis**

* Não pode começar com dígito: use uma letra ou \_
* Não pode ter espaço em branco
* Não usar acentos ou til
* Sugestão: use o padrão “camel case”

**Errado: Correto:**

*int 5minutos; int \_5minutos;*

*int salário; int salario;*

*int salário do funcionário; int salarioDoFuncionario;*

**As três operações básicas de programação**

Um programa de computador é capaz de realizar essencialmente três operações:

* Entrada de dados;
* Processamento de dados;
* Saída de dados.

**Entrada de dados:**

Usuário -> Programa (dentro de variáveis). Também chamada de LEITURA: “O programa está lendo dados.”

**Processamento de dados:**

É quando o programa realiza os cálculos. O processamento de dados se dá por um comando chamado ATRIBUIÇÃO. media = ( x + y ) / 2.0;

**Saída de dados:**

Programa -> Usuário. Também chamada de ESCRITA: “O programa está escrevendo dados.”

**Saída de dados em Java:**

Para escrever na tela um texto qualquer

**Sem quebra de linha ao final:**

System.out.print(“Bom dia!”);

**Com quebra de linha ao final:**

System.out.println(“Bom dia!”);

Para escrever o conteúdo de uma variável de algum tipo básico:

Suponha uma variável tipo int declarada e iniciada:

*int y = 32;*

*System.out.println(y);*

Para escrever o conteúdo de uma variável com ponto flutuante:

Suponha uma variável tipo double declarada e iniciada:

*double x = 10.35784;*

%n = quebra de linha (independente de plataforma)

*System.out.println(x);*

*System.out.printf(“%.2f%n”, x); - System.out.printf(“%.4f%n”, x);* - Localidade do sistema

**ATENÇÃO:**

Para considerar o separador de decimais como ponto, ANTES da declaração do Scanner, faça:

*Locale.setDefault(Locale.US);*

Para concatenar vários elementos em um mesmo comando de escrita:  
Regra geral para print e println:

elemento1 + elemento2 + elemento3 + ... + elementoN

*System.out.println(“RESULTADO = “* ***+*** *x* ***+*** *“ METROS”);*

Para concatenar vários elementos em um mesmo comando de escrita:

Regra geral para **printf**:

“TEXTO1 %f TEXTO2 %f TEXT03”, variavel1, variavel2

%f = ponto flutuante

%n = quebra de linha

*System.out.printf(“RESULTADO = %.2f metros%n”, x);*

%f = ponto flutuante

%d = inteiro

%s = texto

%n = quebra de linha

*String nome = “Maria”;*

*int idade = 31;*

*double renda = 4000.0;*

*System.out.printf(“%s tem %d anos e ganha R$ %.2f reais%n”, nome, idade, renda);*

*Mais informações: https://docs.oracle.com/javase/tutorial/java/data/numberformat.html*

**Processamento de dados em Java, Casting**

Processamento de dados

Comando de atribuição.

**Sintaxe:**

<variável> = <expressão>;

= Lê-se “recebe”

**Regra:**

1. A expressão é calculada;
2. O resultado da expressão é armazenado na variável.

**Exemplo 1:**

*int x, y;*

*x = 5;*

*y = 2 \* x;*

*System.out.println(x);*

*System.out.println(y);*

**Exemplo 2:**

*int x;*

*double y;*

*x = 5;*

*y = 2 \* x;*

*System.out.println(x);*

*System.out.println(y);*

**Exemplo 3:**

*double b, B, h, area;*

*b = 6.0;*

*B = 8.0;*

*h = 5.0;*

*area = (b + B) / 2.0 \* h;*

*System.out.println(area);*

**Boa prática:**

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:

*.0*

Para **float** use:

*f*

**Exemplo:**

***float*** *b, B, h, area;*

*b = 6f;*

*B = 8f;*

*h = 5f;*

*area = (b + B) / 2f \* h;*

*System.out.println(area);*

**Exemplo 4:**

*int a, g;*

*double resultado;*

*a = 5;*

*g = 2;*

*resultado = (double) a / b;* ***-> casting***

*System.out.println(resultado);*

**Exemplo 5:**

*double j;*

*int p;*

*j = 5.0;*

*p = (int) j;* ***-> casting***

*System.out.println(p);*

Entrada de Dados

**Usuário** -> **Programa** (dentro de variáveis)

Dispositivo de ENTRADA (teclado)

*Também chamada de LEITURA:*

*"O programa está lendo dados."*

**Scanner**

Para fazer entrada de dados, nós vamos criar um objeto do tipo "Scanner" da seguinte forma:

*Scanner sc = new Scanner(System.in);*

**E para funcionar precisamos utilizar:**

*import java.util.Scanner;*

faça sc.close() quando não precisar mais do objeto sc

**Para ler uma palavra (texto sem espaços)**

Suponha uma variável tipo String declarada:

*String x;*

*x = sc.next();*

Exemplo 01:

package entradaDeDados;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner keyboard = new Scanner(System.***in***);

String nome;

nome = keyboard.next();

System.***out***.println("Voce digitou: " + nome);

keyboard.close();

}

}

**Para ler um número inteiro**

Suponha uma variável tipo int declarada:

*int x;*

*x = keyboard.nextInt();*

Exemplo 02:

package entradaDeDados;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner keyboard = new Scanner(System.***in***);

String nome;

int idade;

System.***out***.println("Digite o seu nome:");

nome = keyboard.next();

System.***out***.println("Voce digitou: " + nome);

System.***out***.println("Digite a sua idade:");

idade = keyboard.nextInt();

System.***out***.println("Voce digitou: " + idade);

keyboard.close();

}

}

**Para ler um número com ponto flutuante**

Suponha uma variável tipo double declarada:

*double x;*

*x = sc.nextDouble(); -> Localidade do sistema*

ATENÇÃO:

Para considerar o separador de decimais como ponto, ANTES da declaração do Scanner, faça:

*Locale.setDefault(Locale.US);*

Exemplo 03:

package entradaDeDados;

import java.util.Locale;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Locale.*setDefault*(Locale.***US***);

Scanner keyboard = new Scanner(System.***in***);

String nome;

int idade;

double altura;

System.***out***.println("Digite o seu nome:");

nome = keyboard.next();

System.***out***.println("Voce digitou: " + nome);

System.***out***.println("\nDigite a sua idade:");

idade = keyboard.nextInt();

System.***out***.println("Voce digitou: " + idade);

System.***out***.println("\nDigite a sua altura:");

altura = keyboard.nextDouble();

System.***out***.printf("Voce digitou: %.2f%n", altura);

keyboard.close();

}

}

Locale precisa ser feito antes do Scanner para conseguir receber entrada de dados com . Ao invés de virgula.

Se não quiser usar . e sim virgula, basta não usar o Locale.

**Para ler um caractere**

Suponha uma variável char declarada:

*char x;*

*x = sc.next().charAt(0);*

**Para ler vários dados na mesma linha**

*string x;*

*int y;*

*double z;*

*x = sc.next();*

*y = sc;nextInt();*

*z = sc.nextDouble();*

Para ler um texto ATÉ A QUEBRA DE LINHA

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String s1, s2, s3;

s1 = sc.nextLine();

s2 = sc.nextLine();

s3 = sc.nextLine();

System.out.println(“Dados Digitados:”);

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

sc.close();

}

}

**ATENÇÃO: quebra de linha pendente**

int x;

String s1, s2, s3;

x = sc.nextInt();

s1 = sc.nextLine();

s2 = sc.nextLine();

s3 = sc.nextLine();

System.out.println(“DADOS DIGITADOS:”);

System.out.println(x);

System.out.println(s1);

System.out.println(s2);

System.out.println(s3);

Quando você usa um comado de leitura diferente do nextLine() e dá alguma quebra de linha, essa quebra de linha fica “pendente” na entrada padrão.

Se você então fizer um nextLine(), aquela quebra de linha pendente será absorvida pelo nextLine().

**Solução:**

Faça um nextLine() extra antes de fazer o nextLine() do seu interesse.

**Exemplo**:

package entradaDeDados02;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.***in***);

int x;

String s1, s2, s3;

x = sc.nextInt();

sc.nextLine();

s1 = sc.nextLine();

s2 = sc.nextLine();

s3 = sc.nextLine();

System.***out***.println("Dados Digitados:");

System.***out***.println(x);

System.***out***.println(s1);

System.***out***.println(s2);

System.***out***.println(s3);

sc.close();

}

}

Resumo da aula:

**Scanner**

* next()
* nextInt()
* nextDouble()
* next().charAt(0)

**Locale**

**Como ler até a quebra de linha**

* nextLine()
* como limpar o buffer de leitura