# CURSO JAVA – LEONARDO MOURA LEITAO - UDEMY

VISÃO GERAL

* ALGORITMO: Sequência de passos, processamento.
* ESTRUTURAS DE DADOS: Organiza e administra os dados.
  + FIFO: First In First Out (fila)
  + LIFO: Last In First Out (pacote de bolachas)
  + Árvore: Pasta – Subpasta – Subpasta (Pastas no Pc)
  + Tabela: Linhas e colunas (Planilha)
    - Estruturas de dados básicas: int, double, char, bool (true or false)
* VARIÁVEIS E CONSTANTES: utilizadas para rotular valores. (final int == constante de tipo inteiro)
* OPERAÇÕES: comandos.
  + Atribuição
  + Aritmética
  + Relacional
  + Lógica
* Concatenar == juntar

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FUNDAMENTOS

* ATALHOS:

|  |  |  |
| --- | --- | --- |
| Ctrl + m | - | Oculta a barra lateral |
| Comand + shift + f11 | - | Executa o programa |
| Comand + shift + f | - | Organiza o código |
| Alt + setas | - | Move a linha |
| Comand + Alt + setas | - | Duplica a linha |
| syso | System.out.println() | Imprime na tela |

INFERÊNCIA

Método pelo qual o Java é induzido a definir o tipo da variável, através das informações que lhe foram dadas ao inicializa-la:

* + var = 12; //(tipo inteiro)
  + var = 12,01; //(tipo double)
  + var = ‘A’; //(tipo char)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TIPOS PRIMITIVOS

* 8 TIPOS: 6 numéricos, 1 caractere, 1 bool
  + Numéricos: inteiros, reais
    - byte (1): -128 a 127
    - short (2): -32.768 a -32.767
    - int (4): + ou - 2.147.483.648
    - long (8): é possível separar por \_ (underline); deve-se colocar um L ao final do número para que o compilador o identifique como long (e não como int).
    - float (4)
    - double (8)
  + Caractere:
    - char: ‘a’
    - char: ‘1’
  + Boolean: não identifica 0 e 1 como em outras linguagens
    - true
    - false

OBSERVAÇÕES

1 BYTE == 8 BITS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| + ou - | 1 | 0 | 0 | 1 | 0 | 1 | 1 |

* Valor Literal: Valor chumbado pelo desenvolvedor dentro do programa (não será uma entrada manual do usuário).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

IMPORT

Deve ser declarado entre o pacote e a classe, pode ser digitado direto ou chama-lo no meio do código, o Java insere automaticamente quando usamos Ctrl + space.

* + Java.until.Date (importa data e hora)
  + Javax.swing.JButton (insere botão)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

STRING

String é um conjunto de caracteres (char == tipo primitivo), é um objeto imutável.

String nome = "Karina";

String sobrenome = "Kawamura";

int idade = 20;

double salario = 2000.00;

* + System.***out***.println("Nome: " + nome + "\nSobrenome: " + sobrenome + "\nIdade: " + idade + "\nSalário: " + salario + "\n\n");

System.***out***.print("A %s %s tem %i anos, é analista de desenvolvimento " + "de sistemas e seu salário é de R$%d", nome, sobrenome, idade, salario);

String frase = String.*format*("A %s %s tem %i anos, é analista de desenvolvimento " + "de sistemas e seu salário é de R$%d", nome, sobrenome, idade, salario);

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOTAÇÃO PONTO

A notação ponto é usada para chamar propriedades e aplica-las aos comandos.

|  |  |
| --- | --- |
| .charAt | Imprime somente a letra correspondente ao índice indicado |
| .concat | Juntar |
| .startsWith | Confere se começa com o texto que foi digitado |
| .endsWith | Confere se termina com o texto que foi indicado |
| .toLowerCase | Identifica letras minúsculas |
| .toUpperCase | Identifica letras maiúsculas |
| .length | Conta o tamanho da string (incluindo espaços) |
| .contentEquals | Verifica se a sentença contém exatamente o texto digitado |
| .equalsIgnoreCase | Verifica se a sentença contém o texto digitado (ignora se a letra é maiúscula ou minúscula) |
| .contains | Verifica se a sentença contém o trecho digitado |
| .indexOf | Indica em qual índice inicia o texto digitado |
| .substring | Imprime a sentença a partir do índice indicado |
| .subSequence | Imprime a sentença entre os índices indicados (incluindo o primeiro e excluindo o segundo) |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONSOLE

* Variações do “print”:
  + Print --> mantém as próximas impressões na mesma linha
  + Println --> quebra linha automaticamente
  + Printf --> print formatado, como na linguagem C
* Entrada manual:
  + Scanner nomeDaVariavel = new Scanner(System.in); //declara
  + nomeDaVariavel.nextLine() //chama //.nextLine para string, .nextInt para inteiro
  + nomeDaVariavel.close() //encerra

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PRIMITIVO vs OBJETO

* Objetos:

String exemplo = “Texto”

| |

classe objeto --> aceita atributos e comportamentos

* Tipos primitivos:
  + Por não serem considerados objetos estes não aceitam a notação ponto, mas possuem “wrappers” (embrulho)

|

permite comportamentos dos tipos primitivos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WRAPPERS

|  |  |
| --- | --- |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long (coloca o L no final do num) |
| boolean | Boolean |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONVERSAO TIPOS PRIMITIVOS – CAST

* CAST: (tipo)
  + float b = (float) 1.12345678912 //trunca o num após o dig. 8
  + double e = 1.999999; //declara

int f = (int) e; //converte

System.out.println(f); //saída

1. //impressão (não arredonda, apenas descarta os números após a virgula)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONVERSÃO NÚMERO 🡪 STRING

//Wrapper

Integer num1 = 10000;

//System.out.println(num1.length()); //não conta caracteres de num

System.out.println(num1.toString().length());

//declara como primitivo e transforma em Wrap depois

int num2 = 100000;

System.out.println(Integer.toString(num2).length());

//outro método

System.out.println(("" + num1).length());

System.out.println(("" + num2).length());

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CONVERSÃO STRING 🡪 NÚMERO

//JOptionPane == janela

//showInputDialog == mensagem que aparece na janela

String valor1 = JOptionPane.showInputDialog(

"Digite o primeiro número: ");

String valor2 = JOptionPane.showInputDialog(

"Digite o primeiro número: ");

System.out.println(valor1 + valor2);

//parseDouble == string -> número tipo double

double numero1 = Double.parseDouble(valor1);

double numero2 = Double.parseDouble(valor2);

double soma = numero1 + numero2;

System.out.println("Soma é " + soma);

System.out.println("Média é " + soma / 2);

//BiaDecimal == Precisão no resultado das operações

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_