

Algoritmos 1

Sistemas de Informação – Unisul – 1º Semestre 2015

Apostila para estudo com o conteúdo os Slides de 01 a 05 do Professor Ricardo Ribeiro Assink

Slide 01

O que é um Algoritmo ?

É uma sequência finita de instruções executáveis que visa resolver um problema.

Algumas características:

- Um algoritmo não representa, necessariamente, um programa de computador e sim os passos necessários para realizar uma tarefa.
- Após um determinado número de passos o algoritmo deve terminar, ou seja, deve ser FINITO.
- Após cada passo, um próximo passo deve ser indicado, mesmo que seja o fim da execução.
- Podemos dizer que o ser humano executa algoritmos o tempo todo.
- Toda e qualquer rotina seguida por uma pessoa não deixa de ser um algoritmo, mesmo que não esteja escrito em algum lugar, a sequência de passos para executar determinada tarefa está gravada no subconsciente.

Exemplos de algoritmos:

- Receita de bolo
- Montagem de algum equipamento
- Dirigir
- Sequência de exercícios de um treinamento físico.

Como implementar um algoritmo:

Etapas de execução de um algoritmo pronto:

- 1 – ENTRADA
- 2 – PROCESSAMENTO
- 3 – SAÍDA

- Inicialmente é necessário analisar o que se espera do resultado do algoritmo (SAÍDA).
- Após este procedimento, devemos verificar quais são os dados necessários para se alcançar este resultado (ENTRADA).
- Agora é preciso definir quais procedimentos devem ser executados para manipular os dados de entrada e produzir a saída, chegando assim, ao resultado esperado.

Exemplo de um algoritmo computacional:

Problema: Somar dois números

Para apresentar a solução vamos utilizar uma notação chamada de pseudo-código (português estruturado).

Algoritmo: "SomaDoisValores"

Variável:

SOMA, A, B : inteiro;

Início

Escreva("Digite um número:");

Leia(A);

Escreva("Digite um outro número:");

Leia(B);

SOMA <- A + B;

Escreva(SOMA);

FIM

Exercício 1:

Problema: Calcule a média aritmética de 3 valores, use pseudo-código para esta tarefa.

Algoritmo: "CalculaMediaAritmetica"

Variável:

MEDIA, SOMA, A, B, C : inteiro;

Início

Escreva("Digite um número:");

Leia(A);

Escreva("Digite um outro número:");

Leia(B);

Escreva("Digite um terceiro número:");

Leia(C);

SOMA <- A + B + C;

MEDIA = SOMA / 3 ;

Escreva(MEDIA);

FIM

Slide 02

O que é uma Linguagem de Programação ?

É um método padronizado para informar um conjunto de passos ou instruções para o computador.

Programação Estruturada x Programação Orientada a Objetos

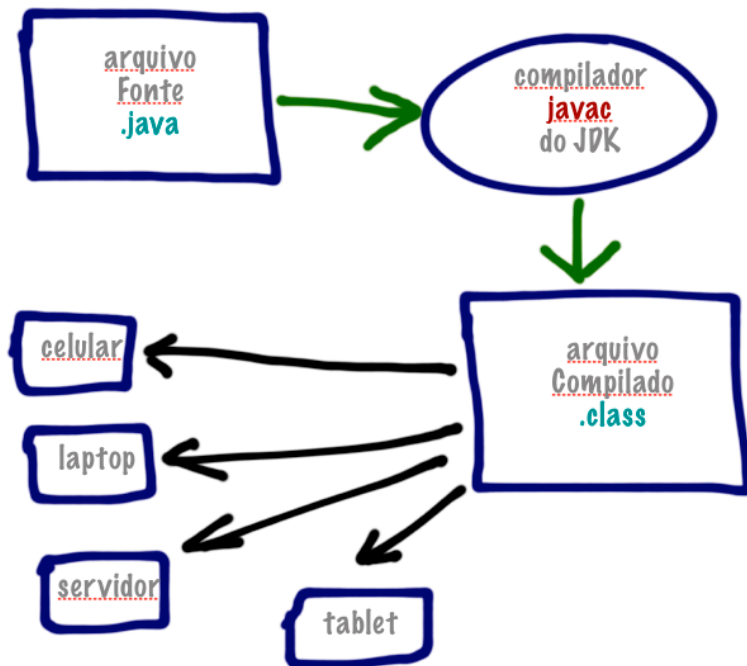
Na programação estruturada a sequência de execução é determinada pela ordem em que as instruções(passos) são escritas.

Na programação orientada a objetos a sequência de execução é determinada por uma rotina principal e as instruções (passos) são escritas e organizadas dentro das classes criadas dentro do programa.

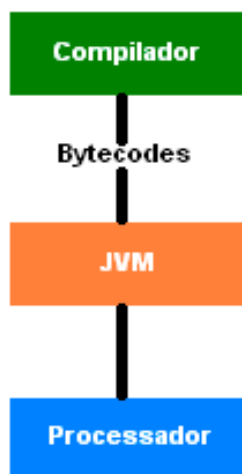
JAVA

- O JAVA é uma linguagem de programação orientada a objetos projetada para ser pequena, simples e portátil.





Máquina Virtual JAVA



Uso do COMPILADOR JAVAC

-O comando **javac** é utilizado para compilar o código fonte do programa.

javac nomedoarquivo.java

- O resultado será a geração de um arquivo chamado: **nomedoarquivo.class**

Execução do programa (Interpretador)

- O comando **java** é utilizado para executar o programa resultante do comando **javac**.

java nomedoarquivo

- Note que utilizamos apenas o nome do arquivo para executá-lo, sem o .class

Vamos escrever nossa primeira classe (programa)?

Vamos aproveitar para entender o que é indentação de código e o que são os delimitadores em JAVA { }

Vamos ver também como mostrar mensagens na tela.

```
import javax.swing.JOptionPane;
public class OlaMundo {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Olá Mundo !");
    }
}
```

Exercício 1

Implemente, compile e execute o exemplo anterior.

Se tiver dificuldades, leia o documento de orientações iniciais disponível para download no site da disciplina.

Outras instruções

\n : usado para quebra de linha em um literal (String)

+ : usado para CONCATENAR (unir) literais

Exercício 2

Faça um programa em JAVA que mostre seu nome, número de matrícula e seu telefone com um dado em cada linha da mensagem.

Use \n para conseguir o efeito desejado.

Exercício 3

Faça um programa em JAVA que reproduza na mensagem a seguinte saída:

```
UNISUL
Universidade do Sul de Santa Catarina
Aluno: Seu nome
Curso: Sistemas de informação
Matrícula: Sua matrícula
```

Slide 03

O que é uma variável ?

Variável é uma região de memória que existe apenas em tempo de execução que armazena um valor ou conjunto de valores.

Características de uma variável

- Toda variável possui um NOME, que identifica a variável para que possamos utilizá-la dentro de programas de computadores.
- Toda variável tem um TIPO, que define o valor que será armazenado dentro dela.
- Toda variável deve ser DECLARADA para que a mesma exista dentro do programa que está sendo escrito.

A declaração dentro do programa funciona assim:

<TIPO> <Nome da Variável> ;

Exemplo de DECLARAÇÃO de uma variável

```
import javax.swing.JOptionPane;
public class OlaMundo {
    public static void main(String[] args) {
        String nome;
    }
}
```

Neste exemplo declaramos a variável **nome** do tipo **String**.

O que é uma instrução de ATRIBUIÇÃO

Uma instrução de ATRIBUIÇÃO redefine o VALOR de uma variável.

Em JAVA, usamos o símbolo de = Para atribuir valores a variáveis.

Exemplo de ATRIBUIÇÃO

```
import javax.swing.JOptionPane;
public class OlaMundo {
    public static void main(String[] args) {
        String nome;
        nome = "Ricardo";
    }
}
```

Note que por se tratar de uma **String**, precisamos colocar o valor **entre aspas**. Agora a variável **nome** tem o valor **Ricardo**

Agora, dentro deste programa, enquanto estiver em execução, podemos invocar a variável **nome** que o valor **Ricardo** aparecerá.

A qualquer momento a variável nome pode ser “reconfigurada”, ou seja, é possível executar uma instrução de atribuição novamente e MUDAR o valor da variável. Exemplo:

nome = “Maria”;

Exercício 4

Faça um programa em JAVA declarando uma variável chamada nome, e siga os seguintes passos:

- 1 – Atribua o valor “João” a esta variável
- 2 – Mostre o valor da variável nome em uma mensagem
- 3 – Mude o valor da variável nome para outro valor qualquer
- 4 - Mostre o valor da variável nome em uma mensagem seguindo o seguinte modelo

O valor da variável é: <valor inserido no passo 3>

Exercício 5

Faça um programa em JAVA, declarando uma variável do tipo inteiro (int), atribua um valor qualquer a esta variável, depois mostre o valor desta variável em uma mensagem.

Exercício 6

Repita o exercício anterior, mas agora mostre a mensagem abaixo:

O valor da variável é : <número escolhido>

Slide 04

Tipos primitivos de dados

Em sistemas de informação, tipo de variáveis ou dados é uma combinação de valores e de operações que uma variável pode executar.

A cada variável está associado um Tipo de Dados.

Com o tipo de dado, o compilador ou interpretador consegue verificar as conversões necessárias para obter os valores em memória.

Em Java existem somente 8 tipos primitivos.

Os tipos primitivos não necessitam de métodos construtores para sua criação na memória.
Por este motivo não precisam ser instanciados (new).

Array e String não são considerados tipos primitivos.

Veja a lista dos tipos primitivos em JAVA:

Há quatro tipos inteiros em JAVA

Tipo	Tamanho	Valor
byte	8 bits	-128 a 127
short	16 bits	-32.768 a 32.767
int	32 bits	-2.147.483.648 a 2.147.483.647
long	64 bits	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

Um número inteiro pode sempre ser atribuído a outro de maior precisão:

```
int a = 274;
```

```
long b = a;
```

Há dois tipos em ponto flutuante em JAVA

Tipo	Tamanho	Valor
float	32 bits	-3.40292347E+38 a +3.40292347E+38
double	64 bits	-1.79769313486231570E+308 a +1.79769313486231570E+308

Há um tipo primitivo que representa um caractere em JAVA

Tipo	Tamanho	Valor
char	16 bits	'\u0000' a '\uFFFF'

O JAVA utiliza o padrão UNICODE para os Caracteres

Há um tipo primitivo booleano em JAVA

Tipo	Tamanho	Valor
boolean	1 bit	true OU false

Em Java, estas variáveis não podem ser interpretadas como os números inteiros 0 e 1.

Entrada e Saída de dados

Já vimos um pouco da utilização da classe JOptionPane para a saída de dados (mostrar mensagens).

Agora vamos ver como usá-la para a Entrada de dados.

A instrução é:

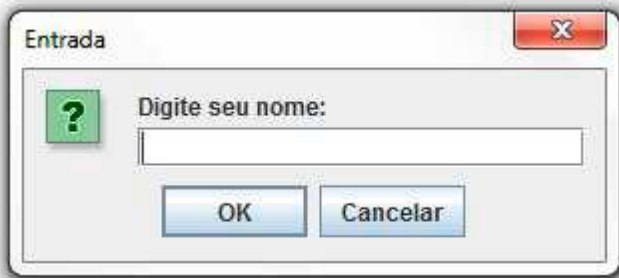
```
JOptionPane.showInputDialog("mensagem");
```

Com esta instrução uma janela com um campo para digitação aparecerá na tela.

Veja o código abaixo:

```
import javax.swing.JOptionPane;
public class Exemplo {
    public static void main(String[] args) {
        String nome;
        nome = JOptionPane.showInputDialog("Digite seu nome:");
    }
}
```

O código anterior gera o seguinte resultado.



Se você digitar João, e clicar em OK, este texto será armazenado na variável **nome**.

Conversão de dados mais utilizadas

Quando se solicita algo usando a instrução **showInputDialog**.

O resultado é sempre LITERAL.

Muitas vezes é necessário solicitar tipos numéricos ao usuário.

Para isso usamos instruções de CONVERSÃO.

Conversão de dados mais utilizadas

Para converter Literal para Inteiro.

Integer.parseInt(<Aqui vai o valor literal>);

Para converter Literal para Double.

Double.parseDouble(<Aqui vai o valor literal>);

Vamos ver isso no código ?

```
import javax.swing.JOptionPane;
public class Exemplo {
    public static void main(String[] args) {
        int numero;
        numero = Integer.parseInt(JOptionPane.showInputDialog("Digite um
número:"));
    }
}
```

Exercício 7

Escreva um programa em JAVA que solicite ao usuário um nome, armazene em uma variável e depois mostre o nome digitado em uma mensagem.

Exercício 8

Escreva um programa em JAVA que solicite ao usuário 2 números inteiros (um de cada vez), armazene os valores em variáveis e depois mostre a soma dos dois em uma mensagem.

Exercício 9

Escreva um programa em JAVA que solicite ao usuário o seu peso(aqui entenda massa) em Kg (int) e a sua altura em metros (Double). Com estes dados calcule o IMC – Índice de Massa

Corporal. Seguindo a seguinte fórmula.

$IMC = massa / (altura * altura)$

Mostre o resultado em uma mensagem.

Slide 05

Expressões

Uma vez que já temos os dados e as variáveis podemos passar ao próximo estágio que seria a criação de expressões.

Exemplo:

$x = (20 * 5) / 10$

Expressões Aritméticas

Expressões aritméticas são aquelas que apresentam como resultado um valor numérico que pode ser um número inteiro ou real. *(Como no exemplo anterior.)*

Para criar expressões aritméticas é necessário o uso de operadores aritméticos.

Operadores Aritméticos

```
+ // soma
- // subtração
* // multiplicação
/ // divisão
% // módulo (resto da divisão inteira)
```

OBS: fique atento ao uso de parênteses (), é igual na matemática básica, os algoritmos resolvem as expressões dando prioridade (resolvendo primeiro) ao interior dos parênteses.

Operadores Lógicos

```
> // maior
>= // maior ou igual
< // menor
<= // menor ou igual
== // igual
!= // diferente
|| // um OU outro
&& // um E outro
```


Exercício 10

Escreva um programa em JAVA que solicite ao usuário 2 números inteiros (um de cada vez), armazene os valores em variáveis e depois mostre a subtração dos dois em uma mensagem.

Exercício 11

Escreva um programa em JAVA que solicite ao usuário 2 números REAIS (double) (um de cada vez), armazene os valores em variáveis e depois mostre a divisão dos dois em uma mensagem.

Exercício 12

Escreva um programa em JAVA que solicite ao usuário 2 números inteiros (um de cada vez), armazene os valores em variáveis e depois mostre a multiplicação dos dois em uma mensagem.

Exercício 13

Faça o exercício 11 novamente mas declare as variáveis como inteiros. Veja o que acontece quando você divide 9 por 2.

Exercício 14

Faça um programa em JAVA que implemente a discriminante da equação quadrática :

$$\Delta = b^2 - 4ac$$