

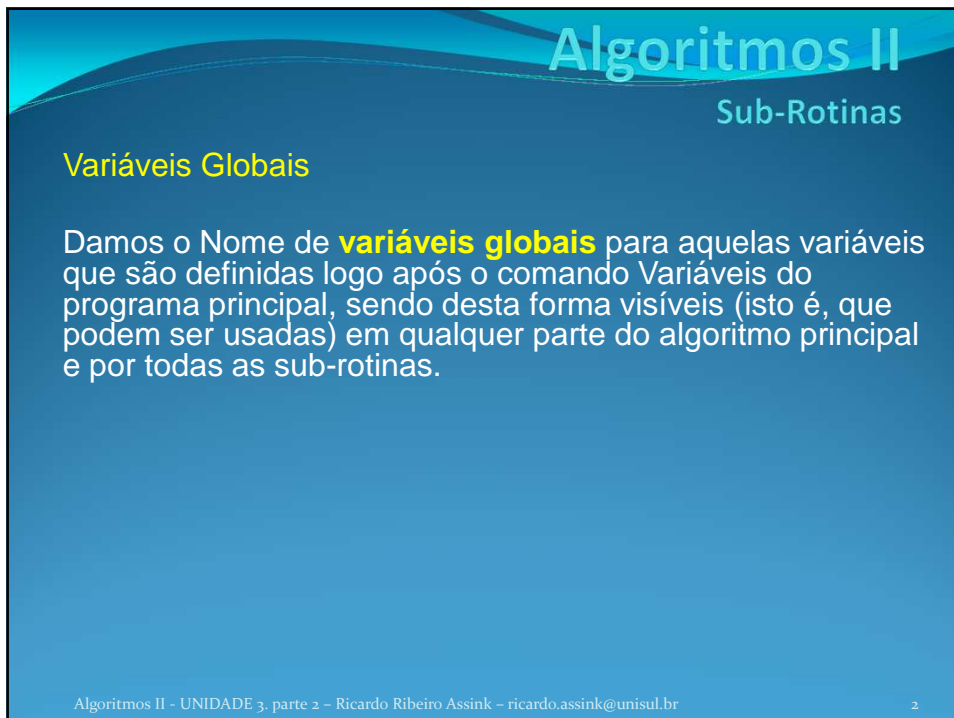
A blue gradient slide with a wavy white line at the top. The title 'Algoritmos II' is in large white font. Below it, the author's name 'Ricardo Ribeiro Assink' and two email addresses are listed. At the bottom, a website URL is shown in large white font. A small footer at the very bottom contains the course name and slide number.

Algoritmos II

Ricardo Ribeiro Assink
ricardo.assink@unisul.br
ricardo@equipedigital.com

<http://www.ricardoassink.com.br/>

Algoritmos II - UNIDADE 3, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 1

A blue gradient slide with a wavy white line at the top. The title 'Algoritmos II' and subtitle 'Sub-Rotinas' are in white font. The section title 'Variáveis Globais' is in yellow. The main text explains the purpose of global variables. A small footer at the very bottom contains the course name and slide number.

Algoritmos II

Sub-Rotinas

Variáveis Globais

Damos o Nome de **variáveis globais** para aquelas variáveis que são definidas logo após o comando Variáveis do programa principal, sendo desta forma visíveis (isto é, que podem ser usadas) em qualquer parte do algoritmo principal e por todas as sub-rotinas.

Algoritmos II - UNIDADE 3, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br 2

Algoritmos II

Sub-Rotinas

Variáveis Globais – Exemplo

Algoritmo LeNome

Variáveis

nome : **Literal** //Variável Global

Sub-rotina preencheNome()

Início

Escreva("Digite o nome")

Leia(nome)

Fim

Início

preencheNome()

Escreva("O nome é:" + nome)

Fim.

Algoritmos II

Sub-Rotinas

Variáveis Globais

No Exemplo acima, a variável “Nome”, sendo definida como global, pode ser manipulada dentro de qualquer ponto do programa, sendo que qualquer mudança no seu conteúdo, será visível nas demais partes da Rotina.

Algoritmos II

Sub-Rotinas

Variáveis Locais

Damos o Nome de **variáveis locais** às variáveis que são declaradas dentro de uma sub-rotina, sendo que as mesmas só podem ser visíveis (manipuladas) dentro da sub-rotina que as declarou, não sendo visíveis em nenhuma outra parte do algoritmo principal ou de outras sub-rotinas.

Algoritmos II

Sub-Rotinas

Variáveis Locais – Exemplo

Algoritmo EscreveNumero

Sub-rotina *mostraNumeros()*

Variáveis

n, numero: Inteiro // **Variáveis Locais**

Início

Escreva("Digite a quantidade de números:")

Leia(n)

Para numero de 1 até n **Faça**

Escreva("Numero: "+numero)

Fim_para

Fim

Início

mostraNumeros()

Fim.

Algoritmos II

Sub-Rotinas

Variáveis Locais e Globais

Obs: É possível definir variáveis globais e locais com o mesmo Nome, sendo qualquer mudança no conteúdo da variável local não afetará o conteúdo da variável global.

No Exemplo que segue, a variável global "Nome" e a variável local "Nome" representam posições de memória totalmente diferentes, logo, qualquer mudança no conteúdo da variável local, não afetará o conteúdo da variável global.

Algoritmos II

Sub-Rotinas

```
public class Principal{
    static String nome; //Variável Global

    public static void modificaNome(){
        nome = JOptionPane.showInputDialog("Digite o nome:");
    }

    public static void leNome(){
        String nome; //Variável Local
        nome = JOptionPane.showInputDialog("Digite o nome:");
    }

    public static void main(String args[]){
        modificaNome();
        JOptionPane.showMessageDialog(null,"O nome é : "+
        nome);
        leNome();
        JOptionPane.showMessageDialog(null,"O nome é : "+
        nome);
    }
}
```

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros

Parâmetros são canais pelos quais se estabelece uma comunicação bidirecional entre uma sub-rotina e o algoritmo chamador (algoritmo principal ou outra sub-rotina). Dados são passados pelo algoritmo chamador a sub-rotina, ou retornados por este ao primeiro por meio de parâmetros.

Como já foi dito, as Sub-rotinas foram criadas para serem genéricas o bastante para se adaptarem a qualquer situação, visando justamente a possibilidade de reutilização do código. Para realizar esta “mágica”, foi criado o conceito de **passagem de parâmetros**, ou seja, passar informações para serem tratadas dentro da Sub-Rotina.

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros

```
public class Principal{

    public static void escreveNome(int n, String nome){
        int i;
        for(i=0;i<n;i++){
            JOptionPane.showMessageDialog(null, nome );
        }
    }

    public static void main(String args[]){
        String funcionario;
        int numero;
        numero = Integer.parseInt(
Numero: ");
        JOptionPane.showInputDialog("Digite o
Funcionario: ");
        funcionario = JOptionPane.showInputDialog("Digite
        escreveNome(numero,funcionario);
        escreveNome(10,"Joao");
    }
}
```

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros

Parâmetros formais são os nomes simbólicos introduzidos no cabeçalho de sub-rotinas, usados na definição dos parâmetros da mesma. Dentro da sub-rotina trabalha-se com estes nomes da mesma forma como se trabalha com variáveis locais ou globais. No exemplo acima são as variáveis **N** e **Nome** no cabeçalho da sub-rotina **escreveNome**.

Parâmetros reais são aqueles que substituem os parâmetros formais quando da chamada de uma sub-rotina. No exemplo acima o algoritmo principal chama a sub-rotina **escreveNome** com os parâmetros **Numero** e **Funcionário** substituindo os parâmetros formais **N** e **Nome** respectivamente. Na outra chamada da sub-rotina são passados os parâmetros reais 10 e “João” para **N** e **Nome**.

Assim, os parâmetros formais são úteis somente na definição (formalização) da sub-rotina, ao passo que os parâmetros reais substituem-nos a cada invocação da sub-rotina.

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros por VALOR

É quando a alteração dos conteúdos dos parâmetros passados para a sub-rotina não será refletida no algoritmo principal.

No Exemplo que segue, o conteúdo da variável “X” não será alterado após o retorno ao algoritmo principal. O conteúdo da variável X foi copiado (atribuído) para a variável A da sub-rotina porValor. Portanto X e A são variáveis diferentes de espaços de memórias diferentes.

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros por VALOR

```
public class TestePorValor {
    public static void porValor(int a){
        JOptionPane.showMessageDialog("A1: " + a
    );
        a = 5;
        JOptionPane.showMessageDialog("A2: " + a
    );
    }
    public static void main(String args[]){
        int x;
        x = 10;
        JOptionPane.showMessageDialog("X1: " + x
    );
        porValor(x);
        JOptionPane.showMessageDialog("X2: " + x
    );
    }
}
```

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros por Referência

Quando a alteração do conteúdo de um parâmetro, dentro de uma sub-rotina, se reflete no algoritmo principal, os parâmetros são ditos por referência.

Obs: Usa variáveis globais.

No Exemplo que segue, o conteúdo da variável "X" será alterado após o retorno ao programa principal.

Algoritmos II

Sub-Rotinas

Passagem de Parâmetros por Referência

```
public class TestePorReferencia {
    static int x;
    public static void porReferencia(){
        JOptionPane.showMessageDialog("A1: " + x );
        x = 5;
        JOptionPane.showMessageDialog("A2: " + x );
    }

    public static void main(String args[]){
        x = 10;
        JOptionPane.showMessageDialog("X1: " + x );
        porReferencia();
        JOptionPane.showMessageDialog("X2: " + x );
    }
}
```

Algoritmos II

Sub-Rotinas

Função

Possui as mesmas características de um “Procedimento” no que se refere a passagem de parâmetros por valor e por referência, variáveis globais e locais, mas possui uma importante diferença, que é o **retorno de um valor ao término** de sua execução, ou seja, uma Função sempre deverá retornar **um valor ao algoritmo principal**.

Deverá ser informado qual o **tipo do valor retornado**, sendo que poderá ser usado, nesta definição, tanto tipos pré-definidos da linguagem, como tipos definidos pelo usuário.

Somente um valor pode ser retornado para o tipo especificado.

Algoritmos II

Sub-Rotinas

Função

```
public class Principal {

    public static int soma(int num1, int num2){
        int total;
        total = num1 + num2;
        return total;
    }

    public static void main(String args[]) {
        int a,b,resultado;

        a = Integer.parseInt(JOptionPane.showInputDialog("Digite
A:"));
        b = Integer.parseInt(JOptionPane.showInputDialog("Digite
B:"));
        resultado = soma(a,b);
        JOptionPane.showMessageDialog("O resultado é : "+resultado);
    }
}
```

Algoritmos II - UNIDADE 3, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

17

Algoritmos II

Sub-Rotinas

Exercícios

Nestes exercícios utilize somente sub-rotinas tipo função. Aplique o conceito de variável global e local. Todos os dados que forem necessários como entrada para a execução da sub-rotina devem ser passados como parâmetro e o dado de saída devem ser feita por retorno a sub-rotina.

1- Faça um algoritmo que leia um número N inteiro e positivo. O algoritmo deverá ter uma sub-rotina tipo função que irá calcular o fatorial deste número. A sub-rotina não deverá mostrar o resultado do cálculo, isto ficará a cargo do algoritmo principal ela também irá receber por parâmetros os dados necessários para o seu processamento e retornar o valor resultante. (passagem de parâmetro por valor)

Algoritmos II - UNIDADE 3, parte 2 - Ricardo Ribeiro Assink - ricardo.assink@unisul.br

18

Algoritmos II

Sub-Rotinas

Exercícios

2- Faça um algoritmo que possui uma sub-rotina tipo função que calcule um valor A inteiro elevado a um expoente B também inteiro. Não usar $A^{**}B$ e sim uma estrutura de repetição. A sub-rotina não deverá mostrar o resultado do cálculo, isto ficará a cargo do algoritmo principal. (use passagem de parâmetro por referência)

3 - Faça uma sub-rotina do tipo função que retorne 1 se o número inteiro passado por parâmetro for positivo e 0 se o número for negativo. Leia um número e passe para função e diga se o mesmo é positivo ou negativo.

Algoritmos II

Sub-Rotinas

Exercícios

4 - Faça uma sub-rotina do tipo função que receba dois números inteiros positivos por parâmetro e retorne a **soma dos N números existentes entre eles**. Faça um algoritmo para utilizar a função e mostrar a soma.

5 – Faça uma sub-rotina que receba um parâmetro inteiro (peso em Kg) outro double (altura em metros) e uma string (para o nome). Calcule o índice de massa corporal (IMC) que é igual ao peso dividido pela altura ao quadrado. No programa principal solicite estes parâmetros ao usuário e chame a sub-rotina. A sub-rotina deve mostrar o resultado com o nome e IMC calculado (double).

