Modularidade

Objetivos:

- Introduzir noções básicas de modularidade
- Funções e procedimentos

Motivação

- É muito difícil manter um código quando ele tende a ser grande (com muitas linhas)

- É preciso organizar o programa em partes bem definidas (módulos)

Motivação

- Modularidade
- Reaproveitamento de código
- Legibilidade

Modularidade

• Programas podem ser divididos em módulos, que são partes bem definidas de um programa que normalmente são soluções de subproblemas de um problema maior

Modularidade

Um programa pode ser separado em:

- Entrada dos dados
- Cálculos propriamente ditos
- Saída dos dados

Reaproveitamento de código

Um programa pode ter cálculos repetidos, ou bem parecidos, que podem ser evitados

- Exemplo: dados 10 variáveis, saber se seus valores são pares:

- x1 mod 2 = 0
- x2 mod 2 = 0
- ...
- x10 mod 2 = 0

Legibilidade

- Um programa é legível quando é fácil de ler o código e entender o que está acontecendo

- Por exemplo, se um cálculo é feito em várias linhas de código que ocupam muitas telas, tem-se maior dificuldade em ler o programa e entender estes cálculos, encontrar erros, etc.

Exemplo

Programa que lê dois números e imprime mensagem se ambos forem pares:

```
Begin
        Read (a,b);
        If (a mod 2 = 0) and (b mod 2 = 0) then
                Writeln ('ambos são pares');
End.
```

Exemplo

Programa que lê dois números e imprime mensagem se ambos forem primos:

```
Begin
        Read (a,b);
        If “a é primo” and “b é primo” then
                Writeln ('ambos são primos');
End.
```

Exemplo

É possível compreender as questões de legibilidade e aproveitamento de código no programa anterior

Funções e procedimentos

A linguagem Pascal não permite escrever programas completamente modulares, mas possui noções de subprogramas do tipo funções e procedimentos, que permitem iniciar o estudo

- Modula2 e Oberon são exemplos de evoluções da linguagem pascal que permitem escrever programas em outros paradigmas, por exemplo orientação a objetos

Funções e procedimentos

Conceitos fundamentais:

- Semântica da função e do procedimento
- Escopo de variáveis (globais ou locais)
- Passagem de parâmetros (por referência ou por valor)

Funções

São subprogramas utilizados para fazer um certo cálculo e retornar um valor

function <id> (parâmetros opcionais): <TIPO>;
<variáveis locais>
begin
(* código da função *)
<id>:= “expressão”;
end;

Exemplo

```
function eh_par (n: integer): boolean;
 begin
            eh_par:= n mod 2 = 0;
 end;
```

Exemplo

Programa que lê dois números e imprime mensagem se ambos forem pares:

```
Begin
        Read (a,b);
        if eh_par (a) and eh_par (b) then
                writeln ('ambos são pares');
End.
```

Exemplo

```
function eh_par (n: integer): boolean;
 begin
          eh_par:= n mod 2 = 0;
 end;
```

Observar:

- nome da função
- parâmetro passado por valor (ou cópia)
- o tipo do valor de retorno
- A instrução eh_par:=

Programa completo

```
program exemplo;
var a,b: integer;

function eh_par (n: integer): boolean;
 begin
            eh_par:= n mod 2 = 0;
 end;

 begin
        read (a,b);
        if eh_par (a) and eh_par (b) then
                writeln ('ambos são pares');
 end.
```

Parâmetros por referência

```
program exemplo;
var a,b: integer;

function eh_par (VAR n: integer): boolean;
 begin
            eh_par:= n mod 2 = 0;
 end;

 begin
        read (a,b);
        if eh_par (a) and eh_par (b) then
                writeln ('ambos são pares');
 end.
```

Parâmetros por referência

```
program exemplo;
var a,b: integer;

function eh_par (n: integer): boolean;
var teste: boolean
 begin
            teste:= n mod 2 = 0;
            eh_par:= teste;
 end;

 begin
        read (a,b);
        if eh_par (a) and eh_par (b) then
                writeln ('ambos são pares');
 end.
```

Funções sem parâmetros

```
program exemplo;       (* uso errado !!! *)
var a: integer;

function eh_par_a: boolean;
 begin
            eh_par_a:= a mod 2 = 0;
 end;

 begin
        read (a);
        if eh_par_a  then
                writeln ('é par');
 end.
```

Funções sem parâmetros

- Funções sem parâmetro existem, mas seu uso é raro

Exemplo

• Ler pares de números inteiros até que seja lido algum par contendo um zero. Para cada par (A,B), se ambos forem positivos, calcular o MDC entre eles pelo método de Euclides. Se o MDC for igual a 1, imprimir “são primos entre si”, caso contrário imprimir “não são primos entre si”

Exercício

• Ler trincas de números inteiros (a,b,c) até que seja lido a trinca (0,0,0). Para cada trinca, se a for diferente de zero calcular as raízes reais da equação do segundo grau $ax^2 + bx + c = 0$. Imprimir mensagem adequada se não houver raízes reais. Use 3 funções: uma para o calculo do discriminante e outras duas para o cálculo das duas raízes.