

Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.^a Dr.^a Mariane Cassenote

mariane@inf.ufpr.br

Desvios condicionais

Objetivos da aula

1. Compreender comandos que permitem alteração do fluxo de execução do programa

Aula elaborada com base na bibliografia da disciplina e nos materiais dos Profs. Marcos Castilho e André Vignatti, disponíveis no link

<https://www.inf.ufpr.br/cursos/ci055/>

Conceitos elementares de linguagens de programação

- Fluxo de execução de um programa
- Comandos que manipulam dados e permitem interação com o usuário
- Expressões aritméticas e lógicas
- **Comandos que permitem alteração do fluxo de execução do programa**
 - Repetição
 - **Condicional**

Na aula passada... Repetição!

- O que o **while** faz?
 - Ele permite que o computador execute o mesmo trecho de código várias vezes enquanto uma condição for verdadeira
- Quem controla a repetição?
 - Contador
 - Decisão do usuário

Na aula passada... Repetição!

- Os 3 Pilares da Repetição
 - **Inicialização**: Onde começo?
 - **Condição**: Até onde vou?
 - **Atualização**: Como avanço?
- Cuidado com **repetições infinitas** e com **repetições nunca executadas**!

Comandos de desvio condicional

- Permitem escolher trechos de códigos para **serem executados ou não**
- O fluxo de execução dos comandos é alterado de maneira controlada

O comando if, primeira versão

```
1  if { expressao booleana } then  
2    { algum comando };
```

- Sintaxe versus semântica
 - Acima está a sintaxe do comando **if**
 - A semântica é o significado do comando

O comando if, primeira versão

```
1  if { expressao booleana } then  
2      { algum comando };
```

- A expressão acima significa que este { algum comando } poderá ou não ser executado
- Quem define se o comando será executado é a { expressão booleana }
- Se esta { expressão booleana } for verdadeira o { algum comando } será executado

Exemplo 1

Problema: Ler um único número do teclado e imprimi-lo se ele for positivo.

- Comando **if-then**;
- Executa comandos baseado no resultado de uma expressão booleana;

```
1 program imprime_se_positivo ;  
2 var a: integer;  
3  
4 begin  
5     read (a);  
6     if a > 0 then  
7         writeln (a); (* so executa se a for positivo *)  
8 end.
```

Exemplo 2

Imprimir as raízes reais pela equação de Bhaskara somente se o discriminante não for negativo

```
1 program bhaskara_v4;  
2 var a, b, c, delta: real;  
3  
4 begin  
5     read (a, b, c);  
6     delta:= b * b - 4 * a * c;  
7     if delta >= 0 then  
8         begin  
9             writeln ((-b - sqrt(delta)) / (2 * a));  
10            writeln ((-b + sqrt(delta)) / (2 * a));  
11        end;  
12 end.
```

- Faz os cálculos e imprime somente se **delta >= 0**
- Senão não faz nada e termina o programa

O comando if, segunda versão

```
1  if { expressao booleana } then  
2      { algum comando }  
3  else  
4      { outro comando }
```

- Sintaxe versus semântica
 - Acima está a sintaxe do comando **if**
 - A semântica é o significado do comando

O comando if, segunda versão

```
1  if { expressao booleana } then
2      { algum comando }
3  else
4      { outro comando }
```

- A expressão acima significa que **ou** { **algum comando** } será executado **ou** { **outro comando** } será executado
- Quem define qual comando será executado é a { **expressão booleana** }
- Se esta { **expressão booleana** } for **verdadeira** o { **algum comando** } será executado
- Se esta { **expressão booleana** } for **falsa** o { **outro comando** } será executado

Exemplo 1

Problema: Ler um único número do teclado e imprimí-lo se ele for positivo.

Se não for, imprimir a mensagem “número inválido”

- Comando **if-then-else**;
- Baseado no resultado de uma expressão booleana executa um fluxo ou outro (mas não os dois)

Exemplo 1

```
1 program imprime_se_positivo ;  
2 var a ,b :integer;  
3  
4 begin  
5     read (a);  
6     if a > 0 then  
7         writeln (a) (* so executa se a for positivo *)  
8     else  
9         writeln ('numero invalido'); (* executa se a for nulo ou  
10                                     negativo *)  
11 end.
```

- **Atenção:** Em Pascal não é permitido colocar um **ponto-e-vírgula (;) antes do else**

Exemplo 2

Problema: Imprimir as raízes reais pela equação de Bhaskara
somente se o discriminante não for negativo.

Se for negativo, imprimir uma mensagem informando isso.

- Faz os cálculos e imprime as raízes somente se **delta ≥ 0**
- Senão, imprime mensagem informando que não há raízes reais

Exemplo 2

```
1 program bhaskara_v5;
2 var a, b, c, delta: real;
3
4 begin
5     read (a, b, c);
6     delta:= b * b - 4 * a * c;
7     if delta >= 0 then
8         begin
9             writeln ((-b - sqrt(delta)) / (2 * a));
10            writeln ((-b + sqrt(delta)) / (2 * a));
11        end
12    else
13        writeln ('nao existem raizes reais');
14 end.
```

- Não deve haver **else** sem **if**
- Quando houver **else**, **não usar** **ponto-e-vírgula (;)** no **end** do bloco do comando **if**

Exercícios

- Fazer os exercícios da seção 5.10.6 do livro

`https://www.inf.ufpr.br/marcos/livro\_alg1/livro\_alg1.pdf`

Fim do tópico

- O conteúdo desta aula está no livro no capítulo 5, seção 5.8
- Na próxima aula veremos aninhamento de desvios condicionais

Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.^ª Dr.^ª Mariane Cassenote

mariane@inf.ufpr.br

Desvios condicionais