

CI1055: Algoritmos e Estruturas de Dados I

Prof. Dr. Marcos Castilho

Departamento de Informática/UFPR

14 de julho de 2020

Resumo

Fluxo de execução, entrada e saída, erros

Conceitos elementares de linguagens de programação

- o fluxo de execução de um programa
- os comandos que manipulam dados e permitem interação com o usuário
- as expressões aritméticas e lógicas
- os comandos que permitem alteração do fluxo de execução do programa

Um exemplo simples de programa em *Pascal*

```
program alomamae;  
  
begin  
    write ('alo mamae!');  
    writeln;  
end.
```

- cabeçalho do programa
- programa principal
 - comandos de impressão
- primeiros elementos da linguagem *Pascal*

O fluxo de execução de um programa

```
program alomamae;  
  
begin  
    write ('alo mamae!');  
    writeln;  
end.
```

- o programa inicia após o `begin`
- o programa termina quando encontra o `end.`
- os comandos são executados de cima para baixo
- linha por linha
- em cada linha, da esquerda para direita

Código fonte versus código executável

```
program alomamae;  
  
begin  
    write ('alo mamae!');  
    writeln;  
end.
```

- o texto acima é o código fonte
- para obter o executável é preciso *compilar*
- vejamos

Comandos de entrada e saída

- permitem a interação com o usuário
- comandos de leitura:
 - `read` e `readln`
- comandos de saída:
 - `write` e `writeln`

Exemplo

```
program le_e_imprime;  
var numero: integer;  
  
begin  
    read (numero);  
    writeln (numero);  
end.
```

- este program lê um valor inteiro do teclado
- depois imprime o valor lido na tela
- vejamos

```
program le_e_imprime;  
var numero: integer;  
  
begin  
    read (numero);  
    writeln (numero);  
end.
```

- variáveis são endereços de memória
- acessíveis por um identificador escolhido pelo programador
- no caso: `numero`

- familia ordinal: `integer`, `longint`, ...
- familia real: `real`, `double`, ...
- ...

- são as que exigem explicitamente dizer qual é o tipo da variável
- ou seja, como os bits serão interpretados pelo compilador
- *Pascal* é uma linguagem tipada
- vejamos

Linguagens fortemente tipadas

- não permitem que se altere o tipo durante a execução do programa
- *Pascal* é fortemente tipada
- vejamos

Até ordem contrária, todos os nossos programas usarão o tipo *longint*. Isso não nos impedirá de aprender os princípios de programação e algoritmos para depois nos preocuparmos com detalhes.

- 1 Antes de iniciar o programa principal, o programa solicita ao *sistema operacional* que reserve um espaço de memória para a variável `numero`, que deve ser interpretada como um `longint`

- 1 Inicia o programa pela primeira instrução (`read(numero);`), que faz com que o computador espere o usuário digitar algo no teclado e apertar ENTER.
- 2 Se o usuário digitar, por exemplo, o número 2, então o endereço de memória associado à variável `numero` conterá o valor 2.

- 1 Executa a segunda instrução (`writeln(numero);`), que faz com que o computador busque na memória o valor (o conteúdo) de `numero`, que é 2,
- 2 imprime esta informação, isto é, 2, na tela.
- 3 depois muda de linha.
- 4 O programa termina.

- o ciclo edição-compilação-testes
- variantes dos comandos estudados até aqui
- erro de compilação
- erro de execução
- erro de lógica

Fim do primeiro tópic

- o conteúdo desta aula está no livro no capítulo 5, seções de 5.1 até 5.4
- depois do esclarecimento de dúvidas passaremos às expressões aritméticas e booleanas

- Slides feitos em \LaTeX usando beamer
- Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>