Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.ª Dr.ª Mariane Cassenote

mariane@inf.ufpr.br

Repetição de comandos

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**

Comandos de repetição

- Permitem repetir trechos de códigos
- Existem três destes comandos em Pascal
- Só veremos uma forma por enquanto
- A motivação será a partir de um problema bem simples

Problema: imprimir os números de 1 até 5

```
program imprimir_de_1_a_5;

begin
    writeln (1);
    writeln (2);
    writeln (3);
    writeln (4);
    writeln (5);
end.
```

- Solução muito simples!
- Poderia ter uma linha só: `writeln ('1 2 3 4 5');`

Problema

- Simples demais!
- Não é generalizável para problemas similares
- Por exemplo:

Problema: imprimir os números de 1 até 10

```
program imprimir_de_1_a_10;

begin
    writeln (1);
    writeln (2);
    writeln (3);
    writeln (4);
    writeln (5);
    writeln (6);
    writeln (7);
    writeln (8);
    writeln (9);
    writeln (10);
end.
```

E se o problema fosse imprimir os números de 1 até 100 milhões?

Outro problema

Ler um número inteiro positivo **n** do teclado e
imprimir todos os números inteiros entre **1** e **n**

- A solução anterior é impossível de se adaptar para este caso!
- Em tempo de edição do código, não conhecemos o valor de **n**
- É preciso encontrar solução melhor

Repetição de comandos

- Repetir comandos é fundamental em computação
- O fluxo de execução dos comandos é alterado de maneira controlada
- Aqui começa o estudo do conceito de **lógica de programação**

O comando `while`

```
while { expressao_booleana } do
    { algum_comando };
```

- **Sintaxe** versus **Semântica**
  - Acima está a sintaxe do comando `while`
  - A semântica é o significado do comando

O comando `while`

```
while { expressao_booleana } do
    { algum_comando };
```

- A expressão acima significa que este `{ algum_comando };` será repetido um certo número de vezes, ou nenhuma vez
- Quem define se o comando será repetido, e até quando será repetido, é a `{ expressao_booleana }`
- Enquanto esta `{ expressao_booleana }` for verdadeira o `{ algum_comando };` será executado

Exemplo

```
1  program exemplo_repeticao;
2  var i: longint;
3  begin
4      read (i);
5      while i < 0 do
6          read (i);
7  end.
```

Exemplo

```
1  program exemplo_repeticao;
2  var i: longint;
3  begin
4      read (i);
5      while i < 0 do
6          read (i);
7  end.
```

- Inicia na linha 4 lendo um número do teclado
- Na linha 5, avalia a expressão `i < 0`
- Se a avaliar **falso**
  - “Pula” para linha 7 e o programa termina
- Se a avaliação resultar **verdadeiro**
  - Executa o comando da linha 6, lendo outro número do teclado
- Volta para a linha 5 e reavalia a expressão `i < 0`

Exemplo

```
1  program exemplo_repeticao;
2  var i: longint;
3  begin
4      read (i);
5      while i < 0 do
6          read (i);
7  end.
```

O que este programa faz? Qual é o significado dele?

Exemplo

```
1  program exemplo_repeticao;
2  var i: longint;
3  begin
4      read (i);
5      while i < 0 do
6          read (i);
7  end.
```

Enquanto o usuário estiver digitando números negativos, o obriga a digitar outro.

Termina quando for digitado um número positivo ou nulo.

Exemplo

```
1  program exemplo_repeticao_2;
2  var i: longint;
3  begin
4      read (i);
5      while i < 0 do
6      begin
7          writeln ('numero negativo, digite outro');
8          read (i);
9      end;
10     writeln ('parabens! o numero ',i,' nao eh negativo');
11 end.
```

Para repetir mais de um comando é preciso colocá-los entre **begin** e **end.**

Exemplo

```
mrscassenote@swift3:~/algoritmos$ ./exemplo_repeticao_2
-1
numero negativo, digite outro
-4
numero negativo, digite outro
9
parabens! o numero 9 nao eh negativo
mrscassenote@swift3:~/algoritmos$
```

- O comando da linha 7 foi executado duas vezes
  - Ele está dentro do escopo do `while`
- O comando da linha 10 foi executado somente uma vez ele
  - Ele está fora do escopo do `while`

Exemplo

```
mrscassenote@swift3:~/algoritmos$ ./exemplo_repeticao_2
9
parabens! o numero 9 nao eh negativo
mrscassenote@swift3:~/algoritmos$
```

- O comando da linha 7 não foi executado desta vez!
  - O teste resultou falso logo de cara
- O comando da linha 10 foi executado, pois ele está fora do escopo do while

Repetição de comandos

- Entender como comandos são repetidos não é suficiente para aprender a usar repetições ao resolver problemas
- É preciso entender o mecanismo de como isso pode ser utilizado para se resolver um problema particular
- Por exemplo, no problema em estudo, como repetir código pode ajudar?

Outro problema

Ler um número inteiro positivo **n** do teclado e imprimir todos os números inteiros entre **1** e **n**

- Em que a repetição ajuda aqui?
- Tipo: alguém me deu um prego e um martelo, o que eu faço agora?

Outro problema

Ler um número inteiro positivo **n** do teclado e imprimir todos os números inteiros entre **1** e **n**

- Para imprimir os número de **1** a **n**
  - Imprime o **1**
  - Depois imprime o **2**
  - Depois imprime o **3**
  - Depois imprime o **4**
  - ...
- É preciso identificar o **padrão repetitivo**
- É **como as informações se transformam de uma iteração para outra**

Outro problema

Ler um número inteiro positivo **n** do teclado e imprimir todos os números inteiros entre **1** e **n**

- Observação
  - Cada número é o anterior **mais 1**
- Isso ajuda?
  - Sim, pois se eu tenho um número, na próxima vez basta somar um nele!
  - Percebeu o **na próxima vez**?

Outro problema

Ler um número inteiro positivo **n** do teclado e imprimir todos os números inteiros entre **1** e **n**

- O **na próxima vez** indica a repetição
- Então a solução pode ser construída com base nisso
  - Imprime um número
  - Soma um a ele
  - Repete este processo

Dois problemas

- Como terminar o processo?
- Como iniciar o processo?

Como terminar o processo?

- Lembrando do enunciado, imprimir os números de **1** até **n**
- Como estamos somando um repetidas vezes, em algum momento o número vai ultrapassar **n**, certo?
- Depende!

Como terminar o processo?

- Sim, depende. O primeiro número tem que ser menor ou igual a **n**, senão o processo nunca vai terminar!
- Isso nos leva ao outro problema

Como iniciar o processo?

- Pelo primeiro número que queremos imprimir, certo?
- Mais ou menos, depende de como você construiu seu programa

Primeiro rascunho da solução

```
1  begin
2      while i <= n do
3      begin
4          writeln (i);
5          i:= i + 1;    (* transforma um numero no proximo *)
6      end;
7  end.
```

- Nos rascunhos não importa o cabeçalho do programa
- Ideia central: transformar um número no seguinte + como parar a repetição?
- Falta saber começar

Primeiro rascunho da solução

```
1  begin
2      while i <= n do
3      begin
4          writeln (i);
5          i:= i + 1;    (* transforma um numero no proximo *)
6      end;
7  end.
```

- Quando entra no laço, a primeira coisa que ocorre é a impressão do número da vez
- Então, na primeira vez, qual número queremos imprimir?
    - É o **1**, certo?
- Então tem que começar o **i** com **1** antes do laço

Uma solução

```
1  program imprimir_de_1_a_n;
2  var i, n: longint;
3
4  begin
5      read (n);
6      i:= 1;
7      while i <= n do
8      begin
9          writeln (i);
10         i:= i + 1;
11     end;
12 end.
```

Outra solução para o mesmo problema

```
1  program imprimir_de_1_a_n_v2;
2  var i, n: longint;
3
4  begin
5      read (n);
6      i:= 0;
7      while i < n do
8      begin
9          i:= i + 1;
10         writeln (i);
11     end;
12 end.
```

Duas soluções?

- Qual é a diferença entre estas soluções?
- Existem outras ou são somente estas duas?

Outra solução para o mesmo problema

```
1  program imprimir_de_1_a_n_v3;
2  var i, n: longint;
3
4  begin
5      read (n);
6      i:= n;
7      while i > 0 do
8      begin
9          writeln (n - i + 1);
10         i:= i - 1;
11     end;
12 end.
```

Você consegue pensar em outras maneiras de resolver o problema?

Exercícios

- Fazer os exercícios contidos na seção 5.10.5 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 5.7.1
- Na próxima aula veremos critérios de parada de uma repetição

Departamento de Informática – UFPR

Algoritmos e Estruturas de Dados I

Prof.ª Dr.ª Mariane Cassenote

mariane@inf.ufpr.br

Repetição de comandos