

CI1055: Algoritmos e Estruturas de Dados I

Aula 5

Carmem Hara

Julho de 2020

- 1 Contador com acumulador
- 2 Repetição com controle por entrada de dados

Vocês lembram deste exemplo?

Escrever valores de uma série

1^2 2^2 3^2 4^2 ... 100^2

```
i:= 1;  
while i <= 100 do  
begin  
    writeln( i * i );  
    i:= i + 1;  
end;
```

```
i:= 1;  
while i <= 100 do  
begin  
    termo:= i * i;  
    writeln( termo );  
    i:= i + 1;  
end;
```

Calcular o valor de uma série

Calcula a soma: $1^2 + 2^2 + 3^2 + \dots + 100^2$

	termo	soma	<i>i</i>
		0	
	1*1	1	1
+	2*2	5	2
+	3*3	14	3
+	4*4	30	4
	...		
+	100*100	338350	100

O que mudou com relação ao programa anterior?

Ao invés de escrever o valor do termo, a cada repetição calculamos o valor da **soma parcial**.

O comando repetitivo

Padrão repetitivo

```
termo:= i * i;  
soma:= soma + termo;
```

Inicialização e incremento

```
i:= 1;  
soma:= 0;  
while ... do  
begin  
    termo:= i * i;  
    soma:= soma + termo;  
    i:= i + 1;  
end;
```

Teste de parada

```
i:= 1;  
soma:= 0;  
while i <= 100 do  
begin  
    termo:= i * i;  
    soma:= soma + termo;  
    i:= i + 1;  
end;
```

Ler uma sequência terminada com zero

Ler uma sequência de valores do tipo inteiro e para cada valor escrever o seu sucessor.

```
program seqSucc;  
var  
    n: integer;  
begin  
    read( n );  
    writeln( succ( n ) );    (* ou writeln( n + 1 ); *)  
    read( n );  
    writeln( succ( n ) );  
    ...  
    read( n );              (* para quando ler zero *)  
end.
```

Qual o padrão repetitivo?

Opção 1

```
read( n );  
writeln( succ( n ));
```

OU

Opção 2

```
writeln( succ( n ));  
read( n );
```

Padrão repetitivo

```
(* inicializacao *)  
while ... do  
begin  
    read( n );  
    writeln( succ( n ));  
end;
```

Controle da Repetição

```
(* inicializacao: parece que nao precisa *)  
while ... do  
begin  
    read( n );  
    writeln( succ( n ));  
    (* incremento: parece que nao precisa *)  
end;
```

Opção 1

Teste de parada

```
(* inicializacao: parece que nao precisa *)  
while n < 0 do  
begin  
    read( n );  
    writeln( succ( n ));  
    (* incremento: parece que nao precisa *)  
end;
```

Mas não funciona. Por que?

Opção 1

Inicialização

```
n:= -1;
while n <> 0 do
begin
  read( n );
  writeln( succ( n ));
  (* incremento: parece que nao precisa *)
end;
```

Ainda não está bom. Por que?

Opção 1

Inicialização

```
n:= -1;  
while n < 0 do  
begin  
    read( n );  
    writeln( succ( n ));  
    (* incremento: parece que nao precisa *)  
end;
```

Ainda não está bom. Por que?

Solução

Opção 2

Opção 2

Padrão repetitivo

```
(* inicializacao *)  
while ... do  
begin  
    writeln( succ( n ) );  
    read( n );  
end;
```

Controle da Repetição

```
read( n );  
while ... do  
begin  
    writeln( succ( n ) );  
    read( n ); (* "incremento" *)  
end;
```

Teste de parada

```
read( n );  
while n <> 0 do  
begin  
    writeln( succ( n ) );  
    read( n );  
end;
```

Ler uma sequência terminada com zero

Ler uma sequência de valores do tipo inteiro e para cada valor escrever o seu sucessor.

```
program seqSucc;  
const  
    FIM = 0;  
var  
    n: integer;  
begin  
    read( n );  
    while n <> FIM do  
        begin  
            writeln( succ( n ) );  
            read( n );  
        end;  
end.
```

O mesmo programa para char

Ler uma sequência de caracteres terminada com ENTER e para cada caracter escrever o seu sucessor.

```
program seqSucc;  
const  
    FIM = chr( 10 );    (* ENTER *)  
var  
    n: char;  
begin  
    read( n );  
    while n <> FIM do  
        begin  
            writeln( succ( n ));  
            read( n );  
        end;  
end.
```

Soma de pares

Ler uma sequência de pares de números e, para cada par, escrever a sua soma. A sequência termina quando os valores 0 0 forem digitados.

```
program soma2;
var
    n1, n2: integer;
begin
    read( n1, n2 );
    writeln( n1, '+', n2, '=', n1 + n2 );
    read( n1, n2 );
    writeln( n1, '+', n2, '=', n1 + n2 );
    ...
    read( n1, n2 );
    writeln( n1, '+', n2, '=', n1 + n2 );
    read( n1, n2 );
end. (* para quando for 0 0 *)
```

Programa soma2

Padrão repetitivo

```
writeln( n1, '+', n2, '=', n1 + n2 );  
read( n1, n2 );
```

Controle da repetição

```
read( n1, n2 );  
while ... do  
begin  
    writeln( n1, '+', n2, '=', n1 + n2 );  
    read( n1, n2 );  
end;
```

Teste de parada

```
read( n1, n2 );  
while (n1 < 0) or (n2 < 0) do  
begin  
    writeln( n1, '+', n2, '=', n1 + n2 );  
    read( n1, n2 );  
end;
```

Dois tipos de controle diferentes

Pelo valor lido

```
program soma2Leitura;  
var  
    n1, n2: real;  
begin  
    read( n1, n2 );  
    while (n1 < 0) or (n2 < 0)  
        do  
            begin  
                writeln( n1 + n2 );  
                read( n1, n2 );  
            end;  
end.
```

Com contador

```
program soma2Contador;  
var  
    cont: integer;  
    n1, n2: real;  
begin  
    cont:= 1;  
    while cont <= 20 do  
        begin  
            read( n1, n2 );  
            writeln( n1 + n2 );  
            cont:= cont + 1;  
        end;  
end.
```

Note a diferença no padrão repetitivo.

- 1 Ler uma sequência de inteiros terminada com zero. Para cada valor escrever "par: true" caso o inteiro seja par e "par: false", caso contrário.
- 2 Escrever um programa que leia 3 valores (a, b, c) que são os coeficientes da expressão $ax^2 + bx + c$. Em seguida, o programa deve ler uma sequência de valores para x e, para cada valor, escrever o valor da expressão. A sequência termina quando o valor digitado para x for zero.
- 3 Ler uma sequência de linhas, cada uma contendo 3 valores do tipo real. Para cada linha, escrever os números na ordem inversa. O programa termina quando a entrada for "0 0 0". Exemplo:

```
1.5  3.2  0.77  
0.77 3.2  1.5  
0 0 0
```

4. Escreva um programa que leia um inteiro $n > 0$ e **para todos os valores i no intervalo $[1, n]$** escreva o fatorial de i . Por exemplo, se n for 5 saída do programa deve ser:

$$1! = 1$$

$$2! = 2$$

$$3! = 6$$

$$4! = 24$$

$$5! = 120$$