INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

Algoritmos

Prof. Jonatas Bastos

Email: jonatasfbastos@gmail.com

Site: http://jonatasfbastos.wordpress.com/

Estrutura de Repetição

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Computadores não reclamam por executar alguma tarefa, nem se cansam em fazer a mesma coisa. Por estas vantagens, podemos ensinar o computador a fazer determinadas tarefas varias vezes, sem preocupar com a fadiga;
- É com base nesta possibilidade que podemos incluir nos algoritmos as estruturas de repetição;
- Uma repetição também é chamada de laço ou loop;
- Loop, é uma estrutura que permite executar um trecho de um algoritmo várias vezes seguidas;
- Veremos dois tipos de estrutura básica:
  - Repetição com teste no início;
  - Repetição com teste no final;

Repetição com Teste no Início

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- **Estrutura de controle de fluxo que permite repetir diversas vezes o mesmo trecho de algoritmo, porém, sempre verificando antes de cada execução se é "permitido" executar o mesmo trecho.**
- **Para repetição com teste no início usamos a estrutura "enquanto";**
- **Ex:**

```
enquanto <condicao> faca
        c1;
        c2;
        .
        .
        .
        cn;
fimenquanto;
```

Repetição com Teste no Início

- **Ex: fazer um algoritmo para ler diversos números informados pelo usuário e após cada leitura exibir se o número é par ou ímpar, se o usuário informar um número negativo encerrar o programa;**

```
programa numero
var
        num, num2, num3, ?????: inteiro;
inicio
        escreva ("Informe um número: ");
        leia(num);

        se ( (num mod 2) = 0 ) entao
                escreva("Número par");
        senão
                escreva("Número impar");

        escreva ("Informe um número: ");
        leia(num1);

        se ( (num1 mod 2) = 0 ) entao
                escreva("Número par");
        senão
                escreva("Número impar");

        escreva ("Informe um número: ");
        leia(num2);

        se ( (num2 mod 2) = 0 ) entao
                escreva("Número par");
        senão
                escreva("Número impar");
        ??????
fim.
```

Problema é que não sabemos quantos dados serão fornecidos;

Repetição com Teste no Início

- **Ex: fazer um algoritmo para ler diversos números informados pelo usuário e após cada leitura exibir se o número é par ou ímpar, se o usuário informar um número negativo encerrar o programa;**

```
programa numero
var
        num: inteiro;
inicio
        escreva ("Informe um número: ");
        leia(num);

        enquanto ( num >= 0 ) faca
                se ( (num mod 2) = 0 ) entao
                        escreva("Número par");
                senão
                        escreva("Número impar");

                escreva ("Informe um número: ");
                leia(num);

        fimenquanto;
fim.
```

Variáveis contadoras

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- **Uma variável é contadora quando tem por característica armazenar dentro de si um número referente a uma certa quantia de elementos ou iterações.**
- **Ex: imprimir os números inteiros de 1 a 100;**

```
programa numero
var
        valor: inteiro;
inicio
        escreva ("Números inteiros de 1 a 100: ");
        valor <- 1;
        enquanto ( valor <= 100 ) faca
                escreva (valor);
                valor <- valor + 1;
        fimenquanto;
fim.
```

Variáveis contadoras

- **Ex: Ler 30 números inteiros fornecidos pelo usuário, e exibir depois a quantidade de números impares informados;**

```
programa numero
var
        num, cont, quantidadeImpar: inteiro;
inicio
        cont <- 1;
        quantidadeImpar <- 0;

        enquanto ( cont <= 30) faca
                escreva ("Informe um número: ");
                leia(num);

                se ( (num mod 2) <> 0 ) entao
                        quantidadeImpar <-quantidadeImpar + 1;
                fimse;
                cont <- cont + 1;
        fimenquanto;
         escreva(` a quantidade de números impares é `, quantidadeImpar);

fim.
```

Variáveis Acumuladoras

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Uma variável é chamada de acumuladora quando tem por característica armazenar dentro de si o resultado acumulado de uma série de valores, em geral uma soma;
- Quando precisamos armazenar a soma de uma quantidade pequena, a atribuição é direta, como por exemplo num1 + num2 + nume3;
- Já em uma repetição podemos armazenar a soma de diversos números sucessivamente;

Variáveis Acumuladoras

- **Ex: Calcular a soma de diversos números reais informados pelo usuário. A entrada de dados termina com o número -999;**

```
programa soma
var
        num, soma: inteiro;
inicio
        escreva ("Informe um número: ");
        leia(num);
        soma <- 0;
        enquanto ( num <> -999) faca
                soma <- soma + num;
                escreva("Informe um número: ");
                leia(num);

        fimenquanto;
        escreva(' a soma foi ', soma);
fim.
```

Repetição com teste no final

- **Para realizar a repetição com teste no final, utilizamos a estrutura repita, que permite que um bloco ou ação primitiva seja repetido até que uma determinada condição seja verdadeira.**
- **Ex:**

```
repita
        c1;
        c2;
        .
        .
        .
        cn;
ate <condicao>;
```

Repetição com teste no final

- **Neste tipo de estrutura o bloco de comandos (c1...cn) é executado pelo menos uma vez, independente da validade da condição;**
- **Ex: fazer um algoritmo para ler diversos números informados pelo usuário e após cada leitura exibir se o número é par ou ímpar, se o usuário informar um número negativo encerrar o programa;**

```
programa numero
var
        num: inteiro;
inicio
        repita
                escreva ("Informe um número: ");
                leia(num);
                se ( (num mod 2) = 0 ) entao
                        escreva("Número par");
                senão
                        escreva("Número impar");
        ate num< 0;

fim.
```

Repetição com variável de controle

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- **Quando executamos um bloco um número repetido de vezes previamente conhecido, utilizamos a estrutura para;**

```
para v de vi ate vf passo p faca
        c1;
        c2;
        .
        .
        .
        cn;
fimpara;
```

Repetição com variável de controle

- **Elabore um algoritmo que efetue a soma de todos os números ímpares que são múltiplos de 3 e que se encontram no conjunto dos números de 1 até 500.**

```
programa numeros
var
        somaInteiros, v: inteiro;
inicio
        si <- 0;
        para v de 1 ate 50 passo 1 faca
                se ( (v mod 2) = 1 ) entao
                        inicio
                                se ((v mod 3) = 0 ) entao
                                        somaInteiros <- somaInteiros + v;
                                fimse;
                        fim;
                fimse;
        fimpara;
        escreva( "Soma =", somaInteiros);
fim.
```

Contagem Regressiva

- **Elabore um algoritmo que simule uma contagem regressiva de 10 minutos, ou seja, mostre 10:00, e então 9:59, 9:58, ..., 9:00, 8:59, até 0:00;**

```
programa contagemRegressiva
var
        min, seg: inteiro;
inicio
        escreva("10:00");
        para min de 9 ate 0 passo -1 faca
                para seg de 59 ate 0 passo -1 faca
                        escreva( mim, ":", seg);
                fimpara;
        fimpara;
fim.
```

Cuidado: Laços Infinitos

- **Ao trabalhar com repetições é preciso ter cuidado para não criar laço infinito, isto é, uma laço cuja decisão de controle nunca provoque o seu término;**
- **Neste tipo de situação a maquina permanecerá executando o laço *ad eternum*, só finalizando mediante alguma intervenção externa;**

```
programa infinito
var
        valor, quandrado: inteiro;
inicio
        escreva("Números inteiros de 1 a 100");
        valor <- 1;
        repita
                quadrado <- pot(valor, 2);
                escreva(quadrado);
                valor <- valor – 1;
        ate valor > 10;
        escreva("Fim da Impressão");
fim.
```

Comparação entre estruturas de repetição

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Estrutura	Condição	Quantidade de Execuções	Condição de Existência
Enquanto	Início	0 ou muitas	condição verdadeira
Repita	Final	mínimo 1	condição falsa
Para	não tem	((vf-vi) div p)+1	v <= vf