Noções de Lógica e Programação

Aula 09

Estruturas de Repetição

Quantidade de Repetições Pré-Determinada

Quantidade de Repetições Indeterminada

Quantidade de Repetições Pré-determinada

- Para calcular a média aritmética de 60 alunos utilizamos um laço de repetição com uma **quantidade** de repetições **pré-determinada** (o laço executa 60 vezes).

Quantidade de Repetições Pré-determinada

- Para calcular a média aritmética de 60 alunos utilizamos um laço de repetição com uma **quantidade** de repetições **pré-determinada** (o laço executa 60 vezes).

- para isso utilizamos o conhecimento a priori de que possuíamos 60 alunos.

- o laço de repetição utiliza um **contador** que **realiza** o **processo** de **contagem**, e interrompe a execução do laço quando o limite de 60 repetições é atingido.

```
Algoritmo: Média_Aritmética_60_Alunos
        real N1, N2, N3, N4, MA;
        inteiro cont;
        cont = 1;
        enquanto (cont <= 60) faça
                escreva("Digite as 4 notas do aluno n° ", cont);
                leia(N1, N2, N3, N4);
                MA = (N1 + N2 + N3 + N4) / 4;
                escreva("média do aluno n° ", cont, " = ", MA);
                se (MA >= 6.0) então
                        escreva ("Aluno n° ", cont, "Aprovado!");
                senão
                        escreva ("Aluno n° ", cont, "Reprovado!");
                fim-se
                cont = cont + 1;
        fim-enquanto
Fim-algoritmo
```

Quantidade de Repetições Pré-determinada

- E para **calcular** a **média** aritmética de uma **quantidade** de alunos **informada** pelo **usuário** (último exercício aula passada)...
  - Neste caso, o **laço** de **repetição** será **executado** uma **quantidade pré-determinada** de vezes?

```
Algoritmo: Média_Aritmética_Alunos
        real N1, N2, N3, N4, MA, MT, meta;
        inteiro cont, qtdAlunos;
        escreva (“Digite a quantidade de Alunos e a média da Instituição: ”);
        leia(qtdAlunos, meta);
        se (qtdAlunos > 0) então
                cont = 1;
                MT = 0.0;
                enquanto (cont <= qtdAlunos) faça
                        escreva (“Digite as 4 notas do aluno n° ”, cont);
                        leia (N1, N2, N3, N4);
                        MA = (N1 + N2 + N3 + N4) / 4;
                        MT = MT + MA;
                        se (MA >= meta) então
                                escreva (“”Aluno n° ”, cont, “ = “, MA, “Aprovado!”);
                        senão
                                escreva (“”Aluno n° ”, cont, “ = “, MA, “Reprovado!”);
                        fim_se
                        cont = cont + 1;
                fim-enquanto
                ....
```

Quantidade de Repetições Pré-determinada

- E para **calcular** a **média** aritmética de uma **quantidade** de alunos **informada** pelo **usuário** (último exercício aula passada)...
  - Neste caso, o **laço** de **repetição** será **executado** uma **quantidade pré-determinada** de vezes?
  - **SIM!**
  - O laço será repetido **qtdeAlunos** vezes!

```
inteiro cont, qtdAlunos;
escreva ("Digite a quantidade de Alunos e a média da
Instituição ");
leia (qtdAlunos, meta);
se (qtdAlunos > 0) então
        cont = 1;
        MT = 0.0;
        enquanto (cont <= qtdAlunos) faça
                escreva ("Digite as 4 notas do aluno n° ",
                cont);
                leia (N1, N2, N4, N4);
        ....
```

Quantidade de Repetições Indeterminada

- E se **não soubermos quantos** seriam os **alunos**, o que faríamos para controlar o laço de repetição?
  - Ou seja, não sabemos que o laço será executado **qtdAlunos** vezes!

Quantidade de Repetições Indeterminada

- E se **não soubermos quantos** seriam os **alunos**, o que faríamos para controlar o laço de repetição?
  - Ou seja, não sabemos que o laço será executado **qtdAlunos** vezes!
  - Precisamos de um **laço** de **repetição** que fosse executado por uma **quantidade indeterminada** de vezes.

Quantidade de Repetições Indeterminada

- E se **não soubermos quantos** seriam os **alunos**, o que faríamos para controlar o laço de repetição?
  - Ou seja, não sabemos que o laço será executado **qtdAlunos** vezes!
  - Precisamos de um **laço** de **repetição** que fosse executado por uma **quantidade indeterminada** de vezes.
  - Devemos **encontrar outro critério** de **parada** (*diferente do processo de contagem*) que possibilite que o laço seja **finalizado** após calcular a média do último aluno (independente de quantos sejam).

Quantidade de Repetições Indeterminada

- Imagine que desejamos calcular a média anual de uma turma de número desconhecido de alunos.
  - **Dados de entrada são**: uma quantidade indeterminada de médias de alunos (1 média anual por aluno).

Quantidade de Repetições Indeterminada

- Imagine que desejamos calcular a média anual de uma turma de número desconhecido de alunos.
  - **Dados de entrada são**: uma quantidade indeterminada de médias de alunos (1 média anual por aluno).
  - **Dados de saída**: a média anual da turma.

Quantidade de Repetições Indeterminada

- Imagine que desejamos calcular a média anual de uma turma de número desconhecido de alunos.
  - **Dados de entrada são**: uma quantidade indeterminada de médias de alunos (1 média anual por aluno).
  - **Dados de saída**: a média anual da turma.
  - **Processamento**:
    - O algoritmo deve contar quantas médias de alunos foram lidas para calcular a média da turma.
    - Podemos ler uma quantidade de médias indeterminada utilizando um valor pré-definido como finalizador (a ser informado após a última média).
    - Usaremos como finalizador o valor -1, que, quando encontrado, encerra o laço sem ter seu valor computado.

Quantidade de Repetições Indeterminada

```
Algoritmo: MediaTurma1
        real MA, MT;
        inteiro cont;
        cont = 0;                   //inicialização do contador
        MT = 0.0;                   //inicialização do acumulador
        MA = 0.0;                   //inicialização da variável de leitura
        enquanto (MA != -1) faça
                escreva (“Digite a media do aluno n°”, cont + 1);
                leia (MA);
                se (MA != -1) então     //evita acumulação do finalizador em MT
                        MT = MT + MA;
                        cont = cont + 1;
                fim-se
        fim-enquanto
        se (cont > 0) então         //houve, pelo menos uma execução
                MT = MT / cont;
                escreva (“Média anual da turma = ”, MT);
        senão
                escreva (“Nenhuma média foi informada”);
        fim-se
Fim-algoritmo
```

Quantidade de Repetições Indeterminada

```
Algoritmo: MediaTurma2
        real MA, MT;
        inteiro cont;
        cont = 0;                   //inicialização do contador
        MT = 0.0;                   //inicialização do acumulador
        escreva (“Digite a media do aluno n°”, cont + 1);
        leia (MA);
        enquanto (MA != -1) faça
                MT = MT + MA;
                cont = cont + 1;
                escreva (“Digite a media do aluno n°”, cont + 1);
                leia (MA);
        fim-enquanto
        se (cont > 0) então         //houve, pelo menos uma execução
                MT = MT / cont;
                escreva (“Média anual da turma = ”, MT);
        senão
                escreva (“Nenhuma média foi informada”);
        fim-se
Fim-algoritmo
```

Exercício 1

- Seja N, a série definida a seguir:

$$N = 1 + \frac{2}{4} + \frac{3}{9} + \frac{4}{16} + \frac{5}{25} + \frac{6}{36} \ldots$$

A. Escreva um algoritmo em pseudocódigo que calcule o valor da série N para os 100 primeiros termos (quantidade de repetições pré-determinada).

B. Escreva um algoritmo em pseudocódigo que calcule o valor da série N com precisão de 0.001 (quantidade de repetições indeterminada).

Exercício 1 – Respostas

```
Algoritmo: SerieN_A
        inteiro cont;
        real N, termo;

        N = 0.0;
        cont = 1;
        enquanto (cont <= 100) faça
                termo = cont / pow(cont,2);
                N = N + termo;
                cont = cont + 1;
        fim-enquanto
        escreva (“Serie N = ”, N);
Fim-algoritmo
```

```
Algoritmo: SerieN_B
        inteiro cont;
        real N, termo;

        N = 0.0;
        cont = 1;
        termo = cont / pow(cont,2);
        enquanto (termo >= 0.001) faça
                N = N + termo;
                cont = cont + 1;
                termo = cont / pow(cont,2);
        fim-enquanto
        escreva (“Serie N = ”, N);
Fim-algoritmo
```

Exercício 2

- Construa um algoritmo que calcule a média aritmética de um conjunto de números pares que forem fornecidos pelo usuário. O valor de finalização será a entrada do número 0. Observe que nada impede que o usuário forneça quantos ímpares quiser, com a ressalva de que eles não poderão ser acumulados.

Exercício 2 – Resposta A

```
Algoritmo: MediaPares
    inteiro N, somaPares, qtdPares;
    real mediaPares;
    qtdPares = 0;
    somaPares = 0;
    N = 1;
    enquanto (N != 0) faça
        escreva (“Digite um número: ”);
        leia (N);
        se ((N != 0) e (N % 2 == 0)) então
            somaPares = somaPares + N;
            qtdPares = qtdPares + 1;
        fim-se
    fim-enquanto
    se (qtdPares > 0) então
        mediaPares = somaPares / qtdPares;
        escreva (“Média dos números pares digitados = ”, mediaPares);
    senão
        escreva (“nenhum número par foi fornecido!”);
    fim-se
Fim-algoritmo
```

Exercício 2 – Resposta B

```
Algoritmo: MediaPares2
        inteiro N, somaPares, qtdPares;
        real mediaPares;
        qtdPares = 0;
        somaPares = 0;
        escreva (“Digite um número: ”)
        leia (N);
        enquanto (N != 0) faça
                se (N % 2 == 0) então
                        somaPares = somaPares + N;
                        qtdPares = qtdPares + 1;
                fim-se
                escreva (“Digite um número: ”)
                leia (N);
        fim-enquanto
        se (qtdPares > 0) então
                mediaPares = somaPares / qtdPares;
                escreva (“Média dos números pares digitados = ”, mediaPares);
        senão
                escreva (“nenhum número par foi fornecido!”);
        fim-se
Fim-algoritmo
```

Exercício 3

- Escreva uma algoritmo que leia um conjunto de 20 números inteiros positivos e mostre qual foi o maior e menor valor fornecido.

Exercício 3 – Resposta A

```
Algoritmo: MaiorMenor1
        inteiro n, menor, maior, cont;
        cont = 1;

        enquanto (cont <= 20) faça
                escreva (“Digite um número: ”);
                leia (n);
                se (cont == 1) então
                        menor = n;
                        maior = n;
                senão
                        se (n < menor) então
                                nenor = n;
                        senão
                                se (n > maior) então
                                        maior = n;
                                fim-se
                        fim-se
                fim-se
                cont = cont + 1;
        fim-enquanto
        escreva (“menor número = ”, menor);
        escreva (“maior número  = “, maior);
Fim-algoritmo
```

```
Algoritmo: MaiorMenor1
        inteiro n, menor, maior, cont;
        cont = 0;

        enquanto (cont < 20) faça
                escreva (“Digite um número: ”);
                leia (n);
                se (cont == 0) então
                        menor = n;
                        maior = n;
                senão
                        se (n < menor) então
                                nenor = n;
                        senão
                                se (n > maior) então
                                        maior = n;
                                fim-se
                        fim-se
                fim-se
                cont = cont + 1;
        fim-enquanto
        escreva (“menor número = ”, menor);
        escreva (“maior número  = “, maior);
Fim-algoritmo
```

Exercício 3 – Resposta B

```
Algoritmo: MaiorMenor2
        inteiro n, menor, maior, cont;

        escreva (“Digite um número: ”);
        leia (n);
        menor  = n;
        maior  = n;
        cont = 2;
        enquanto (cont  <= 20) faça
                escreva (“Digite um número: ”);
                leia (n);
                se (n < menor) então
                        nenor = n;
                senão
                        se (n > maior) então
                                maior = n;
                        fim-se
                fim-se
                cont = cont + 1;
        fim-enquanto
        escreva (“menor número = ”, menor);
        escreva (“maior número  = “, maior);
Fim-algoritmo
```

```
Algoritmo: MaiorMenor2
        inteiro n, menor, maior, cont;

        escreva (“Digite um número: ”);
        leia (n);
        menor  = n;
        maior  = n;
        cont = 1;
        enquanto (cont  < 20) faça
                escreva (“Digite um número: ”);
                leia (n);
                se (n < menor) então
                        nenor = n;
                senão
                        se (n > maior) então
                                maior = n;
                        fim-se
                fim-se
                cont = cont + 1;
        fim-enquanto
        escreva (“menor número = ”, menor);
        escreva (“maior número  = “, maior);
Fim-algoritmo
```

Exercício 4

- Escreva uma algoritmo que leia um conjunto indeterminado de números inteiros positivos e mostre qual foi o maior e menor valor fornecido. O fim da entrada de dados é sinalizado pelo valor -1.

Exercício 4 – Resposta A e B

```
Algoritmo: MaiorMenor1
        inteiro n, menor, maior, cont;
        cont = 0;
        n = 0;
        enquanto (n != -1) faça
                escreva (“Digite um número: ”);
                leia (n);
                se ((cont == 0) e (n != -1) ) então
                        menor = n;
                        maior = n;
                        cont = cont + 1;
                senão
                        se ((n < menor ) e (n != -1) ) então
                                nenor = n;
                                cont = cont + 1;
                        senão
                                se ((n > maior) e (n != -1)) então
                                        maior = n;
                                        cont = cont + 1;
                                fim-se
                        fim-se
                fim-se
        fim-enquanto
        se (cont > 0) então
                escreva (“menor número = ”, menor);
                escreva (“maior número  = “, maior);
        senão
                escreva (“nenhum número foi informado”);
        fim-se
Fim-algoritmo
```

```
Algoritmo: MaiorMenor2
        inteiro n, menor, maior;
        escreva (“Digite um número: ”);
        leia (n);
        se (n != -1 ) então
                menor = n;
                maior = n;
                enquanto (n != -1) faça
                        se (n < menor) então
                                nenor = n;
                        senão
                                se (n > maior) então
                                        maior = n;
                                fim-se
                        fim-se
                        escreva (“Digite um número: ”);
                        leia (n);
                fim-enquanto
                escreva (“menor número = ”, menor);
                escreva (“maior número  = “, maior);
        senão
                escreva (“Nenhum número foi informado”);
        fim-se
Fim-algoritmo
```

Exercício 5

- Suponha que o comando rand() retorna um número aleatório pertencente ao conjunto dos números naturais. Faça um algoritmo que realize o sorteio de N números aleatórios entre 1 e 61, uma quantidade indeterminada de vezes. O fim do algoritmo é sinalizado quando o valor de N é 0. Antes de acabar o algoritmo dê tchau para o usuário.
  - Ex:

    Digite quantos números você quer sortear entre 1 e 61: **8**

    Números sorteados: 12 65 2 24 35 42 10 56

    Digite quantos números você quer sortear entre 1 e 61: **2**

    Números sorteados: 29 61

    Digite quantos números você quer sortear entre 1 e 61: **0**

    Tchau!

Exercício 5

```
Algoritmo: sorteio
    inteiro N, X;
    escreva (“Digite quantos números você quer sortear entre 1 e 61: ”);
    leia (N);
    enquanto (N > 0) faça
        escreva(“Números sorteados: ”);
        enquanto (N > 0) faça
            X = (rand() % 61) + 1;
            escreva (X);
            N = N – 1;
        fim-enquanto
        escreva (“Digite quantos valores você quer sortear entre 1 e 61: ”);
        leia (N);
    fim-enquanto
    escreva (“tchau!”);
Fim-algoritmo
```