Lógica de Programação

Estruturas de Repetição

Roni Schanuel
23-03-2023

Firjan SENAI

Estruturas de Repetição

Servem para executar um trecho de código em um número determinado de vezes até que uma determinada condição seja satisfeita. As estruturas de repetição também são chamadas de laços ou loops.

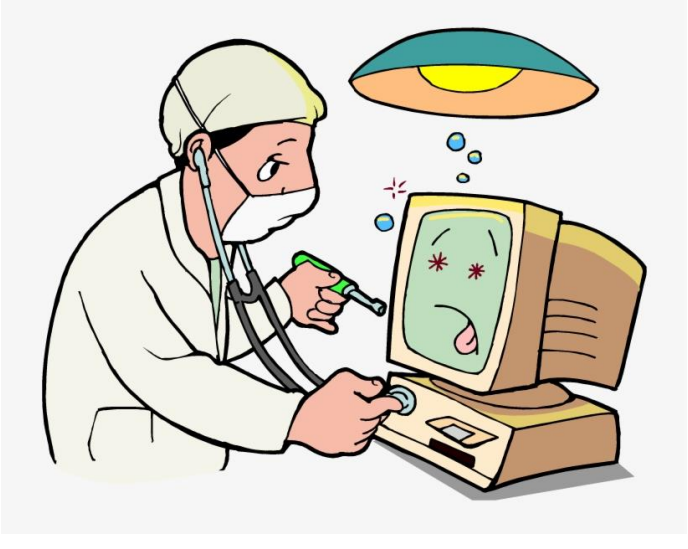
Enquanto
Enquanto o teste(condição) for verdadeiro a sequência de comandos é executada. Como o teste do enquanto é no início as instruções podem não ser executadas.

Sintaxe:

```
Enquanto (Condição) {

        //Instruções
}
```

Enquanto isso...

- Devido à pandemia do coronavírus, não poderíamos começar nossas aulas da Residência de Software **enquanto** não fôssemos notificados. =(
  - **Enquanto** isso, deveríamos ficar em casa aguardando novas notícias
  - Como seria um programa de computador que representasse esse cenário?

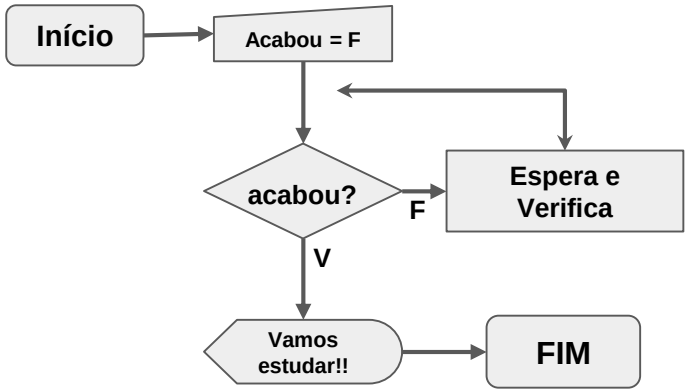


Laços de repetição

- Podemos usar **laços de repetição** para sabermos se podemos sair de casa ou não?

```
programa
{
        funcao inicio () {
                logico acabou_coronavirus = falso
            enquanto (acabou_coronavirus == falso){
                acabou_coronavirus = verifica_pandemia()
                espera(1 dia)
            }// fim enquanto
            escreva("Vamos para a Residencia de software!!")
        }// fim inicio
}// fim programa
```

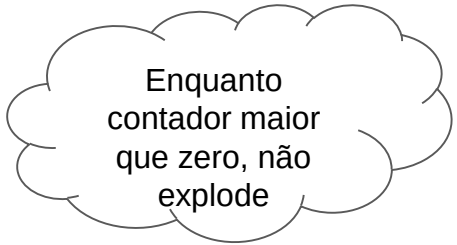
Note que o programa ainda está incompleto pois precisamos programar como verificar a pandemia



Outro exemplo

- Podemos colocar condições dentro da estrutura **enquanto**

```
programa
{
inclua biblioteca Util --> u
        funcao inicio() {
                        inteiro contador = 10
                        enquanto (contador > 0)
                        {
                                limpa()
                                escreva ("Detonação em: ", contador)
                                contador = contador - 1
                                u.aguarde(1000) // Aguarda 1000 milisegundos (1 segu
                        }
                        limpa()
                        escreva ("Booom!\n")
        }
}
```



Outro exemplo

Faça um programa usando o enquanto que escreva na tela números de 1 a 100.

```
programa
{

	funcao inicio()
	{

	inteiro numero
		numero = 1
		enquanto(numero<=100){
			escreva(numero +",")
			numero++
		}
		escreva("Fim")
	}
}
```

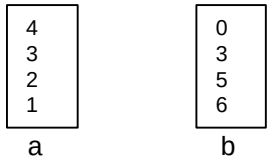
Faça o mesmo exercício usando o para.

```
programa
{

	funcao inicio()
	{

		para(inteiro numero = 1; numero<=100;numero++){
			escreva(numero +",")
		}
		escreva("Fim")
	}
}
```

Teste de Mesa

É uma simulação em papel de como será o processamento e resultado do algoritmo.

```
programa
{

    funcao inicio()
    {
        inteiro a = 4, b=0
        enquanto(a > 1){
            a = a - 1
            b = b + a
            escreva("a-",a,"\n")
            escreva("b-",b,"\n")
        }
    }
}
```




```
a-3
b-3
a-2
b-5
a-1
b-6
```

saída na tela

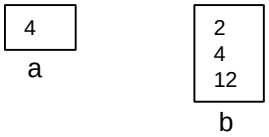
Debug

Permite a execução do programa em passos onde podemos verificar o resultado de variáveis em tempo real facilitando a descoberta de erros no código.

Podemos indicar um ponto de parada ao iniciar o Debug

Teste de Mesa

```
programa
{

    funcao inicio()
    {
        inteiro a = 4, b=2

        para(inteiro i=1; i < a; i++){
            b = b * i
            escreva(b,"\n")
        }

    }
}
```




```
2
4
12
```

saída na tela

Exercício

1) Faça outro exercício para que sejam impressos os números da seguinte forma: 0,10....90

2) Faça outro exercício para que sejam impressos os números da seguinte forma: 100,90...10

```
programa
{

    funcao inicio()
    {
        para(inteiro numero=100; numero > 0; numero=numero-10){
            escreva(numero + ",")
        }
    }
}
```

3) Escrever um programa de computador que leia números inteiros e ao final, apresente a soma de todos os números lidos até que o valor digitado seja zero.

```
programa
{

    funcao inicio()
    {

    inteiro numero, total = 0
        escreva("Digite o número:")
        leia(numero)
        enquanto(numero != 0){
            total = total + numero
            escreva("Digite o número:")
            leia(numero)
        }
        escreva("Total:" + total)
    }
}
```

O **Enquanto** é muito utilizado quando precisamos fazer testes e não sabemos quantas vezes será realizado. Em outro exemplo precisavamos fazer a leitura do número duas vezes porque a estrutura **enquanto** testa no início, neste caso podemos utilizar o **faça enquanto** que testa no final.

```
programa
{

		funcao inicio()
		{
				inteiro numero, total=0
				faca{
						escreva("Digite o número:")
						leia(numero)
						total = total + numero

				}
				enquanto(numero !=0)
				escreva("Total:" + total)


		}

}
```

- Imagine que queremos saber a tabuada de um número.
  - Quais são os requisitos?
    - Escolher um número
    - Multiplicar o número escolhido por 1 até 10
- Então **para** 1 **até** 10 **multiplique** o número escolhido.

Como fica o código??

Tabuada usando laços de repetição

O Para possui uma variável de controle, a qual podemos repetir um conjunto de instruções até um determinado número de vezes. A variável de controle é chamada de contador.

```
programa
{
        funcao inicio()
        {
                inteiro numero, resultado, contador

                escreva("Informe um número para ver sua tabuada: ")
                leia(numero)

                limpa()

                para (contador = 1; contador <= 10; contador++)
                {
                        resultado = numero * contador
                        escreva (numero, " X ", contador, " = ", resultado , "\n")
                }
        }
}
```

Note que ao usar o "para" temos uma estrutura facilitada para intervalos de repetição

Leia a idade de uma determinada quantidade de pessoas que também deverá ser informada pelo usuário e diga no final quantos são de maior e menor idade.

```
programa
{

    funcao inicio()
    {
        inteiro quantPessoas,idade,totalMaior18=0,totalMenor18=0
        escreva("Digite a quantidade de pessoas:")
        leia(quantPessoas)
    para(inteiro i=0; i<quantPessoas; i++){
        escreva("Digite a idade da pessoa:")
        leia(idade)
        se(idade >= 18){
            totalMaior18 ++
        }senao{
            totalMenor18 ++
        }
    }
    escreva("Total Maior de idade:" + totalMaior18, "\n")
    escreva("Total Menor de idade:" + totalMenor18)
    }
}
```

Sobre laços de repetição

- Se uma ação se repete em um algoritmo, em vez de escrevê-la várias vezes, em certos casos podemos resumir anotando uma vez só e solicitando que ela se repita, usando umas das **estruturas de repetição.**
- Podemos pedir que uma ação (ou um conjunto de ações) seja executada um número definido ou indefinido de vezes, ou enquanto um estado permanecer ou até que um estado seja atingido.
- Fora do Portugol, essas estruturas são denominadas do inglês , **while** (enquanto) e **for** (para)

Exercícios

Faça um programa que leia um número e apresente como resultado a multiplicação de 10 até 0.

Exemplo: 3x10 = 30

3x9 = 27

```
programa
{

    funcao inicio()
    {

    inteiro numero
        escreva("Digite o  número:")
        leia(numero)
        para(inteiro i=10; i>=0;i=i-1){
            escreva("\n",numero,"x",i,"=",numero * i)
        }
    }
}
```

Fazer um algoritmo para que seja lida uma determinada quantidade de números. O usuário deverá ser perguntado se deseja continuar (S/s) caso outro caracter for digitado o programa será finalizado somando o total dos números digitados e exibindo a média.

```
programa
{

    funcao inicio()
    {
        caracter continuar='S'
        inteiro numero, total=0, contador=0
        faca{
            escreva("Digite o número:")
            leia(numero)
            total = total + numero
            contador ++
            escreva("Deseja Continuar(S/s:)")
            leia(continuar)
        }enquanto(continuar == 'S' ou continuar =='s')
            escreva("Total:" + total,"\n")
            escreva("Média:" + total/contador)
    }
}
```

Exercício

Faça um exercício para leitura de dados de uma eleição
1 - Candidato - X
2 - Candidato - Y
3 - Branco
0 - Encerrar Votação

Qualquer opção diferente anulará o voto
No final deverá ser exibido o total de votos e o percentual de voto de todos candidados