TRILHA KOTLIN
CAMPINAS - TECH

venturus

DEVELOPING THE FUTURE

ESTRUTURAS DE REPETIÇÃO

São elas:

- Validação no Inicio: **ENQUANTO-FAÇA**
- Validação no final: **REPITA-ATE**
- Repetição com controle: **PARA-FAÇA**
- Contadores e Acumuladores

Validação no início : enquanto-faça

enquanto-faça: antes de entrar na estrutura de repetição, uma expressão lógica é avaliada e caso o resultado da mesma for verdadeiro, os comandos que estão dentro da estrutura serão executados. Após a execução dos comandos, a expressão lógica é novamente avaliada. Caso o resultado da expressão lógica for falso, o algoritmo sai da estrutura de repetição e segue para a próxima linha

Validação no início : enquanto-faça

enquanto <expressão-lógica> **faça:**

    <bloco de comandos>

fim-enquanto

Validação no início : enquanto-faça

```
var valor: inteiro
escreval("Digite um valor : ")
leia(valor)
enquanto (valor <> 0) faca
    escreval("valor digitado foi: ", valor)
    escreval("Digite um valor : ")
    leia(valor)
fimenquanto
```

Validação no fim : faça-enquanto

A estrutura **faça-enquanto** difere da estrutura enquanto-faça somente por executar o bloco de comando antes de testar se a condição é verdadeira, ou seja, o teste da condição é realizado apenas ao final da estrutura.

Assim, utilizando o **faça-enquanto** o bloco de comandos será sempre executado pelo menos uma vez, mesmo que a expressão de controle seja falsa.

Validação no fim : repita-ate

repita

<bloco de comandos>

ate <expressão-lógica>

Validação no fim : repita-ate

```
var   auxAluno, continuar, nome : caractere
Inicio
  repita
  escreval("Digite o nome do Aluno: <ENTER>")
  leia(nome)
  auxAluno <- nome
  escreval("Deseja continuar? (S/N)")
  leia(continuar)
  ate continuar = "N"
  escreval("Ultimo nome digitado: ", auxAluno)
Fimalgoritmo
```

Controle de repetição: para-faça

A estrutura **para-faça** é composta de um mecanismo de controle que estabelece de antemão quantas vezes o laço será executado. A estrutura é mostrada no algoritmo a seguir e para ser utilizada precisa das informações referentes aos valores de **inicio**, **fim** e **incremento**. Nessa estrutura, uma determinada variável assumirá valores pertencentes ao intervalo identificado pelos valores de inicio e fim, respeitando o incremento informado.

Por exemplo a expressão i de 0 ate 10 passo 2 significa que i assumirá os valores 0,2,4,6,8,10. Nesse caso, o laço seria executado **6 vezes**.

Controle de repetição: para-faça

para variavel de **inicio** ate **fim** passo **incremento faça**

<bloco de comandos>

fim-para

Controle de repetição: para-faça

```
var j: inteiro
para j de 1 ate 100 passo 1 faca
   escreval(j)
fimpara
```

Contadores e Acumuladores

Em situações onde é necessário realizarmos contagens de ocorrências, ou somatórios e produtórios de valores dentro de um conjunto de dados, devemos utilizar variáveis específicas para fazer o armazenamento dos resultados. Chamamos de **contadores** para as variáveis que realizam a contagem de ocorrências de um determinado valor (ou situação) e de **acumuladores** para as variáveis responsáveis por armazenar os resultados de somatórios e produtórios de valores.

Contadores

Os **contadores** são normalmente inicializados com valor 0 (zero) e incrementados em 1 (um) a cada vez que uma nova ocorrência (ou situação) é observada.

var contador: **inteiro**

contador ← 0

...

contador ← contador + 1

Contadores

Por exemplo, considere que dentro de um conjunto de informações referentes a idades e sexos de 50 pessoas, desejemos saber quantas dessas pessoas são do sexo feminino e possuem 18 anos ou mais. Para isso, é necessário inserir um contador para armazenar a quantidade de ocorrências da condição definida no enunciado. Esse contador deve ser inicializado com 0 e incrementado em 1 sempre que o sexo de uma dada pessoa é feminino e sua idade é maior ou igual a 18, como no Algoritmo a seguir.

Contadores

var idade, i, n: **inteiro**

n ← 0 // a variavel n será o contador que armazenará o numero de pessoas que pertencem ao conjunto do enunciado, inicializada com valor 1

para i de **1** ate **50** passo **1 faca**

  escreval(“Digite sexo (‘M’ ou ‘F’) ”)

  leia(sexo)

  escreval(“Digite a idade da pessoa”)

  leia(idade)

Contadores

se sexo **= “M”** e idade **>=18 entao**

n ← n + 1 // aumenta em 1 a quantidade de pessoas que pertencem ao conjunto

fimse

fimpara

escreval(“A quantidade de pessoas do sexo feminino com 18 anos ou mais é : **”, n)**

Acumuladores

Como comentado anteriormente, os **acumuladores** são utilizados em dois tipos de situações, para a realização de somatórios e para a realização de produtórios. No caso dos somatórios, o acumulador é normalmente inicializado com o valor 0 e incrementado no valor de um outro termo qualquer, dependendo do problema em questão.

Acumuladores

var acumulador: **inteiro**

acumulador ← 0

...

acumulador ← acumulador + termo

Acumuladores

Considere que no problema anterior, ao invés de desejarmos calcular a quantidade de pessoas que são do sexo feminino e possuem 18 anos, desejemos calcular a soma das idades das pessoas que estão nessa situação. Nesse caso, precisamos inserir no algoritmo um acumulador, que deve ser inicializado em 0, e incrementado no valor da idade da pessoa em questão.

Acumuladores

var idade, i, soma: **inteiro**

soma ← 0 // a variavel soma irá armazenaro somatorio das idades e das pessoas que pertencem ao conjunto solicitado no enunciado, ela é inicializada com o valor 0

para i de **1** ate **9** passo **1 faca**

escreval(“Digite sexo (‘M’ ou ‘F’) ”)

leia(sexo)

escreval(“Digite a idade da pessoa”)

leia(idade)

Acumuladores

se sexo **= "M"** e idade **>=18 entao**

soma ← soma + idade // aumenta o somatorio no valor da idade da pessoa em questão

fimse

fimpara

escreval("A soma das pessoas do sexo feminino com 18 anos ou mais é : **", soma)**

EXERCÍCIOS

CRIAR UM PROGRAMA QUE REALIZE A SOMA DE UM NUMERO INTEIRO ATÉ QUE O USUARIO DIGITE 0

EX: USUARIO DIGITOU 10, USUARIO DIGITOU 20, USUARIO DIGITOU 10, USUARIO DIGITOU 0, DEVE SAIR A SOMA DO NUMERO DIGITADO: 50

EXERCÍCIOS

CRIAR UM PROGRAMA QUE ENTRE COM OS DADOS DO ALUNO NOME E 2 NOTAS REFERENTE AO BIMESTRE DE ESTUDO, VC DEVE USAR O PARA-FACA, DEPOIS CALCULAR A MEDIA DO ALUNO E DEPOIS INFORMAR SE O ALUNO PASSOU(MEDIA >=7), FICOU DE RECUPERAÇAO (>=4 OU <= 6)OU REPROVADO (MEDIA <=3).

MOSTRAR O NOME DO ALUNO, A MEDIA DAS NOTAS E INFORMAR SE ELE, PASSOU, FICOU DE RECUPERACAO OU REPROVOU.

EXERCÍCIOS

CRIAR UM PROGRAMA PARA SOLICITAR O NOME E IDADE DE UM ALUNO ATE QUE A OPÇÃO FOR “S” OU “s”

VERIFICAR QUAL O ALUNO TEM A IDADE MAIOR E GUARDAR

MOSTRAR NO FINAL O NOME E IDADE DO ALUNO QUE TEM A IDADE MAIOR, A SOMA DAS IDADES E A MEDIA DE IDADES DOS ALUNOS INFORMADOS.