RACIOCÍNIO ALGORÍTMICO

ESTRUTURAS DE REPETIÇÃO

ALGORITMO

ESTRUTURAS DE REPETIÇÃO

As estruturas de repetição, também chamadas de laços de repetição, possuem dois papéis importantes no desenvolvimento de software:

- Reaproveitamento de código
- Automatização de tarefas.

ALGORITMO

ESTRUTURAS DE REPETIÇÃO

Existem basicamente dois tipos de estruturas de repetição:

- Finitas
  - Condição pré-declarada / calculada de quantidade de execuções
- Infinitas
  - Condição lógica para finalização do laço
  - Teste condicional

ALGORITMO

LAÇOS FINITOS

Em pseudocódigo, laços finitos são representados pela estrutura PARA:

```
para v de vi até vf passo p faça
    c1;
    c2;
    ...
fimpara;
```

ALGORITMO

LAÇOS FINITOS

Exemplo de aplicação 1: *Elaborar um programa que solicita ao usuário 10 números e efetua a soma, exibindo o resultado.*

```
01. início
02.     inteiro: num, soma, contador;
03.     soma <- 0;
04.     para contador de 1 até 10 passo 1 faça
05.         escreva ("Digite um número para somar: ");
06.         leia (num);
07.         soma <- soma + num;
08.     fimpara
09.     escreva ("A soma dos números é ", soma);
10. fim.
```

ALGORITMO

LAÇOS FINITOS

Exemplo de aplicação 2: *Elaborar um programa que calcula o fatorial de um número. Ex.: Fatorial de 5 é igual a 5*4*3*2*1.*

```
01. início
02.     ingeiro: num, fatorial, contador;
03.     fatorial <- 1;
04.     escreva ("Digite um número: ");
05.     leia (num);
06.     para contador de num até 1 passo -1 faça
07.         fatorial <- fatorial * contador;
08.     fimpara
09.     escreva ("O fatorial de ", num, " é ", fatorial);
10. fim.
```

PYTHON

ESTRUTURA DE REPETIÇÃO: FOR

- Estrutura for:

```
for <<var>> in range([ini], end, [step]):
    <<comandos>>
    [break]
    [continue]
```

ini: valor inicial (incluindo)
end: valor final (excluindo)
step: valor de incremento

break: efetua a saída do loop
continue: chama a próxima iteração do loop

PYTHON

LAÇOS FINITOS

Exemplo de aplicação 3: *Elaborar um programa que solicita ao usuário seu nome, e mostra cada uma das letras do seu nome em uma linha diferente do console.*

```
01. nome = input("Por favor, digite o seu nome: ")
02. for letra in nome:
03.     print(letra)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic03.py
Por favor, digite o seu nome: Joaquim
J
o
a
q
u
i
m

Process finished with exit code 0
```

PYTHON

LAÇOS FINITOS

Exemplo de aplicação 04. *Elaborar um programa que solicita 3 número, soma-os e exibe o resultado para o usuário.*

```
01. soma = 0
02. for i in range(3):
03.     num = int(input("Digite o " + str(i + 1) + " número: "))
04.     soma += num
05. print("A soma dos números é", soma)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic04.py
Digite o 1 número: 10
Digite o 2 número: 20
Digite o 3 número: 30
A soma dos números é 60

Process finished with exit code 0
```

PYTHON

TIPO RANGE

```
range(limite_superior)
range(valor_inicial, limite_final[, passo])
```

```
range(3, 8) => [3, 4, 5, 6, 7]
range(2, 9, 2) => [2, 4, 6, 8]
range(7, 1, -1) => [7, 6, 5, 4, 3, 2]
```

PYTHON

TIPO RANGE

Exemplo de aplicação 05. *Elaborar um programa que calcula o fatorial de um número. Ex.: Fatorial de 5 é igual a 5*4*3*2*1, exibindo esta informação de como é feito o cálculo.*

```
01. fatorial = 1
02. expressao = "Expressão: "
03. num = int(input("Digite um número para o cálculo do fatorial: "))
04. for i in range(num, 0, -1):
05.     fatorial *= i
06.     expressao += str(i)
07.     if i > 1:
08.         expressao += " * "
09. print("O valor do fatorial de", num, "é", fatorial, expressao)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic05.py
Digite um número para o cálculo do fatorial: 5
O valor do fatorial de 5 é 120 Expressão: 5 * 4 * 3 * 2 * 1

Process finished with exit code 0
```

PYTHON

CONTROLE DE FLUXO

Controle de Fluxo	
comando	**Descrição**
break	Interrompe o fluxo da estrutura de repetição
continue	Força uma nova iteração a partir do ponto corrente
pass	Preenche estrutura vazia, não permitida em Python

PYTHON

BREAK

Exemplo de aplicação 06. *Elaborar um programa que solicita ao usuário 10 números e efetua a soma, exibindo o resultado. Porém, se em algum momento o número digitado for 0, interrompe o laço, mostrando a soma apenas dos valores informados até o momento.*

```
01. soma = 0
02. for i in range(10):
03.     num = int(input("Digite o " + str(i + 1) + " número: "))
04.     if num == 0:
05.         break
06.     soma += num
07. print("A soma dos números é", soma)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic06.py
Digite o 1 número: 10
Digite o 2 número: 20
Digite o 3 número: 30
Digite o 4 número: 0
A soma dos números é 60

Process finished with exit code 0
```

PYTHON

CONTINUE

Exemplo de aplicação 07. *Elaborar um programa que solicita ao usuário 10 números e efetua a multiplicação dos mesmos, exibindo o resultado. Porém, se em algum momento o número digitado for 0, pula esta iteração, para que o valor não seja zerado.*

```
01. mult = 1
02. for i in range(10):
03.     num = int(input("Digite o " + str(i + 1) + " número: "))
04.     if num == 0:
05.         continue
06.     mult *= num
07. print("A multiplicação dos números é", mult)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic07.py
Digite o 1 número: 10
Digite o 2 número: 5
Digite o 3 número: 3
Digite o 4 número: 0
Digite o 5 número: 0
Digite o 6 número: 0
Digite o 7 número: 1
Digite o 8 número: 1
Digite o 9 número: 1
Digite o 10 número: 1
A multiplicação dos números é 150

Process finished with exit code 0
```

PYTHON

ESTRUTURA DE REPETIÇÃO: WHILE

- Estrutura while:

```
while <<condição>>:
    <<comandos>>
    [break]
    [continue]
```

break: efetua a saída do loop
continue: chama a próxima iteração do loop

PYTHON

LAÇOS INFINITOS

Exemplo de aplicação 09. *Elaborar um programa que solicita ao usuário que digite indefinidamente números e efetua a soma dos mesmos, parando apenas quando o usuário digitar o número 0.*

```
01. soma = 0
02. num = -1
03. while num != 0:
04.     num = int(input("Digite um número para somar (0 finaliza): "))
05.     soma += num
06. print("A soma dos números é ", soma)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic09.py
Digite um número para somar (0 finaliza): 10
Digite um número para somar (0 finaliza): 20
Digite um número para somar (0 finaliza): 30
Digite um número para somar (0 finaliza): 0
A soma dos números é  60

Process finished with exit code 0
```

PYTHON

LAÇOS INFINITOS - BREAK

Este mesmo exemplo pode ser implementado fazendo uso de uma condição infinita, com o comando de controle de fluxo ***break***.

```
01. soma = 0
02. num = -1
03. while True:
04.     num = int(input("Digite um número para somar (0 finaliza): "))
05.     if num == 0:
06.         break;
07.     soma += num
08. print("A soma dos números é ", soma)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic09.py
Digite um número para somar (0 finaliza): 10
Digite um número para somar (0 finaliza): 20
Digite um número para somar (0 finaliza): 30
Digite um número para somar (0 finaliza): 0
A soma dos números é  60

Process finished with exit code 0
```

PYTHON

TRATAMENTO DE EXCEÇÕES

Até o presente momento, os códigos apresentados levam em consideração que o usuário está entrando os dados de forma correta, sem fazer qualquer validação.

No caso de, por exemplo, digitar um texto em uma entrada com conversão para inteiro, o resultado será:

```
num = int(input("Digite um número para somar (0 finaliza): "))
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/First.py
Digite um número para somar (0 finaliza): zero
Traceback (most recent call last):
  File "/Users/user/projects/untitled1/First.py", line 2, in <module>
    num = int(input("Digite um número para somar (0 finaliza): "))
ValueError: invalid literal for int() with base 10: 'zero'

Process finished with exit code 1
```

PYTHON

TRY...EXCEPTION

```
01. try:
02.     num = int(input("Digite um número: "))
03. except:
04.     print("Valor inválido")
```

```
01. try:
02.     num = int(input("Digite um número: "))
03. except ValueError:
04.     print("Valor inválido")
05. except:
06.     print("Outro tipo de erro ocorreu!")
```

PYTHON

TRATAMENTO DE EXCEÇÕES - INTEIROS

Exemplo de aplicação 10. *Elaborar um programa que solicita um número inteiro ao usuário, validando o mesmo, e imprime este número.*

```
01. while True:
02.     try:
03.         num = int(input("Digite um número: "))
04.         break
05.     except ValueError:
06.         print("Valor inválido")
07. print("número validado:", num)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic10.py
Digite um número: zero
Valor inválido
Digite um número: 3.4
Valor inválido
Digite um número: 3
número validado: 3

Process finished with exit code 0
```

PYTHON

TRATAMENTO DE EXCEÇÕES - FLOAT

Exemplo de aplicação 11. *Elaborar um programa que solicita um número com vírgula ao usuário, validando o mesmo, e imprime este número.*

```
01. while True:
02.     try:
03.         num = float(input("Digite um número: "))
04.         break
05.     except ValueError:
06.         print("Valor inválido")
07. print("número validado:", num)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic11.py
Digite um número: 3,2
Valor inválido
Digite um número: 3.2
número validado: 3.2

Process finished with exit code 0
```

PYTHON

TRATAMENTO DE EXCEÇÕES - OUTRAS CONDIÇÕES

Exemplo de aplicação 12. *Elaborar um programa que solicita um número inteiro ao usuário, em um intervalo entre 1 e 5.*

```
01. while True:
02.     try:
03.         num = int(input("Digite um número inteiro entre 1 e 5: "))
04.         if 1 <= num <= 5:
05.             break
06.         else:
07.             print("O número deve estar entre 1 e 5.")
08.     except ValueError:
09.         print("Valor inválido")
10. print("número validado:", num)
```

```
/Users/user/untitled1/bin/python /Users/user/projects/untitled1/ExAplic12.py
Digite um número: 3.1
Valor inválido
Digite um número: 7
O número deve estar entre 1 e 5.
Digite um número: 3
número validado: 3

Process finished with exit code 0
```

IMPLEMENTAÇÃO

CÁLCULO DE IMPOSTO DE RENDA MENSAL DA PESSOA FÍSICA (IRPF)

Requisitos:

Refatorar o exemplo anterior usando as seguintes estruturas de repetição:

- while
- for

ESTRUTURAS DE REPETIÇÃO

EXERCÍCIOS

Laços finitos

1. Criar um programa que solicita ao usuário 10 números, contando quantos são pares e quantos são ímpares. Informar ao final estas informações.

2. Criar um programa que pede ao usuário 5 números, e informa qual é o menor e qual é o maior deles.

3. Criar um programa que recebe um texto digitado pelo usuário e o imprime apenas com consoantes, removendo as vogais. Obs.: desconsiderar letras maiúsculas e acentos.

ESTRUTURAS DE REPETIÇÃO

EXERCÍCIOS

Laços finitos

4. Criar um programa que solicita um número ao usuário e exibe a tabuada deste número de 1 a 10, no formato:

```
Tabuada do n
n x 1 = n
n x 2 = 2n
...
n x 10 = 10n
```

5. Criar um programa que pede dois números ao usuário. O primeiro será o numerador, e o segundo será o expoente. A saída do programa deve ser o resultado da operação numerador elevado a expoente. Obs.: não usar função interna que calcula automaticamente potência.

ESTRUTURAS DE REPETIÇÃO

EXERCÍCIOS

Laços infinitos

6. Criar um programa que pede para o usuário inserir um login e uma senha. Caso os valores sejam iguais, informar que os dados são inválidos e pedir novamente as informações. Caso contrário, exibir a mensagem "Bem-vindo ao sistema!!!".

7. Criar um programa que solicite ao usuário vários números e, ao digitar 0, calcula a média destes números informados.

8. Criar um programa que gera a série de Fibonacci até enquanto o valor for menor que um valor informado pelo usuário. Obs.: Série de Fibonacci = 0, 1, 1, 2, 3, 5, 8 ,13, 21, 34, 55,... é formada por 0, 1 e partir deste ponto sempre será a soma dos dois valores anteriores.

ESTRUTURAS DE REPETIÇÃO

EXERCÍCIOS

Desafio I

9. Criar um programa que simula um carrinho de compras, onde solicita o nome do produto (não pode ser vazio), o valor deste produto (valor com vírgula, positivo) e a quantidade deste produto a ser comprada (valor inteiro, positivo). Ao incluir um produto, deve perguntar se o usuário deseja fechar o pedido ou incluir mais produtos. Todos os dados devem ser **validados**. Ao final da compra, deve ser informado o valor total do pedido.

ESTRUTURAS DE REPETIÇÃO

EXERCÍCIOS

Desafio II

10. Criar um Programa que simule um caixa eletrônico. O programa deverá perguntar ao usuário a valor do saque e depois informar quantas notas de cada valor serão fornecidas. A saber:

    a. Notas disponíveis: 1, 5, 10, 50 e 100 reais;
    b. Valor mínimo de saque: R$ 10,00 reais;
    c. Valor máximo de saque: R$ 600,00 reais.

    Este é o mesmo exercícios de desafio de condicionais. Porém, agora o desafio é resolver de forma mais otimizada fazendo uso de estruturas de repetição.