

Lógica de Programação

Estruturas de Decisão

Profa. Ma. Samara Souza
040601858@prof.unama.br

Estruturas de Controle

- ❑ As estruturas de entrada e saída são uma parte da estrutura de controle de fluxo, que permite que a linearidade da execução de um programa possa ser alterada.
- ❑ Entre as estruturas de controle estão:
 - Entrada e saída
 - Estrutura de decisão
 - Múltipla escolha
 - Repetição

- ❑ Estruturas de **controle de fluxo**, executando um ou vários comandos **se a condição** testada for **verdadeira** e, em alguns casos, executando um ou vários comandos se for **falsa** (LOPES e GARCIA, 2002).
- ❑ Condição é uma comparação que possui dois valores possíveis, verdadeiro ou falso (ASCENIO e CAMPOS, 2008).

Estruturas de Decisão

- ❑ Estruturar de forma automática essas decisões é fundamental para criar algoritmos simples.
- ❑ Operações Relacionais e Operações Lógicas são a base para controle de fluxo em um algoritmo.

Operadores Relacionais

Operador	Ação	Exemplo
>	Maior que	3 > 7
>=	Maior ou igual a	10 >= 2
<	Menor que	31 < 50
<=	Menor ou igual a	3 <= 9
==	Igual a	12 == 12
!=	Diferente de	31 != 13

Operadores Relacionais

- ❑ Utilizamos os operadores relacionais para comparação de valores ou uma variável que pode assumir valor booleano **True** (verdadeiro) ou **False** (falso).
- ❑ Também utilizamos os operadores relacionais para comparar caracteres, números, variáveis e expressões.

Fixação

1) $(3 + 7) / 2 > 3 + 7 / 2$

2) $5 * (3 * 4) == (5 * 3) * 4$

3) $21 / 3 >= 63 / 9$

4) $21 / 3 <= 63 / 9$

5) $21 / 3 >= 63 / 8 + 1$

A precedência é como na aritmética: primeiro os parênteses e depois multiplicações e divisões, seguido de somas e subtrações, sempre da esquerda para a direita.

Operações Lógicas

- ❑ Na álgebra de Boole existem apenas 2 estados lógicos:
 - Sim / Verdadeiro / 1
 - Não / Falso / 0
- ❑ Os operadores dessa álgebra estão implementados em linguagens de programação e diretamente em dispositivos físicos:
E, OU e NÃO:

Tabela Verdade E

Condição		Resultado
Verdadeiro	E Verdadeiro	Verdadeiro
Verdadeiro	E Falso	Falso
Falso	E Verdadeiro	Falso
Falso	E Falso	Falso

Fixação

Quais as respostas para as questões abaixo, considerando:

`X <- 5` `Y <- 10`

- 1) `X = 5` **E** `Y = 11`
- 2) `X <> 2` **E** `Y <> 11`
- 3) `X = 5` **E** `Y < 10`
- 4) `X >= 3` **E** `Y <= 12`

Tabela Verdade OU

Condição			Resultado
Verdadeiro	OU	Verdadeiro	Verdadeiro
Verdadeiro	OU	Falso	Verdadeiro
Falso	OU	Verdadeiro	Verdadeiro
Falso	OU	Falso	Falso

Fixação

Quais as respostas para as questões abaixo, considerando:

`X <- 5` `Y <- 10`

- 1) `X = 5` **OU** `Y = 11`
- 2) `X <> 2` **OU** `Y <> 11`
- 3) `X = 5` **OU** `Y < 10`
- 4) `X >= 3` **OU** `Y <= 12`

Tabela Verdade NÃO

Condição	Resultado
NÃO Verdadeiro	Falso
NÃO Falso	Verdadeiro

Fixação

Quais as respostas para as questões abaixo, considerando:

`X <- 5` `Y <- 10`

- 1) NÃO (`X = 5`)
- 2) NÃO (`X = Y`)
- 3) NÃO (`Y > 10`)
- 4) NÃO (`Y <= 10`)

Operadores Lógicos

Algoritmo	Python
E	and
OU	or
NÃO	not

Na Prática

```
a = 5
```

```
b = 15
```

```
c = 20
```

```
print("A == B AND B > C: ", a == b and b > c)
```

```
print("A < B OR B > C: ", a < b or b > c)
```

```
print("not A == B: ", not a == b)
```


- ❑ Executam uma sequência de comandos de acordo com o resultado de um teste
 - **Decisão Simples:** seleciona ou ignora uma única ação
 - **Decisão Composta:** seleciona entre duas ações ou grupos de ações diferentes

❑ Decisão Simples – Se...Então

```
se <expressão-lógica> entao  
    <sequência-de-comandos>  
fimse
```

❑ Decisão Simples – Se...Então

```
// Verificando se a idade é maior que 18
se idade > 18 então
    escreva("Você é maior de idade.")
fimse
```

❑ Decisão Simples – If

```
if <expressão-lógica>:  
    <sequência-de-comandos>
```

❑ Decisão Simples - If

```
idade = int(input("Digite a sua idade: "))  
  
if idade >= 18:  
    print("Maior Idade!")
```

Indentação

- ❑ Indentação é uma forma de arrumar o código, fazendo com que algumas linhas fiquem mais à direita que outras, à medida que adicionamos espaços em seu início.
- ❑ Enquanto na maioria das linguagens, como C, Java e PHP, os blocos de código são delimitados por chaves ({ }) ou comandos, em Python, os blocos são delimitados por espaços ou tabulações, formando uma indentação visual.
- ❑ Não existem símbolos de “abre” e “fecha”.

E se a condição for falsa?

Estruturas de Decisão - Pseudocódigo

❑ Decisão Composta

Se...Então...Senão

```
se <expressão-lógica> entao  
    <sequência-de-comandos>  
  
senao  
    <sequência-de-comandos>  
  
fimse
```


Estruturas de Decisão - Pseudocódigo

❑ Decisão Composta

Algoritmo "DecisaoSimples"

Var

idade: inteiro

Inicio

Escreva("Digite sua idade: ")

Leia(idade)

se idade >= 18 entao

Escreva("Você é maior de
idade.")

senao

Escreva("Você é menor de
idade.")

fimse

FimAlgoritmo

❑ Decisão Composta

○ if...else

```
if <expressão-lógica>:  
    <sequência-de-comandos>  
  
else:  
    <sequência-de-comandos>
```

❑ Decisão Composta

○ if...else

```
idade = int(input("Digite a sua idade: "))

if idade >= 18:
    print("Maior Idade!")
else:
    print("Menor Idade")
```

Estruturas de Decisão - Pseudocódigo

❑ Decisão Composta

```
se <expressão-lógica> entao  
    <sequência-de-comandos>  
senaoSe <expressão-lógica> entao  
    <sequência-de-comandos>  
senao  
    <sequência-de-comandos>  
fimse
```

Estruturas de Decisão - Pseudocódigo

❑ Decisão Composta

```
Algoritmo "DecisaoComposta"  
Var  
    nota: real  
Inicio  
    Escreva("Digite sua nota: ")  
    Leia(nota)  
    Se nota >= 7.0 Entao  
        Escreva("Aprovado")  
    SenaoSe nota >= 5.0 Entao  
        Escreva("Recuperação")  
    Senao  
        Escreva("Reprovado")  
    FimSe  
FimAlgoritmo
```

Estruturas de Decisão - Python

❑ Decisão Composta

If...elif...else

```
if <expressão-lógica>:  
    <sequência-de-comandos>  
elif <expressão-lógica>:  
    <sequência-de-comandos>  
else:  
    <sequência-de-comandos>
```

Estruturas de Decisão - Python

```
numero = float(input("Digite um número: "))  
  
✓ if numero > 0:  
|     print("O número", numero, "é positivo.")  
✓ elif numero < 0:  
|     print("O número", numero, "é negativo.")  
✓ else:  
|     print("O número é zero.")
```

- ❑ Faça um programa que leia um número N e imprima “F1”, “F2” ou “F3”, conforme a condição:
 - “F1”, se $N < 10$;
 - “F2”, se $N = 10$;
 - “F3”, se $N > 10$;

- ❑ Faça um algoritmo que leia um inteiro e mostre uma mensagem indicando se este número é par ou ímpar.

- ❑ Faça um programa que leia as notas da primeira e segunda avaliação de um aluno e calcule a sua média. Se a média for maior ou igual a sete mostre “Aprovado!”, caso contrário mostre “Reprovado!”.
- ❑ Faça um algoritmo/programa que receba dois números e diga se o primeiro é maior que o segundo, se são iguais ou se o segundo é maior que o primeiro.
- ❑ Escreva um programa que, para uma conta bancária, leia o seu número, o saldo, o tipo de operação a ser realizada (depósito ou retirada) e o valor da operação. Após, determine e mostre o novo saldo.

- ❑ Faça um programa que receba a idade de um nadador e imprima a sua categoria seguindo as regras:

Categoria	Idade
Infantil A	5 – 7 anos
Infantil B	8 – 10 anos
Juvenil A	11 – 13 anos
Juvenil B	14 – 17 anos
Sênior	Maiores de 18 anos

❑ Múltipla Escolha

```
escolha <expressão-de-seleção>  
caso <exp11>, <exp12>, ... <exp1n>  
    <sequência-de-comandos-1>  
caso <exp21>, <exp22>, ... <exp2n>  
    <sequência-de-comandos-2>  
outrocaso  
    <sequência-de-comandos-extra>  
fimescolha
```

Estruturas de Decisão - Pseudocódigo

Algoritmo "ExemploEscolha"

Var

 numeroDia: inteiro

Inicio

Escreva("Digite o número do dia da semana (1 a 7): ")

 Leia(numeroDia)

 Escolha numeroDia

 Caso 1

 Escreva("Domingo")

 ...

 Caso 7

 Escreva("Sábado")

 Outro caso

 Escreva("Número inválido de dia.")

 FimEscolha

FimAlgoritmo

Lógica de Programação

Estruturas de Decisão

Profa. Ma. Samara Souza
040601858@prof.unama.br