Lógica de Programação

Aula 05

Professor: Henrique Augusto Maltauro

Fecomércio PR
Sesc Senac IFPD
Senac

Pseudocódigo

- **Tipos de Dados: logico**

O **tipo de dado** de **logico** vai armazenar um **valor lógico**, também chamado de **valor booleano**.

Esse **valor lógico** armazena valores de **verdadeiro** e **falso**.

Pseudocódigo

- Tipos de Dados: logico

```
logico L1 = verdadeiro
logico L2 = falso
```

Pseudocódigo

- **Proposições**

As **proposições** são sentenças declarativas, a qual possui um **valor lógico** de **verdadeiro** ou **falso**.

A **lógica proposicional** possui **três princípios** fundamentais.

Pseudocódigo

- **Proposições**

1. **Princípio da Identidade:** **Uma proposição verdadeira é verdadeira, e uma proposição falsa é falsa.**
2. **Princípio da Não-Contradição:** **Uma proposição não pode ser verdadeira e falsa ao mesmo.**
3. **Princípio do Terceiro Excluído:** **Uma proposição ou é verdadeira ou é falsa, não existe uma terceira opção.**

Pseudocódigo

- **Proposições**

No contexto de **programação**, essas **proposições** são validadas através de **comparações**.

Normalmente, nós vamos **comparar o valor** de uma **variável** com outro **valor**, utilizando os **operadores relacionais**.

Pseudocódigo

- **Operadores Relacionais**

Dentro da **programação**, nós temos os **operadores relacionais**, para realizar uma **comparação entre duas expressões**.

O **resultado dessa comparação** sempre será um **valor lógico**, ou de **verdadeiro** ou de **falso**.

Pseudocódigo

- **Operadores Relacionais**

Temos 6 **operadores relacionais**:

- **Igual a ==**
- **Diferente de !=**
- **Maior que >**
- **Menor que <**
- **Maior ou igual a >=**
- **Menor ou igual a <=**

Pseudocódigo

- **Operadores Relacionais: Igual a ==**

O **operador relacional** igual a, vai comparar se **duas expressões são iguais**.

Representado pelo sinal de **duplo igual ==**.

Pseudocódigo

- Operadores Relacionais: Igual a ==

```
inteiro numero
leia(numero)
logico log = (numero == 5)
```

Pseudocódigo

- **Operadores Relacionais: Diferente de !=**

O **operador relacional** diferente de, vai comparar se **duas expressões são diferentes**.

Representado pelo sinal de **exclamação e igual !=**.

Pseudocódigo

- Operadores Relacionais: Diferente de !=

```
inteiro numero
leia(numero)
logico log = (numero != 5)
```

Pseudocódigo

- **Operadores Relacionais: Maior que >**

O **operador relacional** maior que, vai comparar se a **primeira expressão é maior que a segunda**.

Representado pelo sinal de **maior >**.

Pseudocódigo

- Operadores Relacionais: Maior que >

```
inteiro numero
leia(numero)
logico log = (numero > 5)
```

Pseudocódigo

- **Operadores Relacionais: Menor que <**

O **operador relacional** menor que, vai comparar se a **primeira expressão é menor que a segunda**.

Representado pelo sinal de **menor <**.

Pseudocódigo

- Operadores Relacionais: Menor que <

```
inteiro numero
leia(numero)
logico log = (numero < 5)
```

Pseudocódigo

- **Operadores Relacionais: Maior ou igual a >=**

O **operador relacional** maior ou igual a, vai comparar se a **primeira expressão é maior ou igual a segunda**.

Representado pelo sinal de **maior e igual >=**.

Pseudocódigo

- Operadores Relacionais: Maior ou igual a **>=**

```
inteiro numero
leia(numero)
logico log = (numero >= 5)
```

Pseudocódigo

- **Operadores Relacionais: Menor ou igual a <=**

O **operador relacional** menor ou igual a, vai comparar se a **primeira expressão é menor ou igual a segunda**.

Representado pelo sinal de **menor e igual <=**.

Pseudocódigo

- Operadores Relacionais: Menor ou igual a <=

```
inteiro numero
leia(numero)
logico log = (numero <= 5)
```

Pseudocódigo

- **Proposições**

Voltando as **proposições**, nós então precisamos compreender que podemos ter **dois tipos de proposições**.

As **proposições simples**, da forma como vimos até agora.

Ou, **proposições compostas**, que vão **unir várias proposições simples** em uma só, e para isso, fazemos uso dos **operadores lógicos**.

Pseudocódigo

- **Operadores Lógicos**

Dentro da **programação**, nós temos os **operadores lógicos**, para realizar **operações lógicas**, unindo duas ou mais **proposições**.

No caso do **pseudocódigo**, temos apenas dois **operadores lógicos**:

- **e**
- **ou**

Pseudocódigo

- **Operadores Lógicos: e**

O **operador lógico** e, vai validar se o resultado de duas proposições são verdadeiras.

Caso, o resultado de **uma delas seja falso**, o resultado da **operação lógica** será **falsa**.

Pseudocódigo

- Operadores Lógicos: e

```
inteiro numero
leia(numero)
logico log = (numero <= 5) e (numero != 3)
```

Pseudocódigo

- **Operadores Lógicos: e**

Para facilitar, é sempre bom ter em mente esta tabela:

Proposição da Esquerda	Proposição da Direita	Resultado do E
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

Pseudocódigo

- **Operadores Lógicos: ou**

O **operador lógico** ou, vai validar se pelo menos o resultado de uma proposição é verdadeira.

Caso, o resultado de **uma delas seja verdadeiro**, o resultado da **operação lógica** será **verdadeira**.

Pseudocódigo

- Operadores Lógicos: ou

```
inteiro numero
leia(numero)
logico log = (numero <= 5) ou (numero != 3)
```

Pseudocódigo

- **Operadores Lógicos: ou**

Para facilitar, é sempre bom ter em mente esta tabela:

Proposição da Esquerda	Proposição da Direita	Resultado do OU
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

Pseudocódigo

- **Estruturas de Decisão**

Dentro da **programação**, nós temos as **estruturas de decisão**, que determinam um **bloco de código** que será ou não executado de acordo com decisões tomadas por **condições lógicas**.

Essas **condições lógicas**, são determinadas pelas **proposições** e pelos **operadores lógicos**.

Pseudocódigo

- **Estruturas de Decisão**

Temos 3 **estruturas de decisões**:

1. **se**
2. **senao**
3. **escolha-caso**

Pseudocódigo

- **Estruturas de Decisão: se**

A **estrutura de decisão** se, vai definir um **bloco de código** que será executado **se e somente se** uma **condição lógica for verdadeira**.

Para definir essa **estrutura de decisão** é necessário a **palavra reservada** se, seguida de dois **delimitadores**.

Pseudocódigo

- **Estruturas de Decisão: se**

1. **Delimitador de parênteses ( ), o qual é utilizado para definir a condição lógica que será validada**
2. **Delimitador de chaves { }, para definir o bloco de código da estrutura de condição**

Pseudocódigo

- **Estruturas de Decisão: se**

```
inteiro numero
leia(numero)
se (numero > 5) {
   escreva("O número é maior que 5")
}
```

Pseudocódigo

- **Estruturas de Decisão: senao**

A **estrutura de decisão** senao, é um complemento da **estrutura de condição** se, que vai definir um **bloco de código** que será executado se e aquela **condição lógica** do se for **falsa**.

Pseudocódigo

- **Estruturas de Decisão: senao**

```
inteiro numero
leia(numero)
se (numero > 5) {
    escreva("O número é maior que 5")
}
senao {
    escreva("O número é menor que 5")
}
```

Exercício

Exercício

gg.gg/LogicaSenac05

github.com/hmaltaurodev/Slides

Login: uept15-user@aluno.pr.senac.br

Senha: senac@1515

Fecomércio PR
Sesc Senac IFPD
Senac