

UNIP
UNIVERSIDADE PAULISTA

Lógica de Programação e Algoritmo

Estrutura Condicional – Parte II


Prof. Msc Wanderley Gonçalves Freitas
Wanderley.unip@gmail.com
www.professor.wanderley.nom.br

Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 1 de 18

UNIP
UNIVERSIDADE PAULISTA

Agenda

- Objetivo
- Conceito básicos : estrutura de condicional
- Tipos de estrutura de controle – Condicional
 - Condição Encadeada
 - `if(condicao){if (condicao){...}...}else {...}`
 - Condição Múltipla Escolha
 - `switch (expressão)`



Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 2 de 18

UNIP
UNIVERSIDADE PAULISTA

Objetivo

- Reconhecer a necessidade do uso de estruturas de condição encadeadas ou condição de múltipla escolha;
- Aprender a sintaxe da estrutura de condição múltipla escolha `switch (expressão)`
- Aprender a sintaxe da estrutura de condição encadeada `if(condicao){if (condicao){ ...}...}else {...}`
- Aprender a resolver problemas que requeiram o uso de estrutura de condição encadeada com a aplicação da técnica de interpretação de enunciados.

Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 3 de 18

UNIP
UNIVERSIDADE PAULISTA

Algoritmo & Programa

- Propriedades dos algoritmos
 - Possuir ações simples e bem definidas (não ambíguas);
 - Possuir seqüência ordenada;
 - Possuir seqüência finita de passos;
 - Possuir fim (todas as ações devem ser executadas em um tempo finito);
 - Pode receber entradas externas;
 - Pode fornecer saídas externas.

Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 4 de 18

UNIP
UNIVERSIDADE PAULISTA

Estrutura de Condicional – Encadeada

Seleção Encadeada

- Quando, devido à necessidade de processamento, agrupamos várias condições (se-entao-senao) usadas para o teste de múltiplos casos

```

algoritmo "CondicaoSimple4"
var
inicio
    se ( condicao 1) entao
        comando1
    senao
        se ( condicao 2) entao
            comando2
        senao
            se ( condicao 3) entao
                comando3
            senao
                comando4
        fimSe
    fimSe
fimSe
  
```

Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 5 de 18

UNIP
UNIVERSIDADE PAULISTA

Estrutura de Condicional- Encadeada Exercício (4)

Exercício (4). Faça um algoritmo que receba o **nome e as duas notas** obtidas por um aluno durante o semestre. Calcular a sua média (aritmética), informar o **nome e conceito final**, conforme tabela abaixo. (*observação: quatro condições*)

Conceito	Média
Conceito "A"	Media >= 9
Conceito "B"	Media >= 7
Conceito "C"	Media >= 6
Conceito "D"	Media < 6

Prof. Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 6 de 18

Estrutura de Condicional- Encadeada Exercício (4)

```

public class LPE02Exercicio04Problema {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double nota1;
        double nota2;
        double media;
        System.out.print("digite nota1");
        nota1 = teclado.nextDouble();
        System.out.print("digite nota2:");
        nota2 = teclado.nextDouble();

        media = (nota1 + nota2) / 2;

        if (media >= 9) {
            System.out.print(" conceito A \n");
        }
        if (media >= 7) {
            System.out.print(" conceito B\n");
        }
        if (media >= 6) {
            System.out.print(" conceito C\n");
        }
        if (media < 6) {
            System.out.print(" conceito D");
        }
    }
}
    
```

Problema?

O que será impresso se informar
 nota1 = 9
 nota2 = 9.5 ?

```

<terminated> LPE02Exercicio04Problema2 [jav
digite nota1
9
digite nota2:
9,5
conceito A
conceito B
conceito C
    
```

Por que foram impressas as 3 mensagens?

7 de 18

Estrutura de Condicional- Encadeada Exercício (4)

```

public class LPE02Exercicio04Problema1 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double nota1;
        double nota2;
        double media;
        System.out.print("digite nota1");
        nota1 = teclado.nextDouble();
        System.out.print("digite nota2:");
        nota2 = teclado.nextDouble();
        media = (nota1 + nota2) / 2;

        if (media >= 9) {
            System.out.print(" conceito A \n");
        }
        if (media >= 7) && (media < 9) ) {
            System.out.print(" conceito B\n");
        }
        if (media >= 6) && (media < 7) ) {
            System.out.print(" conceito C\n");
        }
        if (media < 6) {
            System.out.print(" conceito D");
        }
    }
}
    
```

Desvantagens?

O que será impresso se informar
 nota1 = 9
 nota2 = 9.5 ?

```

<terminated> LPE02Exercicio04Problema2 [jav
digite nota19,0
digite nota2:9,5
conceito A
    
```

Desvantagens nessa solução?

DESVANTAGEM – Teste Redundante quando para o(s) "primeiro(s) SE(s)" o resultado do teste de condição for verdadeiro.

8 de 18

Estrutura de Condicional- Encadeada - Exercício (4)

```

public class LPE02Exercicio04Problema2 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double nota1;
        double nota2;
        double media;
        System.out.print("digite nota1");
        nota1 = teclado.nextDouble();
        System.out.print("digite nota2:");
        nota2 = teclado.nextDouble();
        media = (nota1 + nota2) / 2;

        if (media >= 9) {
            System.out.print(" conceito A \n");
        } else if (media >= 7) {
            System.out.print(" conceito B\n");
        } else if (media >= 6) {
            System.out.print(" conceito C\n");
        } else {
            System.out.print(" conceito D");
        }
    }
}
    
```

Solução?

O que será impresso se informar
 nota1 = 9
 nota2 = 9.5 ?

```

<terminated> LPE02Exercicio04Problema2 [jav
digite nota19,0
digite nota2:9,5
conceito A
    
```

9 de 18

Estrutura de Condicional- switch

Seleção de Múltipla Escolha – switch (expressão)

O comando switch (cas0) é útil em situações onde há necessidade de se testar uma mesma variável (ou expressão) que possa conter diversos valores

- Cada valor exige a execução de comandos diferentes.

10 de 18

Estrutura de Controle - Seleção

A sintaxe do switch é:

```

switch (key) {
    case value:
        expressões;
        break;

    case value :
        expressões;
        break;

    default:
        expressões;
}
    
```

11 de 18

Estrutura de Condicional- Múltipla Escolha

Seleção de Múltipla Escolha – Escolha(Caso)

```

algoritmo "CondicaoSimples5"
var
inicio
    escolha <identificador>
    caso <valor1>
        primeiro bloco de instrução
    caso <valor2>
        primeiro bloco de instrução
    caso <valor3>
        primeiro bloco de instrução
    outrocaso
        outro bloco de instrução
    fimsecolha
fimalgoritmo
    
```

12 de 18

UNIP UNIVERSIDADE PAULISTA

Estrutura de Condicional- Multipla Escolha Exercício (5)

Exercício (4). Faça um algoritmo que receba o **numero inteiro de 1 a 7**. Informe o dia da semana correspondente, conforme tabela abaixo. Caso seja informado um **numero maior que 7**, exibe uma mensagem de número inválido. (*observação: oito condições*)

Numero	Dia da semana
1	Domingo
2	Segunda
3	Terça
4	Quarta
5	Quinta
6	Sexta
7	sabado

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 13 de 18

UNIP UNIVERSIDADE PAULISTA

Estrutura de Condicional- Multipla Escolha Exercício (5)

```

public class LPE02Exercicio05 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int dia;
        System.out.print("Informe dia: ");
        dia = teclado.nextInt();
        switch (dia) {
            case 1:
                System.out.print("domingo");
                break;
            case 2:
                System.out.print("segunda");
                break;
            case 3:
                System.out.print("terca");
                break;
            case 4:
                System.out.print("quarta");
                break;
            case 5:
                System.out.print("quinta");
                break;
            case 6:
                System.out.print("sexta");
                break;
            case 7:
                System.out.print("sabado");
                break;
            default:
                System.out.print("numero invalido");
                break;
        }
    }
}
    
```

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 14 de 18

UNIP UNIVERSIDADE PAULISTA

Exercício

LPE - Lista Exercício Parte II

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 15 de 18

UNIP UNIVERSIDADE PAULISTA

Referências Bibliográficas

ASCENCIO, Ana F. Gomes; CAMPOS, Edilene A. V. de. **Fundamentos de programação de computadores: algoritmos, Pascal e C/C++**. São Paulo: Prentice Hall, 2002.

CRUZ, Adriano J. O.; KNOPMAN, Jonas. O que são algoritmos? **Projeto de Desenvolvimento de Algoritmos: Núcleo de Computação Eletrônica**. Rio de Janeiro, 2001. Disponível em: <<http://equipe.nce.ufrj.br/adriano/algoritmos/apostila/indice.htm>>. Acesso em: 14 ago. 2013.

CRUZ, Adriano J. **Algoritmo "LPA02Exercicio04Problema" envolvimento de Algoritmos: Núcleo de Computação Eletrônica**. Rio de Janeiro, 2001. Disponível em: <<http://equipe.nce.ufrj.br/adriano/algoritmos/apostila/indice.htm>>. Acesso em: 14 ago. 2013.

MARTINS, Luiz G. A. **Introdução a algoritmos**. Uberlândia: UFU, 2009. Disponível em: <http://www.facom.ufu.br/~gustavo/IC/Programacao/Apostila_Algoritmos.pdf>. Acesso em: 14 ago. 2013.

SOUZA, Cláudio M. de. **VISUALG: editor e interpretador de algoritmos**. Disponível em: <<http://apoioinformatica.inf.br>>. Acesso em: 14 ago. 2013.

XAVIER, Gley Fabiano Cardoso. **Lógica de programação**. 9. ed. São Paulo: Senac São Paulo, 2005. (Nova Série Informática).

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 16 de 18

UNIP UNIVERSIDADE PAULISTA

Perguntas

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 17 de 18

UNIP UNIVERSIDADE PAULISTA

Obrigado

Wanderley

Wanderley.unip@gmail.com

www.wg.pro.br

Prof: Msc Wanderley Gonçalves Freitas LPE 02 – Estrutura Condicional 18 de 18