

Aula 02

Estruturas de controlo condicionais

Tipo de dados, operadores e instruções

Programação 1, 2015-2016

v1.0, 01-10-2015

, DETI, Universidade de Aveiro

02.1

Conteúdo

1	Estruturas de controlo condicionais (decisão)	1
2	Tipos de dados booleanos, operadores relacionais e operadores lógicos	1
3	Estrutura de decisão <code>if</code>	2
4	Estrutura de decisão múltipla <code>switch</code>	3

02.2

1 Estruturas de controlo condicionais (decisão)

Estruturas de controlo: decisão

- Um das particularidades de um computador é a capacidade de repetir tarefas ou executar tarefas consoante determinadas condições.
- Para implementar programas mais complexos, temos a necessidade de executar instruções de forma condicional.
- Determinadas instruções só podem/devem ser executadas depois da avaliação de determinadas condições.
- As instruções que permitem condicionar a execução de outras designam-se por *estruturas de controlo*. Nestes slides vamos apresentar as *estruturas de decisão*.
- Temos em Java (e em muitas outras linguagens) dois tipos de instruções de decisão: `if` e `switch`.

02.3

2 Tipos de dados booleanos, operadores relacionais e operadores lógicos

Alguns conceitos essenciais...

- Tipo de dados `boolean` – Podem assumir os valores `true` e `false` (verdadeiro e falso).
- Operadores relacionais: `<`, `<=`, `>`, `>=`, `==`, `!=`
- Operadores lógicos: `!`, `||`, `&&`
- Exemplos:

```

boolean cond1, cond2, cond3, cond4, cond5;

cond1 = 3 > 0;           // cond1 fica com true
cond2 = 5 != 5;          // cond2 fica com false
cond3 = cond1 || cond2;  // cond3 fica com true
cond4 = cond1 && cond2;   // cond4 fica com false
cond5 = !cond4;          // cond5 fica com true

```

3 Estrutura de decisão `if`

Instrução de decisão `if`

```

if (condição)
    instrução;

```

- a condição é avaliada;
- tem que ser uma expressão cujo resultado seja do tipo booleano;
- se verdadeira, é executada a instrução;
- se falsa, o programa continua na linha seguinte;
- exemplo:

```

int x;
System.out.print("Um valor inteiro:");
x = sc.nextInt();
if ( x < 0)
    x = -x;
System.out.println("O valor absoluto e' " + x);

```

Instrução de decisão `if` (2)

```

if (condição) {
    // bloco1
}
else {
    // bloco2
}

```

- a condição é avaliada;
- se verdadeira, é executado o bloco1;
- se falsa, é executado o bloco2.

Exemplo:

```

Scanner kb = new Scanner(System.in);
double nota;
System.out.print("Nota do aluno: ");
nota = kb.nextDouble();
if (nota >= 9.5) {
    System.out.println("Aluno aprovado");
}
else {
    System.out.println("Aluno reprovado");
}

```

Instrução de decisão `if` (3)

- A seguir à instrução decisória `if` ou ao separador `else`, podemos ter qualquer tipo de instrução, inclusive outras instruções de decisão.

```

if (condiçao1) {
    if (condição2) {
        // bloco1;
    }
    else {
        // bloco2;
    }
}
else {
    // bloco3;
}

```

```

if (condiçaoA) {
    // blocoA;
}
else if (condiçãoB) {
    // blocoB;
}
else {
    // blocoC;
}

```

✔ **Indentação:** Escrever programas com espaçamentos à esquerda (indentação) para facilitar a compreensão da estrutura de controlo de fluxo do programa.

02.7

Exemplo

```

if (sexo == 'F' || sexo == 'f') {
    if (imc <= 18.5)
        System.out.println("Excessivamente magra");
    else
        System.out.println("Normal");
}
else if (sexo == 'M' || sexo == 'm') {
    if (imc < 20)
        System.out.println("Excessivamente magro");
    else
        System.out.println("Jovem ok");
}
else {
    System.out.println("Sexo desconhecido!");
}

```

02.8

4 Estrutura de decisão múltipla switch

Instrução de decisão múltipla switch

Algumas situações de decisão encadeadas com a instrução `if` podem ser resolvidas através da instrução de decisão múltipla `switch`.

```

switch (expressão)
{
    case valor1:
        // bloco1;
        break;
    case valor2:
        // bloco2;
        break;
    default:
        // bloco3;
        break;
}

```

- A expressão deve ser do tipo enumerado (número inteiro ou carácter no caso dos tipos primitivos do Java – `byte`, `short`, `int` ou `char`).
- As constantes que constituem a lista de alternativas são do mesmo tipo da expressão.
- Primeiro é calculada a expressão e depois o seu valor é pesquisado na lista de alternativas existentes em cada case, pela ordem com que são especificados.
- Se a pesquisa for bem sucedida, o bloco de código correspondente é executado.
- Caso não exista na lista e se o default existir, o bloco de código correspondente é executado.
- A execução do `switch` só termina com o aparecimento da instrução `break`.

02.9

Exemplo com switch

```

Scanner kb = new Scanner(System.in);
int a, b, r = 0;

```

```

int op;
char opChar = ' ';
boolean ok = true;
System.out.print("Operand 1: ");
a = kb.nextInt();
System.out.print("Operand 2: ");
b = kb.nextInt();
System.out.print("Operation (1[+],2[-],3[*],4[/]: ");
op = kb.nextInt();
switch(op) {
    case 1:
        r = a + b; opChar = '+'; break;
    case 2:
        r = a - b; opChar = '-'; break;
    case 3:
        r = a * b; opChar = '*'; break;
    case 4:
        if (b != 0) { r = a / b; opChar = '/'; } else ok = false;
        break;
    default:
        ok = false;
        System.out.println("ERROR: Invalid operation!");
        break;
}
if (ok) {
    System.out.printf("%d %c %d = %d\n", a, opChar, b, r);
}

```

02.10

Instrução condicional e erros externos

- A detecção e tratamento de erros externos ao programa (ex: informação com origem no utilizador), pode ser sistematicamente resolvida com a instrução condicional:

```

Scanner kb = new Scanner(System.in);
int a, b, r = 0;
int op;
char opChar = ' ';
System.out.print("Operand 1: ");
a = kb.nextInt();
System.out.print("Operand 2: ");
b = kb.nextInt();
System.out.print("Operation (1[+],2[-],3[*],4[/]: ");
op = kb.nextInt();
if (op < 1 || op > 4 || (op == 4 && b == 0)) {
    System.out.println("ERROR: Invalid operation!");
    System.exit(1); // finish program!
}
switch(op) {
    case 1:
        r = a + b; opChar = '+'; break;
    case 2:
        r = a - b; opChar = '-'; break;
    case 3:
        r = a * b; opChar = '*'; break;
    case 4:
        r = a / b; opChar = '/'; break;
}
System.out.printf("%d %c %d = %d\n", a, opChar, b, r);

```

02.11