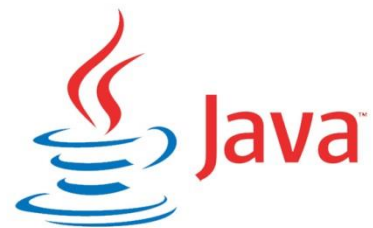


Lógica de Programação

Unidade 12 – Comando Switch



QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

INTRODUÇÃO	3
COMANDO SWITCH.....	3
SINTAXE DO COMANDO SWITCH	3
EXEMPLOS DE USO DO SWITCH	4

INTRODUÇÃO

Além da estrutura *if/else*, a linguagem Java conta com a estrutura *switch*, que apesar de ser mais limitada que o *if/else*, oferece uma sintaxe mais enxuta para testar possibilidades de igualdade.

COMANDO SWITCH

Serve para testar diferentes possibilidades de valores para uma variável (que pode ser um atributo ou um argumento). Funciona com dados do tipo *int*, *char*, *byte*, *String*¹. Só testa **igualdade**. Não é possível inserir testes lógicos no *switch* para verificar se é maior, menor, diferente, etc.

Sintaxe do comando Switch

```
switch(variável){ //seleciona a variável
    case 1: //caso o valor dela seja 1
        comandos; //executa os comandos
        break; //termina
    case 2: //caso o valor dela seja 2
        comandos; //executa os comandos
        break; //termina
    default: //caso não seja nenhuma das opções anteriores
        comandos; //executa esses comandos
}
```

OBS: na estrutura *switch*, o uso do *default* não é obrigatório.

A estrutura **switch** é delimitada por um conjunto de chaves. Cada **case** é considerado um **rótulo de teste**. Ao lado da palavra **case**, inserimos o possível valor que estamos testando, seguido de dois pontos.

O comando **break**, que vemos ao final de cada **case**, funciona como um comando de interrupção. Ele impede que os próximos rótulos sejam executados a partir da correspondência verdadeira.

O rótulo **default** é executado sempre que a estrutura não encontrar uma instrução de parada (**break**).

¹ A partir da versão 7 do Java.

Exemplos de uso do Switch


Para demonstrar como podemos utilizar o comando *switch*, utilizaremos como base o diagrama da classe Data a seguir.

Data
-dia:int -mes:int -ano:int
+gets e sets +obterMesPorExtenso():String +obterTotalDeDiasDoMes():int

O exemplo a seguir, a classe Data, tem como atributos o dia, o mês e ano. O método **obterMesPorExtenso** verifica o número do mês, e com base no número retorna o nome correspondente. O método **obterTotalDeDiasDoMes** verifica o número do mês e retorna a quantidade de dias que ele tem, demonstrando que podemos **agrupar rótulos** de *cases* que terão o mesmo resultado.

Vamos observar o código exemplo nas páginas a seguir.

Código da classe Data - Parte 1



```

1 public class Data{
2     private int dia;
3     private int mes;
4     private int ano;
5
6     public void setDia(int dia){
7         this.dia = dia;
8     }
9     public void setMes(int mes){
10        this.mes = mes;
11    }
12    public void setAno(int ano){
13        this.ano = ano;
14    }
15
16    public int getDia(){
17        return this.dia;
18    }
19    public int getMes(){
20        return this.mes;
21    }
22    public int getAno(){
23        return this.ano;
24    }

```

Código da classe Data - Parte 2

```

26 public String obterMesPorExtenso() {
27     String nomeDoMes;
28     switch (this.mes) {
29         case 1:
30             nomeDoMes = "Janeiro";
31             break;
32         case 2:
33             nomeDoMes = "Fevereiro";
34             break;
35         case 3:
36             nomeDoMes = "Março";
37             break;
38         case 4:
39             nomeDoMes = "Abril";
40             break;
41         case 5:
42             nomeDoMes = "Maio";
43             break;
44         case 6:
45             nomeDoMes = "Junho";
46             break;
47         case 7:
48             nomeDoMes = "Julho";
49             break;
50         case 8:
51             nomeDoMes = "Agosto";
52             break;
53         case 9:
54             nomeDoMes = "Setembro";
55             break;
56         case 10:
57             nomeDoMes = "Outubro";
58             break;
59         case 11:
60             nomeDoMes = "Novembro";
61             break;
62         case 12:
63             nomeDoMes = "Dezembro";
64             break;
65         default:
66             nomeDoMes = "Mês inválido";
67     }
68     return nomeDoMes;
69 }

```

```

public int obterTotalDeDiasDoMes() {
    int totalDeDias;
    switch (this.mes) {
        case 1: case 3: case 5:
        case 7: case 8: case 10:
        case 12:
            totalDeDias = 31;
            break;
        case 4: case 6:
        case 9: case 11:
            totalDeDias = 30;
            break;
        case 2:
            totalDeDias = 28;
            break;
        default:
            totalDeDias = 0;
    }
    return totalDeDias;
}

```

Vamos ver como utilizar estes métodos na principal.

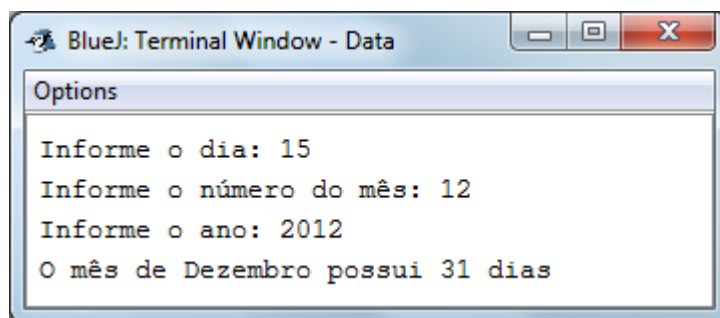
Primeiramente, criamos o objeto **data**, solicitamos ao usuário o dia, o mês e o ano e em seguida, após executar os métodos, exibimos a mensagem com as respostas.

```

1 import java.util.Scanner;
2 public class Main{
3     public static void main(String args[]){
4         Scanner ler = new Scanner(System.in);
5         Data d1 = new Data();
6
7         System.out.print("Informe o dia: ");
8         d1.setDia(ler.nextInt());
9         System.out.print("Informe o número do mês: ");
10        d1.setMes(ler.nextInt());
11        System.out.print("Informe o ano: ");
12        d1.setAno(ler.nextInt());
13
14        String nomeDoMes = d1.obterMesPorExtenso();
15        int quantidadeDeDias = d1.obterTotalDeDiasDoMes();
16
17        System.out.print("O mês de " + nomeDoMes + " possui " + quantidadeDeDias + " dias");
18    }
19 }

```

Testando no terminal:



```

BlueJ: Terminal Window - Data
Options
Informe o dia: 15
Informe o número do mês: 12
Informe o ano: 2012
O mês de Dezembro possui 31 dias

```