Lógica de **P**rogramação

Unidade 9 – Estruturas condicionais - *if*

QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

INTRODUÇÃO

Na nossa vida cotidiana verificamos condições antes de executarmos alguma tarefa. Exemplo: se eu tiver dinheiro, vou comprar aquele tênis. Se a loja me der desconto, pagarei à vista, senão pagarei no cartão. E assim por diante. Nos sistemas de computador não é diferente. Por vezes eles precisam analisar o que irão executar, de acordo com as informações recebidas do usuário.

Até o momento os algoritmos que criávamos passavam por todas as etapas impostas pelo *Main*, todos os métodos invocados executavam tudo o que estivesse dento deles. A partir de agora vamos fazer com que nossos programas tenham condições, analisem os dados para verificar qual resposta será exibida, qual cálculo será realizado, qual mensagem aparecerá na tela.

COMANDOS CONDICIONAIS

As linguagens de programação normalmente possuem duas estruturas que permitem a análise condicional de dados, o **if**(se) e o **switch**. Vamos nos concentrar no primeiro.

A estrutura **if** é utilizada para que o sistema teste possibilidades, e de acordo com cada possibilidade executa um ou mais comandos. Por esta razão muitas vezes é chamado de **desvio condicional**, ou **estrutura de decisão**.

TESTES LÓGICOS

Um teste lógico é uma comparação entre dois valores compatíveis. Cada teste lógico precisa ter **dois valores** a serem comparados e **um operador relacional**. Os valores podem ser variáveis e/ou constantes.

Um teste lógico tem como possível resultado apenas: ***true*** ou ***false***.

- ***True***: se a comparação for verdadeira
- ***False***: se a comparação for falsa

Operadores Relacionais

Para que possamos montar testes lógicos para o processador, devemos sempre utilizar um dos operadores abaixo demonstrados.

Operador	Significado
>	Maior do que
<	Menor do que
>=	Maior ou igual a
<=	Menor ou igual a
==	Igual a
!=	Diferente de

Exemplos de testes lógicos

a>b	Verifica se o valor de "a" é maior do que o valor de "b"
a!=b	Verifica se o valor de "a" é diferente do valor de "b"
a+b> c	Verifica se o resultado da soma de "a" e "b" é maior que o valor de "c"

INSTRUÇÃO IF

A estrutura **if** é utilizada para gerenciar as condições de um programa; assim, podemos programar para que o computador execute um bloco de comandos caso a condição seja verdadeira, ou um outro bloco de comandos caso seja falsa.

Sintaxe da estrutura if-else

```
if(condição){
    <comando>;
}else{
    <comando>;
}
```

Vale lembrar que o uso do else nem sempre é obrigatório; podemos ter apenas um if com seu comando, e caso a condição seja falsa ele não executa nada. Porém se estivermos utilizando em um método que requer retorno temos que colocar o else retornando uma resposta caso nenhuma das anteriores forem executadas.

Vamos considerar o exemplo da classe Numero, tendo como atributo um valor, e um método que diga se este número é positivo ou negativo.

Numero
-valor: int
+getValor():int +setValor(valor:int):void +verificarValor():String

O método verificarValor deve nos retornar apenas uma resposta. Logo, para qualquer valor, temos duas possibilidades: ou ele é positivo (vamos considerar o zero

positivo) ou ele é negativo. Vamos observar as condições que determinam se um número é positivo ou negativo:

Possibilidade	Condição	Teste lógico
Positivo	Valor precisa ser maior ou igual a 0	this.valor>=0
Negativo	Valor precisa ser menor que 0	this.valor<0

Como temos apenas duas possibilidades, se o número não for positivo, logo ele é negativo. Portanto, podemos utilizar apenas a primeira condição, se ela for verdadeira, teremos um número positivo, se ela for falsa, teremos um número negativo.

```
Public String verificarValor(){
      if(this.valor>=0){
            return "Positivo";
      }else{
            return "Negativo";
      }
}
```

If com Desvio Condicional Encadeado

É muito comum que dentro de um algoritmo tenhamos várias possibilidades de resultado. Para este tipo de situação podemos utilizar o encadeamento (aninhamento) de instruções **if – elseif.**..

Sintaxe

```
if(condição1){
    <comandos>;
}else if(condição2){
    <comandos>;
}else{
    <comandos>;
}
```

Vamos pensar no mesmo algoritmo citado anteriormente, porém iremos considerar o zero como nulo. Portanto, o número pode ser Positivo, Negativo ou Nulo. Note que temos três possibilidades de resposta:

Possibilidade	Condição	Teste lógico
Nulo	Valor precisa ser igual a 0	this.valor == 0
Positivo	Valor precisa ser maior que0	this.valor> 0
Negativo	Valor precisa ser menor que 0	this.valor< 0

Como temos três possibilidades, testamos a primeira. Se o valor por igual a 0, então ele é nulo; porém, se não for nulo, ele pode ser maior que zero, então ele será positivo; mas se ele não for nem nulo e nem positivo, então ele só pode ser negativo.

```
public String verificarValor(){
      if(this.valor ==0){
            return "Nulo";
      }else if(this.valor>0){
            return "Positivo";
      }else{
            return "Negativo";
      }
}
```

Exemplo completo

Vamos utilizar como exemplo um programa que após o usuário digitar um número, mostre que dia da semana este número equivale.

Diagrama

Semana
-dia:byte
+getDia():byte +setDia(dia:byte):void +verDiaDaSemana():String

Para o método **verDiaDaSemana**() retornar o nome de acordo com o número, temos que testar todas as possibilidades:

Possibilidade	**Condição**	**Teste lógico**
Domingo	O dia precisa ser igual a 1	this.dia == 1
Segunda-feira	O dia precisa ser igual a 2	this.dia == 2
Terça-feira	O dia precisa ser igual a 3	this.dia == 3
Quarta-feira	O dia precisa ser igual a 4	this.dia == 4
Quinta-feira	O dia precisa ser igual a 5	this.dia == 5
Sexta-feira	O dia precisa ser igual a 6	this.dia == 6
Sábado	O dia precisa ser igual a 7	this.dia == 7
Sem correspondência	O dia não corresponde a nenhum anterior	-

Codificação da classe

```java
public class Semana{
    private int dia;

    public void setDia(int dia){
        this.dia = dia;
    }
    public int getDia(){
        return this.dia;
    }
    public String verDiaDaSemana(){
        if(this.dia==1){
            return "Domingo";
        }else if(this.dia==2){
            return "Segunda-feira";
        }else if(this.dia==3){
            return "Terça-feira";
        }else if(this.dia==4){
            return "Quarta-feira";
        }else if(this.dia==5){
            return "Quinta-feira";
        }else if(this.dia==6){
            return "Sexta-feira";
        }else if(this.dia==7){
            return "Sábado";
        }else{
            return "Sem correspondência";
        }
    }
}
```

Classe Main

```java
import java.util.Scanner;
public class Main{
    public static void main(String args[]){
        Scanner ler = new Scanner(System.in);
        Semana s1 = new Semana();
        System.out.println("Digite um número: ");
        s1.setDia(ler.nextInt());

        System.out.println("RESPOSTA:");
        String dia = s1.verDiaDaSemana(); //atribuindo a resposta à uma variável.
        System.out.println(dia);
    }
}
```

Resultado no terminal

```
Digite um número:
6
RESPOSTA:
Sexta-feira
```