Programação Estruturada

Aula 04 - Comandos de seleção

Apresentação

Na aula passada, você conheceu em detalhes as funções da linguagem Java para realização da entrada e saída de dados via console. Essas funções são essenciais para a resolução de problemas, pois na prática há, na maioria das vezes, uma grande interação entre os usuários e os programas de computador. Lembra que falamos sobre isso? Entretanto, uma vez com os dados do usuário, os programas de computador normalmente têm que processar os dados e tomar decisões. Por exemplo, dado as notas de um aluno ao longo do ano, o computador deve ser capaz de calcular a média e decidir se ele foi aprovado por média ou se ele vai para a recuperação. Assim, diferentemente do que acontece com os seres humanos, as decisões do computador são todas programadas.

Nesta aula, você vai aprender a fazer uso dos comandos de seleção da linguagem Java, comandos esses fundamentais para se especificar as decisões que um computador irá realizar. Entre os comandos de seleção, temos os comandos ***if/else* (se/senão) e *switch* (escolha)**. Como já foi dito, esse conteúdo é de grande importância para o desenvolvimento de programas Java, pois é através desses comandos que são feitos inúmeros testes condicionais ao decorrer dos programas.

Mais uma vez, vale salientar, para que esse assunto seja bem absorvido, é necessário que você tenha trabalhado bem com as atividades práticas dos conteúdos anteriores. Tal como nas aulas passadas, serão sugeridas atividades durante o desenvolvimento do texto e ao final da aula. Tudo isso para facilitar o seu aprendizado.

Vídeo 01 - Apresentação

Objetivos

Nesta aula, você será capaz de:

- Descrever o funcionamento dos comandos de seleção da linguagem Java.
- Saber aplicar adequadamente os comandos de seleção no desenvolvimento de programas Java.

1. Comandos de seleção

Os comandos de um programa seguem uma sequência linear de execução, ou seja, cada instrução (linha do programa) é executada uma após a outra. Quando queremos ter sequências de execução diferentes de acordo com os dados de entrada, fazemos uso dos comandos de seleção.

A linguagem Java suporta dois tipos principais de comandos de seleção: **if** e **switch**.

1.1 Comando de seleção IF

O comando **if** é o mais básico de todos os comandos de seleção. Ele determina que o programa deve executar um certo comando ou seção de código apenas se um determinada condição for verdadeira. A forma geral do comando **if** é:

if (condição) comando;

Uma condição é verdadeira se o seu valor for diferente de zero. Usualmente, o valor de uma expressão verdadeira é igual a 1.

Considere o seguinte trecho de código a seguir que calcula a média de um aluno, a partir das suas notas, e observe as ilustrações do problema para facilitar o entendimento:

```
import java.util.Scanner;

public class IfDemo {
    public static void main(String[] args) {
        float nota1, nota2, nota3, nota4, media;
        Scanner leitor = new Scanner(System.in);
        System.out.println("Digite as quatro notas que você tirou:");
        nota1 = leitor.nextFloat();
        nota2 = leitor.nextFloat();
        nota3 = leitor.nextFloat();
        nota4 = leitor.nextFloat();
        media = (nota1 + nota2 + nota3 + nota4)/4;
        if(media>=7) {
            System.out.println("O aluno foi aprovado!");
        }
        System.out.println("FIM!");
    }
}
```

O programa mostrado solicita as notas do aluno e utiliza o comando de seleção if para imprimir a mensagem “O aluno foi aprovado!” somente quando a média do aluno for maior ou igual a 7. Isso porque a expressão media >=7 será verdadeira (**true**), quando o valor da variável média for igual a superior a 7, fazendo com que o comando System.out.println("O aluno foi aprovado!\n") seja executado.

No caso da expressão ser falsa, ela resultará no valor **false** e o comando que imprime a mensagem “O aluno foi aprovado!” não será executado.

No exemplo mostrado, a instrução **if** determinará a execução de apenas um único comando, quando a condição for verdadeira (media >=7), que imprime a mensagem de que o aluno foi aprovado. Caso desejássemos que fossem executados vários comandos como, por exemplo, que fossem exibidas várias mensagens, teríamos que inserir a sequência de comandos dentro das chaves “{“ e “}”, determinando o bloco do comando if.

Veja como fica isso no exemplo a seguir:

```java
import java.util.Scanner;

public class IfDemo {
    public static void main(String[] args) {
        float nota1, nota2, nota3, nota4, media;
        Scanner leitor = new Scanner(System.in);
        System.out.println( "Digite as quatro notas que você tirou:");
        nota1 = leitor.nextFloat();
        nota2 = leitor.nextFloat();
        nota3 = leitor.nextFloat();
        nota4 = leitor.nextFloat();
        media = (nota1 + nota2 + nota3 + nota4)/4;
        if(media>=7) {
            System.out.println("O aluno foi aprovado!");
            System.out.println("Parabéns!");
        }
        System.out.println("FIM!");
    }
}
```

Na verdade, quando a instrução if determina a execução de um único comando, é opcional o uso das chaves. Porém, é recomendado que sempre sejam utilizadas para facilitar a clareza e o entendimento do código.

Vídeo 02 - Comando de Seleção If

É recomendado que sempre se utilize os símbolos “{“ e “}” para delimitar a atuação da instrução if, mesmo que o bloco delimitado seja de apenas um comando. Essa boa prática irá facilitar a clareza e o entendimento do código por outras pessoas!

Atividade 01

1. Crie um programa em Java que receba 5 números e imprima a mensagem “Múltiplo de 2” caso a soma dos números digitados seja múltiplo de 2. Dica: para saber se um número é multiplo de 2, basta verificar se o resto da divisão (operador %) do número por 2 é igual a zero.

2. Crie um programa para receber o comprimento de 3 pedaços de madeira e mostrar uma mensagem caso eles formem um triângulo. Dica: monte uma expressão usando operadores lógicos e que seja verdadeira somente quando o comprimento de cada pedaço de madeira é menor que a soma do comprimento dos 2 pedaços restantes (A < B + C e B < A + C e C < A + B).

Vamos agora analisar outra situação. Considere que desejamos não somente imprimir a mensagem “Aluno aprovado!” quando a média for igual ou superior a 7, mas também exibir a mensagem “Aluno não foi aprovado” no caso da sua média ser menor do que 7. O que será preciso alterar, no exemplo mostrado anteriormente, para que isso passe a acontecer? Uma alternativa é executar dois testes sequencialmente:

```
1 if(media>=7) {
2    System.out.println("O aluno foi aprovado!");
3 }
4
5 if(media < 7) {
6    System.out.println("O aluno não foi aprovado!");
7 }
```

Apesar desse código estar correto, ele se utiliza de duas expressões complementadoras (quando a primeira é verdadeira, a segunda é falsa e vice-versa). Nesse caso, é indicado que você faça uso da instrução if-else, cuja forma geral é:

```
1 if (condiçao) {
2    comando1;
3 } else {
4    comando2;
5 }
```

Assim sendo, o código anterior poderia ser reescrito da seguinte forma:

```
if(media>=7) {
    System.out.println("O aluno foi aprovado!");
} else {
    System.out.println("O aluno não foi aprovado!");
}
```

O comportamento desse novo código é igual ao anterior, porém agora não existem mais duas condições sendo avaliadas, apenas uma. Nesse caso, se a média for maior ou igual a 7, a expressão é verdadeira e apenas o primeiro comando println é executado. Caso contrário (média menor que 7), a expressão é falsa e apenas o segundo comando println é executado.

Atividade 02

1. Crie um programa para receber um número e indicar se ele é par ou ímpar.
2. Crie um programa que receba o nome e a idade de duas pessoas que nasceram em anos diferentes e que diga o nome da pessoa mais velha.

Comandos if-else encadeados

Você já viu nesta aula que podemos utilizar a instrução *if-else* para situações que envolvem duas expressões complementares — ou seja, condições equivalente e contrária. Entretanto, existem casos em que as expressões não são complementares. Considere o programa abaixo, o qual deve determinar se o número indicado é igual, menor ou maior do que 15. Veja um possível código para implementar esse comportamento:

```java
import java.util.Scanner;

public class IfElseDemo {
    public static void main(String[] args) {
        int numero;
        Scanner leitor = new Scanner(System.in);
        System.out.println ("Digite um numero: ");
        numero = leitor.nextInt();
        if (numero>15) {
            System.out.println("O número é maior que 15");
        }
        if (numero==15) {
            System.out.println("O número é igual a 15.");
        }
        if (numero<15) {
            System.out.println("O número é menor que 15");
        }
    }
}
```

Nesse caso, apesar das expressões serem complementares (se uma for verdadeira, as outras não são), temos mais de duas opções, o que nos impede de usarmos um if-else simples, conforme visto anteriormente. O que fazemos nesse caso é um sequenciamento de if-elses, como mostrado a seguir:

```java
import java.util.Scanner;

public class IfElseDemo {
    public static void main(String[] args) {
        int numero;
        Scanner leitor = new Scanner(System.in);
        System.out.println ("Digite um numero: ");
        numero = leitor.nextInt();
        if (numero>15) {
            System.out.println("O número é maior que 15");
        } else if(numero==15) {
            System.out.println("O número é igual a 15.");
        } else if(numero<15) {
            System.out.println("O número é menor que 15.");
        }
    }
}
```

Vídeo 03 - Comando de Seleção if-else

Atividade 03

1. Crie um programa para receber dois números e verificar se eles são iguais ou se um é maior que o outro. Imprima uma mensagem indicando se os números são iguais ou, no caso deles serem diferentes, imprima o maior valor digitado.
2. Crie um programa para receber uma nota e imprimir uma mensagem de acordo com a seguinte tabela:
    - De 0 até 3 – Você precisa melhorar muito!
    - Maior que 3 e menor que 7 – Você está quase conseguindo!
    - Maior ou igual a 7 e menor que 9 – Você conseguiu!
    - Maior ou igual a 9 – Você conseguiu com distinção!

3. Comando switch

Na seção anterior, apresentamos o comando **if-else** encadeados, o qual pode ser utilizado para resolver problemas de seleção. Porém, em alguns casos, o uso de comandos if-else encadeados torna a escrita e compreensão do código mais trabalhosa. Por exemplo, considere um programa que deve receber uma letra do usuário e imprimir uma palavra que comece com a letra digitada. O código a seguir mostra a rotina main considerando, por questões de espaço, apenas as letras de A a E:

```
import java.util.Scanner;

public class ProgramaLetras {
    public static void main(String[] args) {
        char letra;
        Scanner leitor = new Scanner(System.in);
        System.out.println("Digite uma letra maiúscula:");
        letra = leitor.nextLine().charAt(0);
        if (letra == 'A') {
            System.out.println("Arara");
        } else if (letra == 'B') {
            System.out.println("Bola");
        } else if (letra == 'C') {
            System.out.println("Casa");
        } else if (letra == 'D') {
            System.out.println("Dados");
        } else if (letra == 'E') {
            System.out.println("Escada");
        } else {
            System.out.println("Letra ainda sem palavra cadastrada...");
        }
    }
}
```

Observe que a leitura do caractere é feita através do leitor.nextLine() — leitura de uma String — e que o primeiro caractere dessa String é retornada, usando-se a função charAt(0), a qual será melhor vista em outra aula, mais adiante neste curso.

Pois bem, o comando **switch** facilita a escrita de trechos de programa em que a seleção deve ser feita entre várias alternativas. O **switch** é chamado de comando interno de seleção múltipla, ele testa o valor de uma expressão contra uma lista de constantes numéricas ou de caracteres. Veja o mesmo código fazendo uso do comando **switch**:

```java
import java.util.Scanner;

public class ProgramaLetras2 {
   public static void main(String[] args) {
      char letra;
      Scanner leitor = new Scanner(System.in);
      System.out.println("Digite uma letra maiúscula:");
      letra = leitor.nextLine().charAt(0);
      switch (letra) {
         case 'A':
            System.out.println("Arara");
            break;
         case 'B':
            System.out.println("Bola");
            break;
         case 'C':
            System.out.println("Casa");
            break;
         case 'D':
            System.out.println("Dados");
            break;
         case 'E':
            System.out.println("Escada");
            break;
         default:
            System.out.println("Letra ainda sem palavra cadastrada...");
      }
   }
}
```

A forma geral do comando **switch** é a seguinte:

```
switch ( expressao )
{
   case constante1 :
      sequencia_de_comandos ;
      break;
   case constante2 :
      sequencia_de_comandos ;
      break;
   case constante3 :
      sequencia_de_comandos ;
      break;
   ...
   default:
      sequencia_de_comandos;
}
```

O comando **switch** funciona da seguinte maneira: o valor da expressão é *testado*, em ordem, contra os valores das constantes especificadas nos comandos **case**. Quando ocorrer uma condição em que a expressão seja aceita, a sequência de comandos associado ao **case** em questão será executado até chegar ao comando **break** (que para a execução da case em questão e salta para a próxima linha de código) ou ao fim do comando **switch**. O comando **default** será apenas executado caso nenhum valor seja aceito. Esse comando é opcional. Se ele não existir, nenhuma ação será realizada caso todos os testes falhem.

Vídeo 04 - Comando Switch

Veja a seguir algumas observações sobre o comando switch.

- Os comandos switch e if diferem pois o switch testa igualdade. Já o if testa uma expressão lógica ou relacional.
- Duas constantes case não podem ter o mesmo valor dentro de um mesmo switch.

Atividade 04

1. Analise e descreva o comportamento do seguinte programa:

```
import java.util.Scanner;
public class ProgramaExemplo {
    public static void main(String[] args) {
        int numero;
        Scanner leitor = new Scanner(System.in);
        System.out.println("Digite um número:");
        numero = leitor.nextInt();
        switch (numero) {
            case 9:
                System.out.println("O número é igual a 9.");
                break;
            case 10:
                System.out.println("O número é igual a 10.");
                break;
            case 11:
                System.out.println("O número é igual a 11.");
                break;
            default:
                System.out.println("O número não é nem 9, nem 10, nem 11.");
        }
    }
}
```

2. Remova os comandos break do código do exercício anterior e observe o que muda em seu comportamento.

3. Crie um programa para receber uma letra do usuário e imprimir o nome de um país que se inicie com essa letra. Caso não exista um país com nome iniciado pela letra digitada, informar isso ao usuário.

Conclusão

Pois bem, você agora já sabe utilizar o comando switch, que é um comando de seleção múltipla, onde várias cláusulas são testadas sucessivamente. Um bom exemplo de aplicação do comando switch é na criação de programas que necessitem entradas do usuário para seleção da operação a ser realizada (menus de operações disponíveis no sistema).

4. Resumo

Nesta aula, você foi apresentado aos comandos de seleção if e switch. Aprendeu que o comando if testa uma determinada condição lógica, ou seja, uma expressão que pode assumir valor verdadeiro ou falso, lembrando que em Java qualquer valor diferente de zero é interpretado como verdadeiro e zero como falso. Você observou que o comando if também pode vir associado ao else no caso de duas situações complementares (quando uma for verdadeira, a outra é falsa e vice-versa). O uso do else reduz o tamanho e facilita o entendimento do código, além de aumentar sua eficiência computacional, pois reduz a quantidade de testes a serem realizados. Além disso, comandos if-else podem ser encadeados para melhorar a legibilidade e desempenho do código.

5. Autoavaliação

1. Utilizando o comando if simples, crie um programa que exiba um menu com as seguinte opções: A - multiplicar" e "B - somar", o programa deve ler a opção desejada, ler dois valores, executar a operação e exibir o resultado.

2. Aproveitando o código anterior, complemente-o utilizando if-else encadeados, inclua no programa as opções "C - Subtrair" e "D - Dividir".

3. Crie um programa que leia um número. Se o número for positivo, imprimir o dobro, senão o quadrado dele.

6. Referências

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores**: algoritmos, Pascal, C/C++ e Java. São Paulo: Editora Pearson, 2008.

THE JAVA tutorials. Disponível em: <http://download.oracle.com/javase/tutorial/>. Acesso em: 6 dez. 2011.