

CURSO: Ciência da Computação
PERÍODO: 4º.
DISCIPLINA: Técnicas Alternativas de Programação

DATA: ____/____/2013
PROFESSOR: Andrey
AULA: 03

APRESENTAÇÃO:

Hoje vamos conhecer o comando switch, o bloco try catch; como funcionam os arrays unidimensionais e multidimensionais em java e as Strings e seus métodos.

DESENVOLVIMENTO:

1. Comando switch

O comando switch é um comando usado quando é necessário fazer uma seleção entre várias opções a partir do valor de uma variável.

```
switch (<expressão>)  
{  
    case 1:<instruções>; break;  
    case 2:<instruções>; break;  
    case 3:<instruções>; break;  
    case 4:<instruções>; break;  
    case 5:<instruções>; break;  
    default::<instruções>;  
}
```

Exemplo. O programa abaixo lê um número do teclado e imprime o número por extenso.

```
import java.io.*;  
  
public class Escolhe  
{  
    public static void main(String args[ ]) throws IOException  
    {  
        int a;  
        BufferedReader teclado;  
        teclado = new BufferedReader(new InputStreamReader(System.in));  
        System.out.println('Digite o numero');  
        a = Integer.parseInt(teclado.readLine());  
        switch (a)  
        {  
            case 0: System.out.println("zero"); break;  
            case 1: System.out.println("um"); break;  
            case 2: System.out.println("dois"); break;  
            case 3: System.out.println("tres"); break;  
            case 4: System.out.println("quatro"); break;  
            case 5: System.out.println("cinco"); break;  
            case 6: System.out.println("seis"); break;  
            case 7: System.out.println("sete"); break;  
            case 8: System.out.println("oito"); break;  
            case 9: System.out.println("nove"); break;  
            default : System.out.println("Numero desconhecido");  
        }  
    }  
}
```

2. Bloco try catch

Quando uma exceção ocorre em um programa java o programa é interrompido imediatamente. Este fato nem sempre é bem vindo e deve ser evitado ao máximo.

O bloco try catch é usado para capturar as exceções que podem ocorrer em um programa, em vez de usar o throws, que apenas repassa esta exceção.

A exceção que ocorrer entre o try e o primeiro catch será tratada com a mensagem adequada e após este tratamento, serão executadas as instruções do bloco finally.

```
class Exemplo0303
{
    public static void main (String args[])
    {
        int num1,num2;
        try
        {
            BufferedReader teclado;
            teclado = new BufferedReader(new InputStreamReader(System.in));
            System.out.println('Digite o numero');
            num1 = Integer.parseInt(teclado.readLine());
            System.out.println('Digite o numero');
            num2 = Integer.parseInt(teclado.readLine());
            System.out.println("Soma = " + (num1+num2));
            System.out.println("Subtracao = " + (num1-num2));
            System.out.println("Multiplicacao = " + (num1*num2));
            System.out.println("Divisao = " + (num1/num2));
        }
        catch (ArithmeticException aex)
        {
            System.out.println("Erro de divisao por zero!" + aex);
        }
        catch (IOException ioex)
        {
            System.out.println("Numero de argumentos invalidos!" + ioex);
        }
        catch (NumberFormatException nfex)
        {
            System.out.println("Digite apenas numeros inteiros!" + nfex);
        }
        finally
        {
            System.out.println("Fim!");
        }
    }
}
```

3. Arrays

Em determinadas partes de programas torna-se necessário manipular diversas variáveis de um mesmo tipo de dado. Normalmente muitas destas variáveis estão relacionadas entre si, por exemplo os nomes dos 30 alunos de uma turma. Então, em vez de declararmos 30 variáveis vamos usar apenas uma, um array de nomes.

Um array é uma coleção de variáveis de um mesmo tipo ou classe. Os arrays em Java podem ser unidimensionais ou bidimensionais. Nos arrays, as variáveis são colocadas em posições contíguas. E cada posição do array possui um índice, que em java começa com zero.

Exemplo: array de inteiros chamado numeros

	78	9	17	5	20
Índice	0	1	2	3	4

No array acima o elemento da posição 3, que é denotado por `numeros[3]` tem o valor 5. O tamanho do array acima é igual ao número de elementos, ou seja 5. Pode-se acessar cada elemento do array através do seu índice, por exemplo:

`A <- numeros[3]` A recebe o elemento do array na posição 3
`numeros[0] <- 5` o elemento do array na posição 0 recebe 5, o que altera o elemento

Em Java:

Para declarar um array podemos fazer desta forma:

`int notas[] = new int[10];` cria um array de 10 posições de inteiros

`double valores[] = new double[15];` cria um array de 10 posições de double

`String nomes[] = new String[10];` cria um array de 10 posições de Strings

ou atribuindo diretamente os valores :

`int notas[] = { 3, 5, 75, 2, 0}`

cria um array de 5 posições de inteiros já com os valores para cada posição.

Exemplo: Um programa em java que lê as 4 notas de um aluno, calcula e imprime a média, e imprime as notas.

```
import java.io.*;

public class Vetores
{
    public static void main(String args [ ]) throws IOException
    {
        double notas[ ] = new double[4];
        double soma = 0.0;
        double media;
        int i;
        BufferedReader teclado = new BufferedReader(new InputSteramReader(System.in));
        for(i=0 ;i<4; i++)
        {
            notas[i] = new Double(teclado.readLine()).doubleValue();
            soma = soma + notas[i];
        }
        media = soma / 4;
        System.out.println("a media é "+media);
        for(i=0 ;i<4; i++)
        {
            System.out.println("a nota "+(i+1)+" é "+notas[i]);
        }
    }
}
```

3.1 Exemplos com arrays.

Para somar os elementos de um array de números é necessário percorrer o array e ir adicionando cada elemento do array a uma variável soma.

```
public class SomaArray {
    public static void main( String args[] )
    {
        int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        int soma = 0;

        for ( int i = 0; i < a.length; i++ )
            soma += a[ i ];

        System.out.println("Soma dos elementos: " + soma);
    }
}
```

O atributo **a.length** nos informa o número de elementos do array.

3.2 Pesquisando em um Array

Para pesquisar se um determinado elemento está em um array ou não é necessário comparar cada elemento do array com o elemento que se está procurando.

```
import java.io.*;
public class Busca {
    public static void main(String args[])throws IOException
    {
        int chave;
        int a[] = {1,2,3,5,6,8,9,10,20,34,23,12,32,18,16,14};
        String resultado;
        BufferedReader teclado = new BufferedReader(
            new InputStreamReader(System.in));
        System.out.println("Digite a chave de busca");
        chave = Integer.parseInt(teclado.readLine());
        resultado = "nao achou "+chave;

        for ( int n = 0; n < a.length; n++ )
            if ( a[n] == chave )
                resultado = "achou "+chave;
        System.out.println(resultado);
    }
}
```

3.3 Arrays Multi dimensionais

São a representação em Java de matrizes. Eles possuem uma dimensão que corresponde às linhas da matriz e outra dimensão que corresponde às colunas da matriz.

São declarados assim:

```
matriz 3x3:
int a[][] = new int[3][3];
```

ou os elementos podem ser atribuídos diretamente assim:

```
int b[][] = {{2,3,1},{4,5,1},{2,4,8}}
```

exemplo:

```
public class MultMatriz {
    public static void main( String args[] )
    {
        int a[][] = {{2,3,1},{4,5,1},{2,4,8}};
        int mult[][] = new int[3][3];
        int soma = 0;

        for ( int i = 0; i < a.length; i++ )
            for (int j = 0; j < a[i].length; j++)
                mult[i][j] = 3*a[i][j];

        for ( int i = 0; i < a.length; i++ )
        {
            for (int j = 0; j < a[i].length; j++)
                System.out.print(" " + a[i][j]);
            System.out.println(" ");
        }
        for ( int i = 0; i < multi.length; i++ )
        {
            for (int j = 0; j < multi[i].length; j++)
                System.out.print(" " + multi[i][j]);
            System.out.println(" ");
        }
    }
}
```

4. Strings e seus métodos

Como nós já vimos uma String é uma sequência de caracteres. É o tipo de dados adequado para manipular palavras, frases, nomes, etc. A grande diferença é que uma String é um objeto da classe String. Em outras palavras, as Strings possuem métodos para manipulá-las.

O método `length()` retorna um número inteiro com o tamanho da String (número de caracteres, incluindo os espaços em branco).

```
String frase="Uma String é um objeto";
int tamanho = frase.length();
```

Para podermos comparar duas Strings, não podemos usar o operador `==`, pois Strings são objetos e não tipos primitivos. Temos que usar o método `equals()`

```
if (frase.equals("João"))
```

O método `charAt(pos)` retorna o caractere que está na posição `pos` da String (lembrando que o primeiro caractere da String está na posição 0).

```
char letra1 = frase.charAt(0);
char letra5 = frase.charAt(4);
```

O método `substring()` retorna uma parte da String original. No primeiro exemplo ele retorna a parte da String original a partir do terceiro caractere.

```
String parte = frase .substring(2);
```

No primeiro exemplo ele retorna a parte da String original do primeiro caractere até o décimo caractere.

```
String parte2 = frase.substring (0, 10);
```

O método `indexOf()` retorna a posição da primeira ocorrência de um caractere na String.

```
int posicao = frase.indexOf('c');
```

O método `replace()` troca todas as ocorrências de um caracteres por outro caractere.

```
frase.replace('a', 'i').
```

Exemplo: Um programa que imprime uma String invertida.

```
public class Palavras
{
    public static void main(String args[])
    {
        String palavra = "engenharia de software";
        System.out.println("a frase correta "+palavra);
        int i;
        int tamanho = palavra.length();
        System.out.print("a frase invertida é :");
        for(i = tamanho-1; i>=0; i--)
        {
            System.out.print(palavra.charAt(i));
        }
        System.out.println(" ");
    }
}
```

INTEGRAÇÃO:

1. fazer um programa em java que execute um sorteio entre 5 pessoas usando o método

```
int sorteio = (int) (Math.random()*5) e imprima o nome do vencedor.
```

2. fazer um programa em Java que leia 10 valores de idades de pessoas, calcule e imprima a média e a seguir imprima quais idades estão acima e quas estão abaixo da media.

3. Digite e rode o programa de exemplo das Exceções e mude as exceções capturadas para testar quando a captura funciona e quando a captura não funciona.

4. Construa um programa em java que leia 10 nomes e a seguir guarde-os em um array e leia um nome e verifique se ele faz parte da lista

5. Contrua um programa em java que some duas matrizes 3x3 e imprima as duas matrizes e o resultado

6. Fazer um programa em Java que leia uma frase e conte e escreva quantas palavras existem .

7. Fazer um programa em Java que leia um nome no formato comum, por exemplo “José Silva”, e o imprima no formato “Silva, J.”.

BIBLIOGRAFIA:

FURGERI, SÉRGIO. *Java 2 Ensino Didático. Desenvolvendo e Implementando Aplicações*. ed. Érica. São Paulo, 2002.

DEITEL, H. M. e DEITEL, P. J.. *Java, como Programar*. Ed. Bookman. Porto Alegre. 2001.

ARNOLD Ken, GOSLING James: "The Java Programming Language Second Edition", Addison-Wesley, 1997.