

Programação Orientada a Objetos

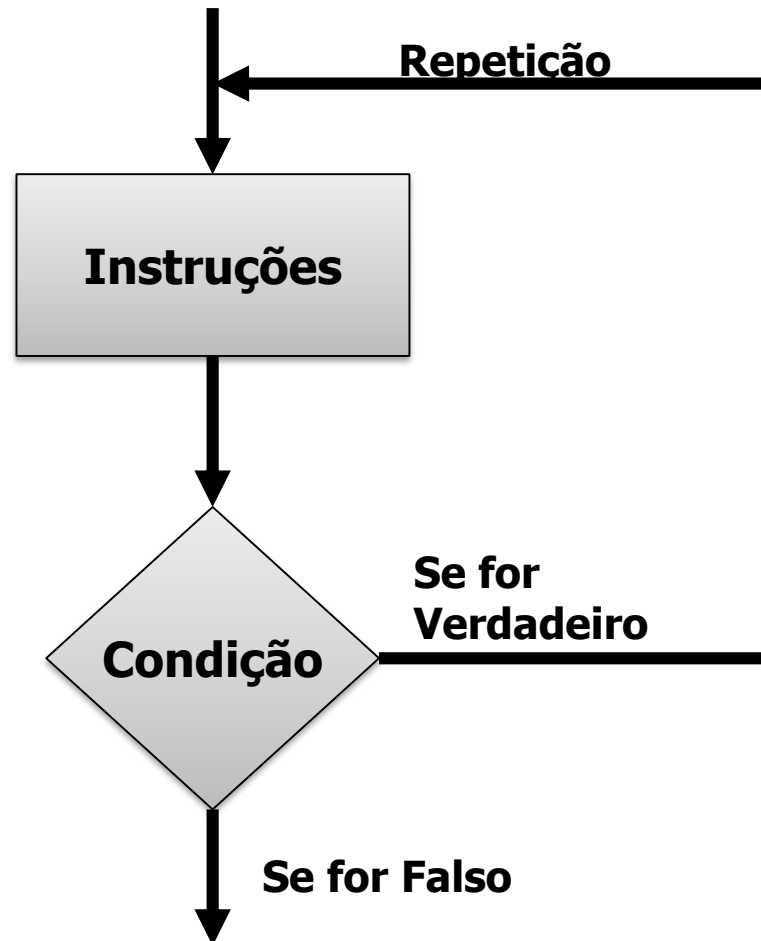
Semana 03 Programação Java

Reflexão

“A alegria não chega apenas no encontro do achado, mas faz parte do processo da busca. E ensinar e aprender não pode dar-se fora da procura, fora da boniteza e da alegria.”

Paulo Freire


Estrutura de Repetição For



Estrutura de Repetição For

```
1 public class For {  
2     public static void main(String[] args) {  
3         int num = 1;  
4         for (num = 1; num <= 10; num++) {  
5             System.out.println(num);  
6         }  
7     }  
8 }  
9 }
```

Estrutura de Repetição For



```
1 public class teste {  
2     public static void main(String[] args) {  
3         int num = 1;  
4         for (num = 1; num <= 10; num++) {  
5             System.out.println(num);  
6         }  
7     }  
8 }  
9 }
```

Estrutura de Repetição While

```
1 public class While {  
2     public static void main (String[] args) {  
3         int num = 1;  
4         while (num <= 10) {  
5             System.out.println(num);  
6             num++;  
7         }  
8     }  
9 }
```

Enquanto o Número
for Menor ou igual a 10
(Verifica ante de executar)

Mostra o número

Soma 1 ao número

Estrutura de Repetição Do-While

```
1 public class Do_While
2     public static void main (String[] args)
3     {
4         int num = 1;
5         do{
6             System.out.println(num);
7             num++;
8         } while (num <= 10);
9     }
10 }
```

Executa
(Não verifica antes)

Soma 1 ao número

Mostra o número

Enquanto o Número
for Menor ou igual a 10
(Só agora verifica)

Exceção em Java

- ✓ **Definição:** condição excepcional que altera o fluxo normal do programa.
- ✓ **Algumas causas:** falhas de hardware, exaustão de recursos e erros diversos.
- ✓ **Nomenclatura:** quando um evento excepcional ocorre em Java, diz-se que uma exceção será *lançada*. O código que é responsável por fazer algo com a exceção é chamado de *manipulador de exceções*, que captura a exceção lançada.

Vantagens

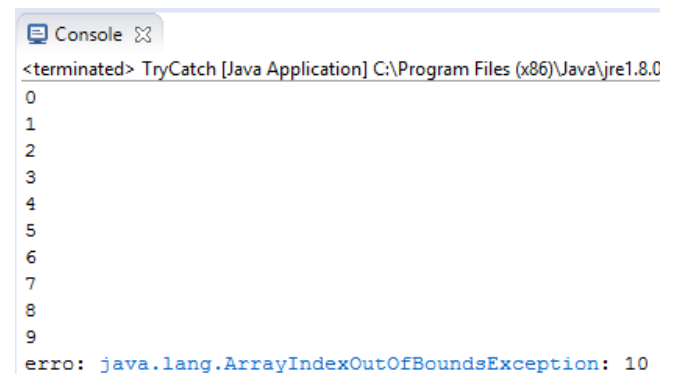
- ✓ Fácil detecção de erros sem a escrita de um código especial para testar valores retornados;
- ✓ Permite manter um código de manipulação de exceções nitidamente separado do código que gerará a exceção;
- ✓ Permite que o mesmo código de manipulação de exceções lide com as diferentes exceções possíveis.

A estrutura try catch

- ✓ A palavra chave **try** indica um bloco de código no qual possam ocorrer exceções (*região protegida*);
- ✓ Uma ou mais cláusulas **catch** permitem associar uma exceção específica a um bloco de código que a manipulará.

Exemplo try catch

```
3 public class TryCatch {
4     public static void main(String args[]) {
5         int[] array = new int[10];
6         try {
7             for (int i = 0; i <= 15; i++) {
8                 array[i] = i;
9                 System.out.println(i);
10            }
11        } catch (ArrayIndexOutOfBoundsException e) {
12            System.out.println("erro: " + e);
13        }
14    }
15 }
```



Console

<terminated> TryCatch [Java Application] C:\Program Files (x86)\Java\jre1.8.0

0
1
2
3
4
5
6
7
8
9
erro: [java.lang.ArrayIndexOutOfBoundsException](#): 10

Array

Estruturas de dados que consistem em itens de dados do mesmo tipo relacionados.

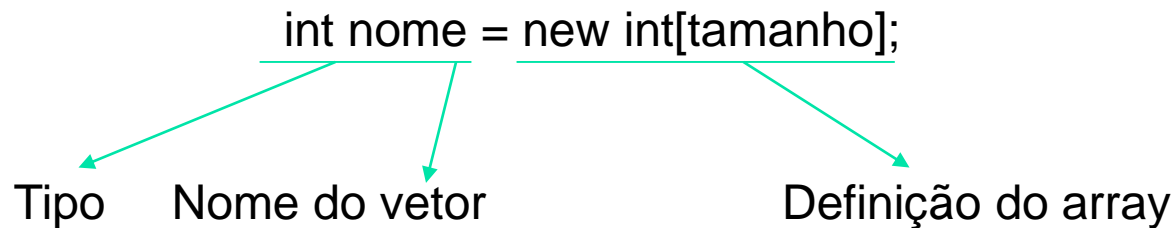
- ✓ Estruturas Unidimensionais
- ✓ Estruturas Multidimensionais

Permanecem com o mesmo tamanho depois de criados.

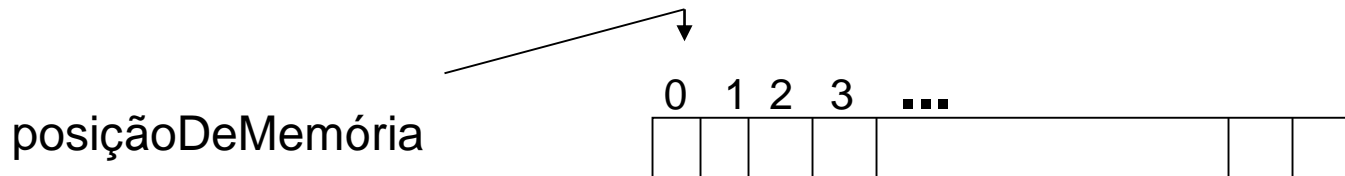
Vetores

Vetores são arrays unidimensionais.

Definição:

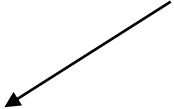


Inicialização do array: explícita ou implícita (tipos nativos são inicializados com o valor padrão do tipo)



Vetores

Todos os elementos do vetor passam a ter o mesmo nome, no caso 'c'



c [0]	- 128
c [1]	8
c [2]	0
c [3]	82
c [4]	64
c [5]	- 12
c [6]	65
c [7]	43
c [8]	76
c [9]	11
c [10]	0

```
...  
public static void main (String args[])  
{  
    int c = new int[11];  
    ...  
    c = {-128,8,0,82,64,-12,65,43,76,11};  
    // c[10] é zero por default (inicialização)  
    ...  
}
```

Vetores

Vetores são objetos que ocupam espaços contíguos de memória. O programador deve especificar o tipo, nome do vetor e utilizar o operador **new** para reservar o espaço necessário.

```
int [] c; // declaração do vetor
```

```
c = new int[12]; // declaração e reserva de espaço do do vetor
```

A reserva de espaço feita pelo operador **new** é automaticamente realizada pela máquina virtual Java.

Vetores podem ser declarados e inicializados ao mesmo tempo:

```
int c[] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

Declarações múltiplas

Um programa Java pode declarar vários vetores em uma única declaração.

```
int [] vetorA, vetorB, vetorC; // três vetores de inteiros
```

```
int vetorD = new int[121]; // criação de espaço com inicialização
```

```
String objTexto[] = new String [120], x[] = new String[20];
```

```
// objTexto contém 120 objetos da classe String
```

```
// x contém 20 objetos da classe String
```


Matriz

Matrizes são arrays multidimensionais.

Definição:

```
int nome = new int[linha][coluna];
```

Tipo

Nome da matriz

Definição do array

posiçãoDeMemória

	0	1	2	
0	00	01	02	
1	10	11	12	
2	20	21	22	
3	30	31	32	...

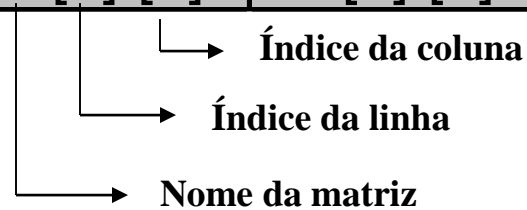
Matriz

Permite representar estruturas mais complexas, tais como tabelas (bidimensional) ou de maior ordem;

Para identificar um elemento específico da tabela devem ser especificados dois índices (subscritos) - o primeiro identifica a linha e o segundo a coluna (bidimensional);

Matriz

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	a [0] [0]	a [0] [1]	a [0] [2]	a [0] [3]
Linha 1	a [1] [0]	a [1] [1]	a [1] [2]	a [1] [3]
Linha 2	a [2] [0]	a [2] [1]	a [2] [2]	a [2] [3]



```
int array1[][] = { { 1, 2, 3 }, { 4, 5, 6 } };  
int array2[][] = { { 1, 2 }, { 3 }, { 4, 5, 6 } };  
int b[][];  
b = new int[3][4];
```

Matriz

```
String[ ][ ] desenhos = {  
    { "Flintstones", "Fred", "Wilma", "Pebbles", "Dino" },  
    { "Rubbles", "Barney", "Betty", "Bam Bam" },  
    { "Jetsons", "George", "Jane", "Elroy", "Judy", "Rosie", "Astro" },  
    { "Scooby Doo", "Scooby Doo", "Shaggy", "Velma", "Fred", "Daphne" }  
};
```

Exercícios

1. Reproduza o código abaixo, execute-o e responda:

```
3 public class TryCatch {  
4     public static void main(String args[]) {  
5         int[] array = new int[10];  
6         try {  
7             for (int i = 0; i <= 15; i++) {  
8                 array[i] = i;  
9                 System.out.println(i);  
10            }  
11        } catch (ArrayIndexOutOfBoundsException e) {  
12            System.out.println("erro: " + e);  
13        }  
14    }  
15 }
```

- O que podemos observar ao executar o programa acima?
- O que acontece se inserirmos a instrução try dentro do laço for?
- Qual a melhor maneira de inserir o bloco try..catch nesse exemplo? Justifique.

Exercícios

2. Faça uma aplicação que crie um bloco try...catch para uma classe que realiza a divisão entre dois números. Caso o resultado da divisão retorne zero, mostre uma mensagem informando que a divisão não pode ser realizada.
3. Faça uma aplicação que represente um jogo de loteria, a aplicação deve gerar 5 cartões de loteria com seis números em cada um, lembrando que esses números devem ser gerados de forma aleatória. Apresente em tela os valores sorteados em cada cartão.
4. Faça uma aplicação que realize a leitura de N elementos de um vetor A e um valor x. Criar o vetor B contendo os elementos do vetor A multiplicados por x.

Exercícios

5. Utilizando os métodos da classe String resolva:

a. Faça um programa que permita ao usuário digitar o seu nome e em seguida mostre o nome do usuário de trás para frente utilizando somente letras maiúsculas.

b. Nome na vertical. Faça um programa que solicite o nome do usuário e imprima-o na vertical. Exemplo:

F
U
L
A
N
O

c. Modifique o exercício da questão b para mostrar o nome em escala:

F
FU
FUL
FULA
FULAN
FULANO

Exercícios

6. Uma loja utiliza o código V para compras a vista e P para compras a prazo. Faça um programa que receba o código e o valor de 5 compras, calcule e mostre:

- a. O valor das compras a vista;
- b. O valor total das compras a prazo;
- c. O valor total das compras efetuadas;
- d. A média das compras a vista e a prazo;
- e. O valor máximo e mínimo das compras a vista e a prazo

Exercícios

7. Escreva um programa em Java que leia dois números representando os limites inferior e superior de um intervalo de números inteiros. Em seguida, o programa deve calcular e exibir na tela as seguintes informações:

- a) A soma dos inteiros que estão no intervalo;
- b) O número de inteiros dentro do intervalo;
- c) para cada número do intervalo, a indicação se ele é par ou ímpar.

Exercícios

8. Execute o programa a seguir, e depois responda:
- Quando se cria um array de valores numéricos, com que valor os seus elementos são inicializados?
 - Como se faz para saber qual o número de elementos de um array?

```
public class Array0{  
    public static void main(String args[]){  
        int cont;  
        int[] dados = new int[6];  
        System.out.println("O número de elementos do array dados é: "+  
            dados.length);  
        for (cont = 0; cont < dados.length; cont++){  
            System.out.println("O elemento de índice "+cont+  
                " vale "+ dados[cont]);  
        }  
    }  
}
```

Exercícios

9. Crie uma classe que faça a raiz quadrada de todos os elementos de um vetor que possua 6 posições. Exemplo de uma saída:

A raiz quadrada de 1 é 1.

A raiz quadrada de 4 é 2.

A raiz quadrada de 9 é 3.

A raiz quadrada de 16 é 4.

A raiz quadrada de 25 é 5.

A raiz quadrada de 36 é 6.

Exercícios

10. Considere que cada letra do alfabeto corresponde a um valor inteiro:
a=1, b=2, c=3,..., z=26

Elabore um programa que receba uma palavra e a converta em código, somando todos os valores correspondentes as letras. Por exemplo:

Palavra fornecida:

acabei = $1 + 3 + 1 + 2 + 5 + 9 = 21$

Saída em tela: **21**