

Arrays (parte 1)

Lilian Passos Scatalon
lpscatalon2@uem.br

Tipos de dados em Java

Os tipos em Java são divididos em tipos primitivos e tipos de referência

Os programas usam variáveis de referência para armazenar o endereço de objetos

```
Scanner input = new Scanner(System.in)
```

input é uma referência ao objeto da classe Scanner que foi instanciado pelo operador new

Arrays

Um array é um grupo de variáveis (chamadas de elementos) contendo valores do mesmo tipo

Arrays são objetos, então são considerados tipos de referência

Os elementos de um array podem ser de tipos primitivos ou de tipos de referência

Declaração e criação de arrays

```
int[] c; // declara a variável do array  
c = new int[12]; // cria o array; atribui à variável do array
```

```
int[] c = new int[12];
```

Quando um array é criado, cada elemento recebe um valor *default* (zero para tipos numéricos, false para boolean e null para referências)

Declaração e criação de arrays

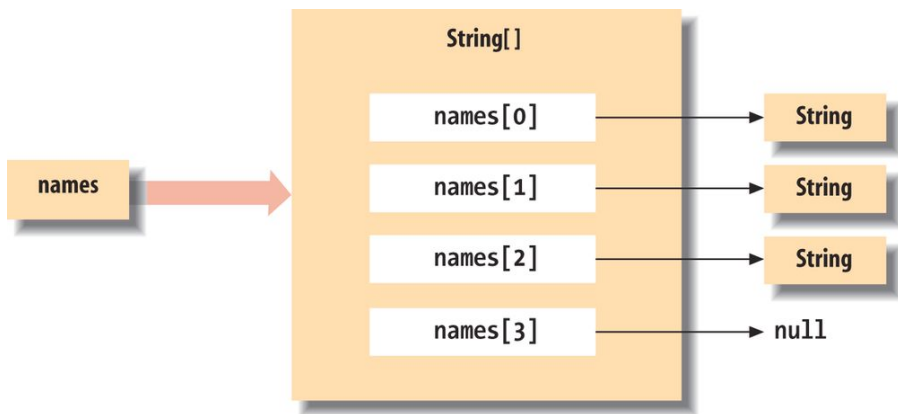
Para cada tipo de dados, há um tipo de array correspondente

```
String[] nomes;  
nomes = new String[4];
```

nomes é um vetor de 4 elementos, que podem ser acessados por
nomes[0], ... nomes[3]

```
nomes[0] = "Lara"  
nomes[1] = "Davi"  
nomes[2] = "Heitor"  
// nomes[3] = null
```

Representação do array em memória



nomes é uma referência para um array na memória

Para referenciar um elemento do array, usamos a referência do array e o índice do elemento

Tamanho do array

length é a variável do array que indica o seu tamanho

```
char [] alfabeto = new char [26];  
int tamAlfabeto = alfabeto.length; // tamAlfabeto == 26  
  
String [] mosqueteiros = { "um", "dois", "três" };  
int num = mosqueteiros.length; // num == 3
```

Iniciando um array

```
public class InitArray {  
    public static void main(String[] args) {  
  
        // inicializando o valor de cada elemento  
        int[] array = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};  
  
        System.out.printf("%s%8s%n", "Index", "Value"); // cabeçalho das colunas  
  
        // imprimindo o valor de cada elemento  
        for (int counter = 0; counter < array.length; counter++) {  
            System.out.printf("%5d%8d%n", counter, array[counter]);  
        }  
    }  
}
```


Index	Value
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20

Somando elementos de um array

```
public class SumArray {  
    public static void main(String[] args) {  
  
        int[] array = {87, 68, 94, 100, 83, 78, 85, 91, 76, 87};  
        int total = 0;  
        // soma o valor de cada elemento em total  
        for (int counter = 0; counter < array.length; counter++) {  
            total += array[counter];  
        }  
        System.out.printf("Total of array elements: %d%n", total);  
    }  
}
```

Total of array elements: 849

Usando os elementos de um array como contadores

```
import java.security.SecureRandom;

public class JogaDado {
    public static void main(String[] args) {

        SecureRandom random = new SecureRandom(); //gerador de números aleatórios

        int[] frequencias = new int[6]; // array de contadores de frequência

        // joga o dado 60.000.000 vezes
        // usa face resultante de cada jogada como índice
        for (int jogada = 1; jogada <= 60_000_000; jogada++) {
            int face = random.nextInt(6);
            frequencias[face]++;
        }

        System.out.printf("%s %10s %n", "Face", "Frequência");
        // imprime o valor de cada elemento do array
        for (int face = 0; face < frequencias.length; face++) {
            System.out.printf("%4d %10d %n", face+1, frequencias[face]);
        }
    }
}
```

Avaliando resultados de uma pesquisa de opinião

Suponha que foi realizada uma pesquisa de opinião com os alunos numa escala de 1 (“péssima”) a 5 (“excelente”) sobre a qualidade da comida do RU.

As avaliações de 20 alunos foram armazenadas em um array de inteiros, para determinar a frequência de cada avaliação.

```

public class PesquisaOpinioao {

    public static void main(String[] args) {
        // array com as respostas dos alunos
        int[] avaliacoes =
            {1, 2, 5, 4, 3, 5, 2, 1, 3, 3, 1, 4, 3, 3, 3, 2, 3, 3, 2, 14};
        int[] frequencias = new int[5]; // array de contadores de frequência
        // para cada resposta, usar o valor da resposta como índice no vetor
        // e incrementar o elemento correspondente

        for (int i = 0; i < avaliacoes.length; i++) {
            int atual = avaliacoes[i];
            frequencias[atual-1]++;
        }
        System.out.printf("%s %10s %n", "Avaliação", "Frequência");
        // imprimir o valor de cada elemento do vetor
        for (int avaliacao = 0; avaliacao < frequencias.length; avaliacao++) {
            System.out.printf("%6d %10d %n", avaliacao+1, frequencias[avaliacao]);
        }
    }
}

```

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 13 out of bounds for length 5
 at PesquisaOpinioao.main(PesquisaOpinioao.java:14)

Tratamento de exceção

Uma exceção indica que ocorreu um problema durante a execução de um programa

Tratamento de exceção ajuda a criar programas tolerantes a falhas

```
public class PesquisaOpiniaao {

    public static void main(String[] args) {
        // array com as respostas dos alunos
        int[] avaliacaoes =
            {1, 2, 5, 4, 3, 5, 2, 1, 3, 3, 1, 4, 3, 3, 3, 2, 3, 3, 2, 14};
        int[] frequencias = new int[5]; // array de contadores de frequência
        // para cada resposta, usar o valor da resposta como índice no vetor
        // e incrementar o elemento correspondente

        for (int i = 0; i < avaliacaoes.length; i++) {
            try {
                int atual = avaliacaoes[i];
                frequencias[atual-1]++;
            }
            catch (ArrayIndexOutOfBoundsException e) {
                System.out.println(e); // invokes toString method
                System.out.printf(" avaliacaoes[%d] = %d\n\n",
                                    i, avaliacaoes[i]);
            }
        }
        System.out.printf("%s%10s\n", "Avaliação", "Frequência");
        // imprimir o valor de cada elemento do vetor
        for (int avaliacao = 0; avaliacao < frequencias.length; avaliacao++) {
            System.out.printf("%6d%10d\n", avaliacao+1, frequencias[avaliacao]);
        }
    }
}
```

Tratamento de exceção

Bloco try: contém código que pode lançar (throw) uma exceção

Bloco catch: contém código que trata a exceção, caso ocorra

É possível ter muitos blocos catch para tratar diferentes tipos de exceção que podem ocorrer no bloco try correspondente

Enhanced for

O comando enhanced for itera pelos elementos de um array sem usar um contador

Assim, evita-se a possibilidade de usar índices fora dos limites do array

```
for (tipo elemento : array) {  
    ...  
}
```

```
public class TesteEnhancedFor {  
    public static void main(String[] args) {  
        int[] array = {87, 68, 94, 100, 83, 78, 85, 91, 76, 87};  
        int total = 0;  
        // soma o valor de cada elemento ao total  
        for (int numero : array) {  
            total += numero;  
        }  
        System.out.printf("Total: %d%n", total);  
    }  
}
```

Bibliografia

Deitel, P.J. e Deitel, H.M. **Java How to Program: Late Objects**. 11th edition. Pearson, 2020.