

# Programação II

Prof. Alcemir Rodrigues Santos

*Arrays e Collections*



## Agenda

- Arrays
- Collections
  - List
  - Set
  - Map
- Exercícios



9/17/19

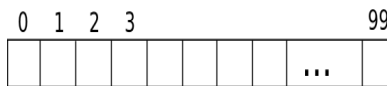
Programação II | Alcemir Santos

2



## Array

- Um array é uma porção de memória fixa e sequencial dividida em pedaços idênticos indexados a partir do 0. Em cada posição do array, podemos guardar um aluno. Na verdade, cada posição pode guardar uma referência para um objeto de um tipo.



9/17/19

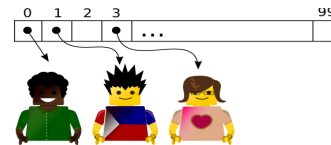
Programação II | Alcemir Santos

3



## Array

- A capacidade de um array é fixa e deve ser informada no momento da criação do array. Não é possível redimensionar um array em Java, teremos de contornar isso mais adiante.



9/17/19

Programação II | Alcemir Santos

4



## Declarando Variáveis do Tipo Vetor

- Na declaração de vetores deverão ser fornecidas três informações:
  - 1) o nome do vetor;
  - 2) o número de posições do vetor (seu tamanho);
  - 3) o tipo de dado que será armazenado no vetor. A declaração de um vetor para "inteiros", denome "vetor", em Java:
 

```
int vetor[];
```
- Embora declarado, o vetor não está pronto para uso, sendo necessário reservar espaço para seus elementos (uma operação de alocação de memória).
 

```
vetor = new int[10];
```

9/17/19

Programação II | Alcemir Santos

5



## Declarando Variáveis do Tipo Vetor

- Na linguagem Java um vetor é uma "classe", portanto, deve-se utilizar o método `new`, que ativa o método construtor correspondente, para criar instâncias, ou exemplares, da classe vetor do tipo selecionado (int, double, float, char, String, entre outros), por exemplo:
  - declarando um vetor para armazenar 10 números inteiros
 

```
int numeros = new int[10];
```
  - declarando um vetor para armazenar 10 valores do tipo real (ou monetários)
 

```
double salario = new double[10];
```
  - declarando um vetor para armazenar o nome dos 12 meses do ano
 

```
String msg = new String[12];
```

9/17/19

Programação II | Alcemir Santos

6



## Declarando Variáveis do Tipo Vektor (Exemplo)

```

5 public class VektorInt {
6     public static void main(String[] args) {
7         int VETOR[] = new int[10];
8         for (int i=0; i < 10; i++) {
9             VETOR[i] = i;
10        }
11
12        for (int i = 0; i < 10; i++){
13            System.out.println("Vetor [" + i + "] -> " + VETOR[i]);
14        }
15    }
16 }
17

```

Saída - POD (run) X

```

Vetor [0] -> 0
Vetor [1] -> 1
Vetor [2] -> 2
Vetor [3] -> 3
Vetor [4] -> 4
Vetor [5] -> 5
Vetor [6] -> 6
Vetor [7] -> 7
Vetor [8] -> 8
Vetor [9] -> 9
CONSTRUIDO COM SUCESSO (tempo total: 0 segundos)

```

9/17/19

Programa: II | Almir Santos

7

## Inicialização de vetores

- Java permite a inicialização de vetores no momento da declaração, por exemplo:
  - `String cidade[] = {"Picos", "Inhumas", "Roque"};`
- Isso significa que `cidade[0]` terá o valor Picos, `cidade[1]` terá o valor Inhumas, `cidade[2]` terá o valor Roque.
- Nota-se que não é necessário indicar o tamanho do vetor e também fazer a alocação de espaço através do operador `new`.
- O tamanho do vetor será conhecido através do campo `length`

9/17/19

Programa: II | Almir Santos

8

## Inicialização de vetores

```

1 public class VektorStringIni {
2
3     public static void main(String[] args) {
4         String cidade[] = {"Picos", "Inhumas", "Roque"};
5         int tam = cidade.length;
6         for (int i = 0; i < tam; i++){
7             System.out.println("cidade["+i+"] = " + cidade[i]);
8         }
9     }
10 }

```

Saída - POD (run) X

```

cidade[0] = Picos
cidade[1] = Inhumas
cidade[2] = Roque
CONSTRUIDO COM SUCESSO (tempo total: 0 segundos)

```

9/17/19

Programa: II | Almir Santos

9

## Estrutura for aprimorada

- Utilizada para percorrer os elementos de um vetor sem utilizar um contador:

```

for (parâmetro: nomeDoVetor) {
    instrução;
}

```

9/17/19

Programa: II | Almir Santos

10

## Estrutura for aprimorada

```

1 public class VektorIntIni {
2
3     public static void main(String[] args) {
4
5         int diasMeses[] = {1,2,3,4,5,6,7,8,9,10,11,12};
6
7         int tam = diasMeses.length;
8         for (int saida : diasMeses){
9             System.out.print(saida + " ");
10        }
11    }
12 }
13

```

Saída - POD (run) X

```

1 2 3 4 5 6 7 8 9 10 11 12 CONSTRUIDO COM SUCESSO (tempo total: 0 segundos)

```

9/17/19

Programa: II | Almir Santos

11

## Array

- Uma Lista implementada com array muitas vezes é denominada Vektor.
- Então criaremos uma classe chamada **Aluno** e outra classe **Vektor** que armazena os alunos em array e tem todas as operações de uma Lista, encapsulando o acesso a esta Array.

Aluno
-nome: String
-nota: double
+Aluno(nome, nota)
+gets e sets()

Vektor
-aluno[]: Aluno
-cont: int
+Vektor()
+adiciona(aluno: Aluno)
+Busca(posicao: int)
+remove(posicao: int)
+getLista()

9/17/19

Programa: II | Almir Santos

12

## Classe Aluno e Vetor

```

public class Aluno {
    private String nome;
    private double nota;

    public Aluno(String nome, double nota) {
        this.nome = nome;
        this.nota = nota;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public double getNota() {
        return nota;
    }

    public void setNota(double nota) {
        this.nota = nota;
    }

    public String toString() {
        return "Nome: " + this.nome + " Nota: " + this.nota;
    }
}

public class Vetor {
    private Aluno aluno[] = new Aluno[10];
    private int count;

    public Vetor() {
        count = 0;
    }

    public boolean adiciona(Aluno a) {
        // ...
    }

    public boolean remove(int pos) {
        // ...
    }

    public boolean busca(int pos) {
        // ...
    }

    public Aluno getAluno(int pos) {
        // ...
    }

    public void getLista() {
        // ...
    }
}

```

9/17/19

Programação II | Alcemir Santos

13

## Classe Vetor(Métodos)

```

public boolean adiciona(Aluno a) {
    if (count < 10) {
        aluno[count] = a;
        count++;
        return true;
    }
    return false;
}

public boolean remove(int pos) {
    if (pos >= 0 && pos < count) {
        return true;
    }
    return false;
}

public boolean busca(int pos) {
    if (pos >= 0 && pos < count) {
        for (int i = pos; i < count - 1; i++) {
            aluno[i] = aluno[i + 1];
        }
        aluno[count] = null;
        count--;
        return true;
    }
    return false;
}

public Aluno getAluno(int pos) {
    if (buscar(pos)) {
        return aluno[pos];
    }
    return null;
}

public void getLista() {
    for (int i = 0; i < count; i++) {
        System.out.println(aluno[i]);
    }
}

```

9/17/19

Programação II | Alcemir Santos

14

## Classe TesteVetor

```

public class TesteVetor {
    public static void main(String[] args) {
        //Criando 4 objetos.
        Aluno a1 = new Aluno("Thiago", 10);
        Aluno a2 = new Aluno("João", 8);
        Aluno a3 = new Aluno("Bastiana", 6);
        Aluno a4 = new Aluno("Lina", 4);
        Vetor vetor = new Vetor();

        vetor.adiciona(a1);
        vetor.adiciona(a2);
        vetor.adiciona(a3);
        vetor.adiciona(a4);
        vetor.getLista();
        //Removendo o objeto da posição 4
        System.out.println(vetor.remove(4));
        //Removendo o objeto da posição 2
        System.out.println(vetor.remove(2));
        vetor.getLista();
        //Insere um aluno na posição
        Aluno b = vetor.getAluno(1);
        if (b != null) {
            System.out.println(b);
        } else {
            System.out.println("posição inválida...");
        }
    }
}

```

9/17/19

Programação II | Alcemir Santos

15