

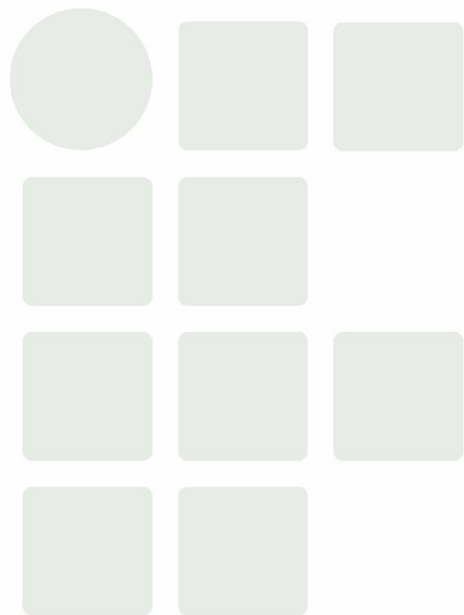


Programação O. O. II

Prof. Ronaldo Serpa da Rosa
ronaldo.rosa@bento.ifrs.edu.br

Conteúdo

- Revisão Vetores (abordagem tradicional)
- Nova abordagem Vetores



Revisão Vetores - Declaração

```
public class Vetor {
```

```
    private String[] elementos; //variavel que irá receber a ref. do vetor
```

```
    elementos = new String[5]; //criacao do um vetor de strings
```

```
    public static void main(String[] args) {
```

```
        elementos[0] = "ABC"; //armazenando valor na primeira posicao do vetor
```

```
        elementos[4] = "DEF"; //armazenando valor na ultima posicao do vetor
```

```
        System.out.println(elementos[0]); //mostra o valor da 1ª posicao do vetor
```

```
        System.out.println(elementos[4]); //mostra o valor da ultima posicao do
```

```
    }  
}
```

0	1	2	3	4
ABC				DEF



Revisão Vetores - Leitura

```
public class Vetor {
```

```
    private String[] elementos; //variavel que irá receber a ref. do vetor
```

```
    elementos = new String[5]; //criacao do um vetor de strings
```

```
    Scanner ler = new Scanner(System.in); //leitura por console
```

```
    public static void main(String[] args) {
```

```
        for(int i=0;i<elementos.length;i++){//laco para leitura do vetor
```

```
            elementos[i] = ler.NextString();//leitura por console e armazenamento na  
posicao i do vetor
```

```
        }
```

```
    }}
```

0

1

2

3

4



Revisão Vetores - Apresentação

```
public class Vetor {
```

```
    private String[] elementos; //variavel que irá receber a ref. do vetor
```

```
    elementos = new String[5]; //criacao de um vetor de strings
```

```
    public static void main(String[] args) {
```

```
        for(int i=0;i<elementos.length;i++){//laco para apresentacao do vetor
```

```
            System.out.println(elementos[i]);
```

```
        }
```

```
    }
```

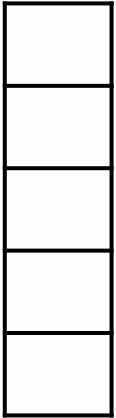
```
}
```

0

1

2

4



Revisão Vetores - Apresentação

```
public class Vetor {
```

```
    private String[] elementos; //variavel que irá receber a ref. do vetor
```

```
    elementos = new String[5]; //criacao do um vetor de strings
```

```
    public static void main(String[] args) {
```

```
        for(String elem:elementos){//laco para apresentacao do vetor
```

```
            System.out.println(elem);
```

```
        }
```

```
    }
```

```
}
```

0

1

2

4



Exercício 1 - Revisão Vetores

- Criar uma classe chamada RevisaoVetores (com método main);
- Declarar um vetor de inteiros de 5 posições;
- Criar um método (void) para realizar a leitura dos dados do vetor;
- Criar um método (void) para mostrar os dados do vetor;
- Criar um método (void) para ordenar em crescente o vetor;
- Criar um método (void) para ordenar em decrescente o vetor;
- Testar as operações no método main;

Exercício 2 - Revisão Vetores

Crie uma classe TestaSplit que reescreva uma frase com as palavras na ordem invertida. "Socorram-me, subi no ônibus em Marrocos" deve retornar "Marrocos em ônibus no subi Socorram-me,". Utilize o método split da String para te auxiliar. Esse método divide uma String de acordo com o separador especificado e devolve as partes em um array de String, por exemplo:

```
String frase = "Uma mensagem qualquer";
```

```
String[] palavras = frase.split(" ");
```

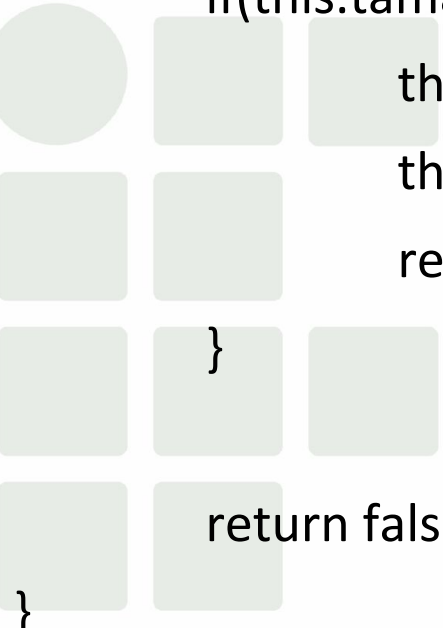
```
// Agora só basta percorrer o array na ordem inversa imprimindo as palavras
```


Nova Abordagem Vetores - Definindo Vetor

```
public class Vetor {  
    private String [] elementos;  
    private int tamanho;  
    public int getTamanho() {  
        return tamanho;}  
    public Vetor(int capacidade) {  
        this.elementos = new String[capacidade];  
        this.tamanho = 0;  
    }  
}
```

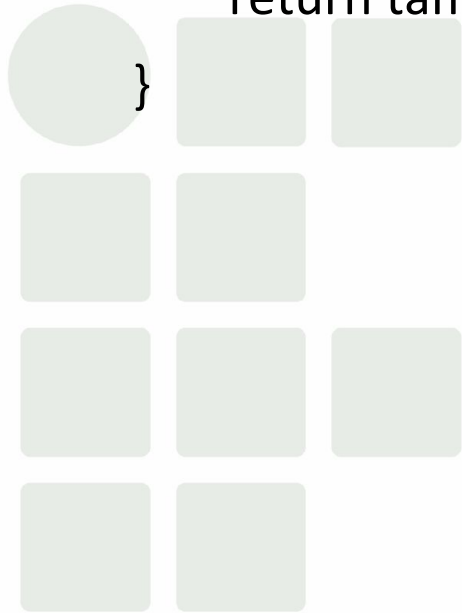
Adicionando elementos no final do Vetor

```
public boolean adicionar(String elemento) {  
    if(this.tamanho<this.elementos.length) {  
        this.elementos[this.tamanho] = elemento;  
        this.tamanho++;  
        return true;  
    }  
    return false;  
}
```



Pegando o tamanho do vetor

```
public int getTamanho() {  
    return tamanho;  
}
```

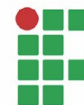


Imprimindo os elementos do vetor

@Override

```
public String toString() {  
    StringBuilder s = new StringBuilder();  
    s.append("[");  
    for(int i=0;i<tamanho;i++) {  
        s.append(this.elementos[i]);  
        s.append(",");  
    }  
    if(this.tamanho>0) {  
        s.append(this.elementos[this.tamanho-1]);  
    }  
    s.append("]");  
    return s.toString();}
```

```
@Override  
public String toString() {  
  
    return "Vetor [elementos=" +  
        Arrays.toString(elementos) + ", tamanho=" +  
        tamanho + "];"  
}
```



Buscando uma posição no vetor

```
public String busca(int posicao) {  
    if(!(posicao >=0 && posicao < tamanho)) {  
        throw new IllegalArgumentException("Posicao invalida");  
    }  
    return elementos[posicao];  
}
```



Buscando uma string no vetor

```
public int busca(String elemento) {  
    for(int i=0;i<tamanho;i++) {  
        if(elementos[i].equals(elemento)) {  
            return i;  
        }  
    }  
    return -1;  
}
```

Adicionando valor em qualquer posição do valor

```
public boolean adicionar(int posicao, String elemento) {  
    if(!(posicao >=0 && posicao < tamanho)) {  
        throw new IllegalArgumentException("Posicao invalida");}  
    for(int i=this.tamanho-1;i>=posicao;i--) {  
        this.elementos[i+1] = this.elementos[i]; }  
    this.elementos[posicao] = elemento;  
    this.tamanho++;  
    return true;  
}
```

Aumentando a capacidade do vetor

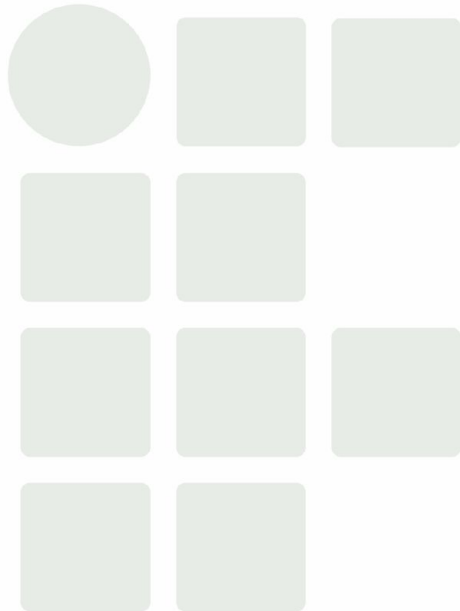
```
private void aumentaCapacidade() {  
    if(this.tamanho == this.elementos.length) {  
        String[] elementosNovos = new String[this.elementos.length*2];  
        for(int i=0;i<this.elementos.length;i++) {  
            elementosNovos[i] = this.elementos[i];  
        }  
        this.elementos = elementosNovos;  
    }  
}
```


Removendo um elemento do vetor

```
public void remove(int posicao) {  
    if(!(posicao >=0 && posicao < tamanho)) {  
        throw new IllegalArgumentException("Posicao invalida");  
    }  
    for(int i=posicao;i<tamanho-1;i++) {  
        this.elementos[i] = this.elementos[i+1];  
    }  
    this.tamanho--;  
}
```

Exercício 3

- Criar uma classe chamada Vetor e implementar os métodos de acordo com conteúdo apresentado do slide 9 até o slide 17



Exercício 4

- Criar uma classe chamada TestaVetor (com método main);
- Importar a classe Vetor (criada anteriormente);
- Realizar as operações abaixo utilizando os métodos da classe Vetor;
 - Criar um vetor com capacidade para 5 elementos;
 - Adicionar 3 elementos no vetor;
 - Imprimir o tamanho do vetor;
 - Imprimir todos os elementos do vetor;
 - Imprimir somente a posição 2 do vetor;
 - Realizar uma busca pelo conteúdo (texto) de elemento do vetor e imprimir a sua posição;
 - Adicionar um elemento na posição 1 do vetor;
 - Adicionar um elemento na posição 3 do vetor;
 - Adicionar um elemento na posição 5 do vetor;
 - Imprimir o vetor novamente;
 - Imprimir o tamanho do vetor novamente;

Referências

- GOODRICH, Michael T.; TAMASSIA, Roberto. Estruturas de dados e algoritmos em Java. 5. ed. Porto Alegre: Bookman, 2013
- <https://www.youtube.com/playlist?list=PLGxZ4Rq3BOBrgumpzz-l8kFMw2DLERdxi>