

Programação Orientada a Objetos mais sobre Java

Alexandre Mello

Fatec Campinas

2024

Roteiro

- 1 Variáveis
- 2 Laços e condicionais
- 3 Vetores e matrizes
- 4 SubStrings
- 5 I/O

Tipos e variáveis

```
int a, b, c;           // Declara três inteiros
int a = 10, b = 10;    // Exemplo de inicialização
byte B = 22;           // Inicializa um tipo byte
// Declara e atribui um valor a pi
double pi = 3.14159;
// Declara e inicializada com o valor 'a'
char a = 'a';
```

Variáveis locais

```
public class Teste {  
    public void caoIdade() {  
        int idade = 0;  
        idade = idade + 7;  
        System.out.println("Idade do cão é : " + idade);  
    }  
  
    public static void main(String args[]) {  
        Teste teste = new Teste();  
        teste.caoIdade();  
    }  
}
```

Variáveis de Instância

```
public class Empregado {  
    // Esta é uma variável de instância  
    public String name;  
  
    // Nome é atribuído no construtor  
    public Empregado (String empNome) {  
        nome = empNome;  
    }  
  
    // Imprime o nome  
    public void printaEmp() {  
        System.out.println("nome : " + nome );  
    }  
  
    public static void main(String args[]) {  
        Empregado emp = new Empregado("Alex");  
        emp.prinatEmp();  
    }  
}
```

Variáveis de Classe ou Estáticas

```
public class Empregado {  
    // salario é privada estática  
    private static double salario;  
  
    // DEPARTAMENTO é uma constante  
    public static final String DEPARTAMENTO = "Informática";  
  
    public static void main(String args[]) {  
        salario = 1000;  
        System.out.println(salario + " é a média salarial da " +  
            DEPARTAMENTO);  
    }  
}
```

Modificadores

```
public class className {  
    // ...  
}  
  
private boolean myFlag;  
static final double weeks = 9.5;  
protected static final int BOXWIDTH = 42;  
  
public static void main(String[] arguments) {  
    // código  
}
```

- 1 Variáveis ou métodos `static` existem independente da criação do objeto
- 2 O modifiador `final` impede que classes ou métodos sejam sobrescritos

Tipos de laços

- ① `while`
- ② `for`
- ③ `do ... while`

Melhoria no for

```
public class Teste {  
  
    public static void main(String args[]) {  
        int [] numeros = {10, 20, 30, 40, 50};  
  
        for(int x : numeros ) {  
            System.out.print( x );  
            System.out.print(",");  
        }  
        System.out.print("\n");  
        String [] nomes = {"João", "Maria", "Marcelo", "Lucia"};  
  
        for( String nome : nomes ) {  
            System.out.print( nome );  
            System.out.print(",");  
        }  
    }  
}
```

Condicionais

- ① if
- ② if ... else
- ③ if aninhado
- ④ switch

Condicional em apenas uma linha:

```
Exp1 ? Exp2 : Exp3;
```

Vetores

```
public class TestArray {  
  
    public static void main(String[] args) {  
        double[] myList = {1.9, 2.9, 3.4, 3.5};  
  
        // Imprime todos os elementos  
        for (int i = 0; i < myList.length; i++) {  
            System.out.println(myList[i] + " ");  
        }  
  
        // Soma todos os elementos  
        double total = 0;  
        for (int i = 0; i < myList.length; i++) {  
            total += myList[i];  
        }  
        System.out.println("Total é " + total);  
    }  
}
```

Vetores

```
public class TestArray {  
  
    public static void main(String[] args) {  
        double[] myList = {1.9, 2.9, 3.4, 3.5};  
  
        // Encontrando o maior elemento  
        double max = myList[0];  
        for (int i = 1; i < myList.length; i++) {  
            if (myList[i] > max) max = myList[i];  
        }  
        System.out.println("Maior é " + max);  
    }  
}
```

Vetores - foreach

```
public class TestArray {  
  
    public static void main(String[] args) {  
        double[] myList = {1.9, 2.9, 3.4, 3.5};  
  
        // Imprime os elementos  
        for (double element: myList) {  
            System.out.println(element);  
        }  
    }  
}
```

Vetores e métodos

```
public static int[] reverse(int[] l) {  
    int[] res = new int[l.length];  
  
    for (int i = 0, j = res.length - 1; i < l.length; i++, j--) {  
        res[j] = l[i];  
    }  
    return res;  
}
```

Matriz

```
public class Matriz {  
  
    public static void main(String[] args) {  
        int m[] [] = new int[4][5];  
        int c = 10;  
        for (int i = 0; i < m.length; i++) {  
            for (int j = 0; j < m[0].length; j++) {  
                m[i][j] = c;  
                c++;  
                System.out.print(m[i][j] + "\t");  
            }  
            System.out.println("");  
        }  
    }  
}
```

SubStrings

```
public class Test {  
  
    public static void main(String args[]) {  
        String Str = new String("Bem-vindo a Fatec Campinas");  
  
        System.out.print("Valore de retorno :" );  
        System.out.println(Str.substring(10, 15) );  
    }  
}
```

Entrada padrão 1

```
import java.io.*;

public class ReadConsole {

    public static void main(String args[]) {
        InputStreamReader cin = null;

        cin = new InputStreamReader(System.in);
        System.out.println("Digite caracteres, 'q' para
            terminar.");
        char c;
        do {
            c = (char) cin.read();
            System.out.print(c);
        } while(c != 'q');
    }
}
```

Entrada padrão 2

```
import java.util.Scanner;

public class ScanConsole {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Digite um inteiro e uma palavra: ");

        int n = scanner.nextInt();
        String f = scanner.nextLine();
        String p = scanner.next();
        System.out.println("Valor digitado = " + n);
        System.out.println("Texto: " + f);
        System.out.println("Palavra digitada = " + p);
    }
}
```
