

## Aula 03

### *Código 01:*

```
public class Objeto {  
    int instancias;  
    Objeto() {  
        instancias++;  
    }  
}
```

Divisão de Código, são dois arquivos

```
public class ExemploStatic {  
    public static void main(String[] args) {  
        Objeto obj1 = new Objeto();  
        System.out.println(obj1.instancias);  
  
        Objeto obj2 = new Objeto();  
        System.out.println(obj2.instancias);  
  
        Objeto obj3 = new Objeto();  
        System.out.println(obj3.instancias);  
  
    }  
}
```

Altere **int** instancias; da Classe Objeto para **static int** instancias; O que aconteceu?

### *Código 02:*

```
public class ExemploSobrecarga {  
    static void metodo(byte b) {  
        System.out.printf("\nEste método recebe um BYTE: %d", b);  
    }  
    static void metodo(short s) {  
        System.out.printf("\nEste método recebe um SHORT: %d", s);  
    }  
    static void metodo(int i) {  
        System.out.printf("\nEste método recebe um INT: %d", i);  
    }  
    static void metodo(long l) {  
        System.out.printf("\nEste método recebe um LONG: %d", l);  
    }  
    static void metodo(float f) {  
        System.out.printf("\nEste método recebe um FLOAT: %f", f);  
    }  
    static void metodo(double d) {  
        System.out.printf("\nEste método recebe um DOUBLE: %f", d);  
    }  
    static void metodo(String texto) {  
        System.out.printf("\nEste método recebe um BYTE: %s", texto);  
    }  
    static void metodo() {  
        System.out.printf("\nEste método não recebe parâmetros");  
    }  
  
    public static void main(String[] args) {  
        metodo(1);  
        metodo(1.0);  
        metodo(1.0f);  
        metodo(1L);  
        metodo("teste");  
    }  
}
```

```

        metodo();
        metodo((byte)1);
        metodo((short)1);
    }
}

```

### ***Código 03:***

```

public class DeclaraArray {
    public static void main(String[] args) {
        int [] numeros = new int[10];

        for (int i = 0; i < numeros.length; i++) {
            System.out.printf("%3d-%3d", i, numeros[i]);
        }
    }
}

```

**length:** todo vetor em Java possui esta propriedade que informa o número de elementos que possui

- length é uma propriedade read-only
- extremamente útil em blocos de repetição

```

for (int x = 0; x < vetor.length; x++) {
    vetor[x] = x*x;
}

```

### ***Código 04:***

```

public class Cliente {
    int idade;
    String nome;
    String telefone;

    Cliente(String n, int i, String t){
        idade = i;
        nome = n;
        telefone = t;
    }
}

public class ClienteArray {

    public static void main(String[] args) {
        Cliente[] clientes = new Cliente[5];
        clientes[0] = new Cliente("Jose", 35, "3231-2356");
        clientes[1] = new Cliente("Maria", 32, "3202-2526");
        clientes[2] = new Cliente("Carolina", 21, "3132-3536");
        clientes[3] = new Cliente("Luciana", 25, "3131-2323");
        clientes[4] = new Cliente("Joao", 55, "3535-5656");

        for (int i = 0; i < clientes.length; i++) {
            System.out.printf("\n\nNome: %s", clientes[i].nome);
            System.out.printf("\nIdade: %s", clientes[i].idade);
            System.out.printf("\nTelefone: %s", clientes[i].telefone);
        }
    }
}

```

### ***Código 05:***

```

public class ArgumentosMain {
    public static void main(String[] args) {
        System.out.println("Os argumentos apresentados foram:");

        for (int i = 0; i < args.length; i++) {
            System.out.println(args[i]);
        }
    }
}

```

### ***Código 06:***

Altere o código 04 para utilizar for aprimorado

### ***Código 07:***

```

public class Calculadora {
    static int somar(int ... parcelas){
        int resultado = 0;
        for(int v : parcelas){
            resultado += v;
        }
        return resultado;
    }
    public static void main(String[] args) {
        int soma = somar(1, 2, 3, 4, 5, 6);
        System.out.printf("Resultado: %d", soma);
    }
}

```

### ***Exercícios***

1. Implemente, na classe calculadora o método multiplicar de forma que ele pode receber uma quantidade de parâmetros aleatória.
2. O que são array's?
3. O que a palavra chave static faz?
4. Implemente uma classe que leia vários números e mostre quais são maiores, menores e iguais a média dos valores lidos (use array);
5. Implemente uma classe que leia um número e mostra todos os números primos menores ou iguais a este número;