

FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

Programação Orientada a Objetos

Métodos e Atributos *static* e Constantes

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Métodos Static

- Geralmente, os métodos são executados em resposta a chamadas a partir de um objetos
- Entretanto, pode ocorrer de um método realizar uma tarefa que não dependa da existência de um objeto ou de conteúdo relacionado ao objeto

Métodos Static: classe Math

- Um exemplo de classe que utiliza constantes e métodos *static* é a Math
 - Ela implementa algumas funções matemáticas

```
System.out.println("Valor de PI: " + Math.PI);  
System.out.println("Raiz Quadrada: "+Math.sqrt(900));  
System.out.println("Graus para Radianos: "+Math.toRadians(45));  
System.out.println("Seno: "+Math.sin( Math.toRadians(45) ));
```

```
Valor de PI: 3.141592653589793  
Raiz Quadrada: 30.0  
Graus para Radianos: 0.7853981633974483  
Seno: 0.7071067811865475
```

Métodos Static

- Esses métodos se aplicam à classe em que é declarado e é conhecido como **método static** ou **método da classe**

```
System.out.println("Valor de PI: " + Math.PI);  
System.out.println("Raiz Quadrada: "+Math.sqrt(900));  
System.out.println("Graus para Radianos: "+Math.toRadians(45));  
System.out.println("Seno: "+Math.sin( Math.toRadians(45) ));
```

```
Valor de PI: 3.141592653589793  
Raiz Quadrada: 30.0  
Graus para Radianos: 0.7853981633974483  
Seno: 0.7071067811865475
```

Métodos Static

- É comum a classe ter um conjunto de métodos **static** para realizar tarefas comuns
 - Por exemplo, utilizamos métodos **static** no JOptionPane
 - JOptionPane.showInputDialog("Digite o saldo da conta")
 - Não foi necessário instanciar um objeto do tipo JOptionPane
 - JOptionPane jp = new JOptionPane();
 - jp.showInputDialog("Digite o saldo da conta");

Métodos Static

- Para declarar um método **static** basta colocar a palavra **static** antes do tipo de retorno do método
 - Exemplo: `public static float calculaMedia(int [] vetor){...}`
- Para utilizar o método basta colocar o nome da classe um '.' e o nome do método **static**
 - Exemplo: `NomeDaClasse.calculaMedia(vetor);`
- O método é invocado sem a necessidade de criar uma instância (objeto) da classe

Exemplo de classe com métodos **static**

Classe Util

- É comum a implementação de uma classe com métodos **static**, os quais podem ser usados por diferentes classes (partes do código)
- Por exemplo, podemos implementar uma classe com um método para calcular a média de números ou encontrar o maior valor de um vetor


```
public class Util {
    public static float media(int [] vetor) {
        int soma = 0;
        for ( int i=0; i<vetor.length; i++) {
            soma = soma + vetor[i];
        }
        return (float) soma / vetor.length;
    } //casting para retornar como float
    public static int maior(int []vetor) {
        int maior = vetor[0];
        for (int i=1; i<vetor.length; i++) {
            if (vetor[i] > maior) {
                maior = vetor[i];
            }
        }
        return maior;
    }
}
```

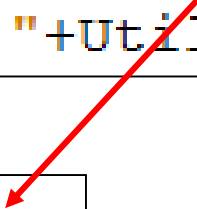
Classe Util

```
///Exemplos de metodos estaticos  
int vet[] = {4, 3, 9, 2, 9};  
System.out.println("Media: "+Util.media(vet));  
System.out.println("Maior: "+Util.maior(vet));
```

```
Media: 5.4  
Maior: 9
```

Classe Util

```
///Exemplos de metodos estaticos  
int vet[] = {4, 3, 9, 2, 9};  
System.out.println("Media: "+Util.media(vet));  
System.out.println("Maior: "+Util.maior(vet));
```

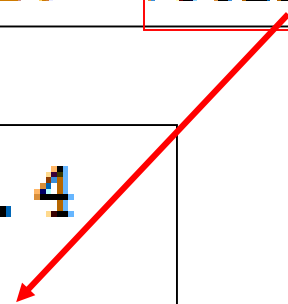


```
Media: 5.4  
Maior: 9
```

Classe Util

```
///Exemplos de metodos estaticos  
int vet[] = {4, 3, 9, 2, 9};  
System.out.println("Media: "+Util.media(vet));  
System.out.println("Maior: "+Util.maior(vet));
```

```
Media: 5.4  
Maior: 9
```



Atributos **static**

Atributos static

- Assim como os métodos, os atributos *static* pertencem à classe e não dependem de um objeto
 - Eles já possuem uma área alocada de memória
- Um atributo *static* tem seu conteúdo compartilhado com todos os objetos
- Se o atributo for public, poderá ser acessado e modificado sem que um objeto seja instanciado
 - ClasseExemplo.atributo = 10;

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade) {  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Atributos static

```
public class Funcionario {
    private static int codAtual=0;
    private int cod;
    private String nome;

    public Funcionario(String nome, int idade){
        this.cod = this.codAtual;
        this.codAtual++;
        this.nome = nome;
        this.idade = idade;
    }
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);
Funcionario f2 = new Funcionario("Func2", 22);
Funcionario f3 = new Funcionario("Func3", 44);

f1.exibir();
f2.exibir();
f3.exibir();
```


Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 0

Funcionario

0
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Para simplificar o atributo idade não está sendo apresentado ¹⁷

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 1

Funcionario

0
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 1

Funcionario

0

"Func1"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 1

Funcionario

1
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0
"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 2

Funcionario

1
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0
"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 2

Funcionario

1

"Func2"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

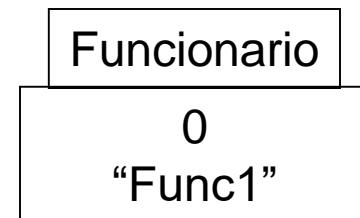
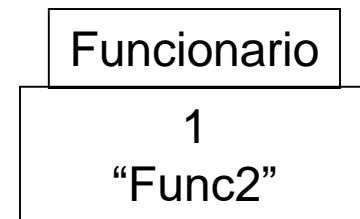
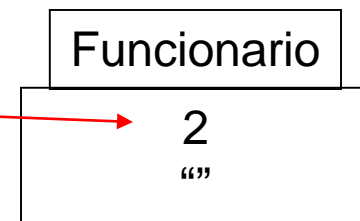
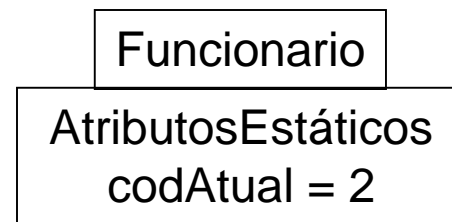
Funcionario

0

"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```



```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

AtributosEstáticos
codAtual = 3

Funcionario

2
""

Funcionario

1
"Func2"

Funcionario

0
"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

AtributosEstáticos
codAtual = 3

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 3

Cod: 0
Nome: Func1

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

2
"Func3"

Funcionario

1
"Func2"

Funcionario

0
"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 3

Cod: 1
Nome: Func2

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

AtributosEstáticos
codAtual = 3

Cod: 2

Nome: Func3

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade) {  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 1
Idade = 0

Funcionario

0
""

O que acontece se o atributo *idade* for colocado como *static*?

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 0
idade = 0

Funcionario

0
""



```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 1
idade = 0

Funcionario

0
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 1
idade = 0

Funcionario

0

"Func1"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```


Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 1
idade = 33

Funcionario

0
"Func1"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 1
idade = 33

Funcionario

1
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0
"Func1"

34



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade) {  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

AtributosEstáticos
codAtual = 2
idade = 33

Funcionario

1
""

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0
"Func1"

35



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 2
idade = 33

Funcionario

1

"Func2"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0

"Func1"

36



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Funcionario

Atributos Estáticos
codAtual = 2
idade = 22

Funcionario

1

"Func2"

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

0

"Func1"

37



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

Atributos Estáticos
codAtual = 2
idade = 22

Funcionario

2
""

Funcionario

1
"Func2"

Funcionario

0
"Func1"

38



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

AtributosEstáticos
codAtual = 3
Idade = 22

Funcionario

2
""

Funcionario

1
"Func2"

Funcionario

0
"Func1"

39



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

Atributos Estáticos
codAtual = 3
idade = 22

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

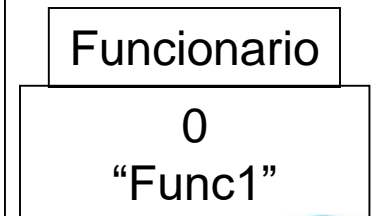
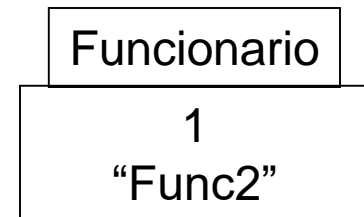
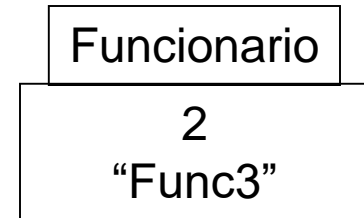
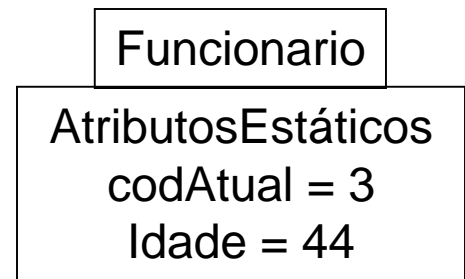
40



Atributos static

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade;  
    public Funcionario(String nome, int idade){  
        this.cod = this.codAtual;  
        this.codAtual++;  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```



41



Atributos static

```
public void exhibir(){  
    System.out.println("Cod: "+cod);  
    System.out.println("Nome: "+nome);  
    System.out.println("Idade: "+idade);  
}
```

```
Cod: 0  
Nome: Func1  
Idade: 44
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

Atributos Estáticos
codAtual = 3
Idade = 44

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

42



Atributos static

```
public void exhibir(){  
    System.out.println("Cod: "+cod);  
    System.out.println("Nome: "+nome);  
    System.out.println("Idade: "+idade);  
}
```

```
Cod: 1  
Nome: Func2  
Idade: 44
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

Atributos Estáticos
codAtual = 3
Idade = 44

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

43



Atributos static

```
public void exhibir(){  
    System.out.println("Cod: "+cod);  
    System.out.println("Nome: "+nome);  
    System.out.println("Idade: "+idade);  
}
```

```
Cod: 2  
Nome: Func3  
Idade: 44
```

```
Funcionario f1 = new Funcionario("Func1", 33);  
Funcionario f2 = new Funcionario("Func2", 22);  
Funcionario f3 = new Funcionario("Func3", 44);  
f1.exibir();  
f2.exibir();  
f3.exibir();
```

Funcionario

Atributos Estáticos
codAtual = 3
Idade = 44

Funcionario

2

"Func3"

Funcionario

1

"Func2"

Funcionario

0

"Func1"

44



Constantes

static final

Constantes

- Um exemplo de classe que utiliza constantes é a Math
 - Nesse caso, PI é uma constante da classe Math

```
System.out.println("Valor de PI: " + Math.PI);  
System.out.println("Raiz Quadrada: "+Math.sqrt(900));  
System.out.println("Graus para Radianos: "+Math.toRadians(45));  
System.out.println("Seno: "+Math.sin( Math.toRadians(45) ));
```

```
Valor de PI: 3.141592653589793  
Raiz Quadrada: 30.0  
Graus para Radianos: 0.7853981633974483  
Seno: 0.7071067811865475
```

Constantes

- Em java, uma constante pode ser definida com a palavra-chave *final*
- Assim, é definido que o conteúdo da variável não poderá ser alterado
 - O identificador de uma constante deve possuir letras maiúsculas

```
final double SALARIO = 1050.56;  
System.out.println("Salário: "+SALARIO);  
SALARIO = 34;
```

```
cannot assign a value to final variable SALARIO  
----  
(Alt-Enter shows hints)
```

Constantes

```
public class Funcionario {
    private static int codAtual=0;
    private int cod;
    private String nome;
    private static int idade; //remover static para exemplificar
    private static final float SALARIO_BASE = 500.00f; //constante privada
    public static final String EMPRESA = "Supermercados Associados";
}
```

- **final** define que a variável é uma constante e que não pode ter seu valor alterado
 - Por padrão o identificador de uma constante é formado por letras maiúsculas
 - Exemplo: SALARIO_BASE, EMPRESA
 - **private** define que ela é acessível somente dentro da própria classe
 - Funcionario.SALARIO_BASE ou SALARIO_BASE
 - **public** define que ela é acessível em qualquer parte do programa
 - Funcionario.EMPRESA ou EMPRESA (dentro da própria classe)

Constantes

- Exemplo de uso dentro da própria classe
 - Constante private

```
public void exibir() {  
    System.out.println("=====");  
    System.out.println("Cod: "+cod);  
    System.out.println("Nome: "+nome);  
    System.out.println("Idade: "+idade);  
    System.out.println("Salario Base: R$ "+SALARIO_BASE);  
    System.out.println("=====\n");  
}
```

Constantes

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade; //remover static para exemplificar  
    private static final float SALARIO_BASE = 500.00f; //constante privada  
    public static final String EMPRESA = "Supermercados Associados";  
  
    System.out.println("=====");  
    System.out.println("Cod: "+cod);  
    System.out.println("Nome: "+nome);  
    System.out.println("Idade: "+idade);  
    System.out.println("Salario Base: R$ "+SALARIO_BASE);  
    System.out.println("=====\n");  
}
```

Constantes

- Exemplo de uso fora da classe onde a constante foi definida
 - Constante public

```
public static void main(String args[]){  
    Funcionario f1 = new Funcionario("Func1", 33);  
    Funcionario f2 = new Funcionario("Func2", 22);  
    Funcionario f3 = new Funcionario("Func3", 44);  
    System.out.println("Funcionarios da Empresa: "+Funcionario.EMPRESA);  
    f1.exibir();  
    f2.exibir();  
    f3.exibir();  
}
```

Constantes

```
public class Funcionario {  
    private static int codAtual=0;  
    private int cod;  
    private String nome;  
    private static int idade; //remover static para exemplificar  
    private static final float SALARIO_BASE = 500.00f; //constante privada  
    public static final String EMPRESA = "Supermercados Associados";
```

```
public static void main(String args[]){  
    Funcionario f1 = new Funcionario("Func1", 33);  
    Funcionario f2 = new Funcionario("Func2", 22);  
    Funcionario f3 = new Funcionario("Func3", 44);  
    System.out.println("Funcionarios da Empresa: "+Funcionario.EMPRESA);  
    f1.exibir();  
    f2.exibir();  
    f3.exibir();
```

Referências

BIBLIOGRAFIA BÁSICA

1. SINTES, A., Aprenda programação orientada a objetos em 21 dias, Pearson Education do Brasil, 2002.
2. VAREJÃO, F., Linguagens de programação : Java, C e C++ e outras : conceitos e técnicas, Campus, 2004.
3. DEITEL, H. M., DEITEL, P. J., **Java:** como programar, São Paulo: Pearson Education do Brasil, 2010. 1144p.
4. DEITEL, H. M., DEITEL, P. J., **Java:** como programar, Porto Alegre: Bookman, 2003. 1386p.
5. SAVITCH, W. J., C++ absoluto, Pearson Education : Addison Wesley, 2004.

BIBLIOGRAFIA COMPLEMENTAR

1. BERMAN, A. M. *Data Structures via C++: Objects by Evolution*, Oxford University Press Inc., 1997.
2. BARNES, D.J. & KÖLLING, M., Programação orientada a objetos com Java, Pearson Education : Prentice Hall, 2004.
3. DEITEL, H. M. e DEITEL, P. J. *C++: Como Programar*, Bookman, 2001.
4. GILBERT, R. F. e FOROUZAN, B. A. *Data Structures: A Pseudo Approach with C++*, Brooks/Cole Thomson Learning, 2001.
5. MUSSER, D. R. e SAINI, A. *STL Tutorial and Reference Guide: Programming with the Standard Template Library*, Addison-Wesley, 1996.
6. SEBESTA, R. W. *Conceitos de Linguagem de Programação*, 4ª Ed., Bookman, 2003.
7. SEDGEWICK, R. *Algorithms in C++*, Addison-Wesley, 2002.
8. STROUSTRUP, B. *A Linguagem de Programação C++*, 3ª Ed., Bookman, 2000.