

Linguagem de Programação 1

this e Membros Estáticos

Prof. Fábio José Rodrigues Pinheiro

OUTUBRO DE 2021



**INSTITUTO
FEDERAL**

Catarinense

Campus
Videira

- `this` é uma referência para o próprio objeto e pode ser usado para garantir que estará referenciando os membros de uma classe e não uma variável local



- `this` é uma referência para o próprio objeto e pode ser usado para garantir que estará referenciando os membros de uma classe e não uma variável local

```
1 public class Pessoa{
2     private String nome;
3     private int idade;
4
5     public void setNome(String nome){
6         this.nome = nome; // o uso do this é obrigatório
7     }
8
9     public void setIdade(int i){
10        // o uso do this não é obrigatório para o entendimento
11        idade = i;
12    }
13 }
```



Palavra reservada: `this`: Invocar outros métodos da classe

```
1 public class Retangulo{
2     private int x, y;
3     private int largura, altura;
4
5     public Retangulo(){
6         this(0,0,10,10);
7     }
8     public Retangulo(int x, int y, int l, int a){
9         this.x = x; this.y = y; this.largura = l; this.altura = a;
10    }
11    public void setX(int x){
12        this.x = x;
13    }
14    public void setY(int y){
15        this.y = y;
16    }
17    public void setXY(int x, int y){
18        this.setX(x); this.setY(y);
19    }
20 }
```



Membros estáticos

Atributos não estáticos

Cada instância da classe terá uma cópia distinta desse atributo

```
62 public class AgenciaBancaria{  
63     private int totalAgencias;  
64  
65     public AgenciaBancaria(){  
66         this.totalAgencias++;  
67     }  
68  
69     public int getTotal(){  
70  
71         return this.totalAgencias;  
72     }  
73 }
```

```
1 AgenciaBancaria b, c;  
2 b = new AgenciaBancaria();  
3 c = new AgenciaBancaria();  
4 int r = b.getTotal();  
5 int t = c.getTotal();
```

■ Qual será o valor de r e t?



Atributos não estáticos

Cada instância da classe terá uma cópia distinta desse atributo

```
62 public class AgenciaBancaria{
63     private int totalAgencias;
64
65     public AgenciaBancaria(){
66         this.totalAgencias++;
67     }
68
69     public int getTotal(){
70
71         return this.totalAgencias;
72     }
73 }
```

```
1 AgenciaBancaria b, c;
2 b = new AgenciaBancaria();
3 c = new AgenciaBancaria();
4 int r = b.getTotal();
5 int t = c.getTotal();
```

■ Qual será o valor de r e t?

■ r = 1

■ t = 1



Membros de classe estáticos

Membro de classe que pertença a classe e não a uma instância específica da classe

- Método estático não pode acessar membros não estáticos

```
77 public class AgenciaBancaria{
78     private static int
        totalAgencias;
79
80     public AgenciaBancaria(){
81         this.totalAgencias++;
82     }
83
84     public static int getTotal(){
85         return totalAgencias;
86     }
87 }
```

```
1 AgenciaBancaria b, c;
2 b = new AgenciaBancaria();
3 c = new AgenciaBancaria();
4 int r = b.getTotal();
5 int t = c.getTotal();
6 int h = AgenciaBancaria.getTotal()
    ;
7 System.out.print(r+", "+t+", "+h);
```

- Qual será o valor de r, t e h?



Membros de classe estáticos

Membro de classe que pertença a classe e não a uma instância específica da classe

- Método estático não pode acessar membros não estáticos

```
77 public class AgenciaBancaria{
78     private static int
        totalAgencias;
79
80     public AgenciaBancaria(){
81         this.totalAgencias++;
82     }
83
84     public static int getTotal(){
85         return totalAgencias;
86     }
87 }
```

```
1 AgenciaBancaria b, c;
2 b = new AgenciaBancaria();
3 c = new AgenciaBancaria();
4 int r = b.getTotal();
5 int t = c.getTotal();
6 int h = AgenciaBancaria.getTotal()
    ;
7 System.out.print(r+", "+t+", "+h);
```

- Qual será o valor de r, t e h?

- r = 2
- t = 2
- h = 2



Membros de classe estáticos

```
1 public class AgenciaBancaria2 {
2     private static int totalAgencias = 0;
3     private int numero = 0;
4
5     public AgenciaBancaria2(int num){
6         totalAgencias++; this.numero = num;
7     }
8     // OK
9     public int getNumero() { // não estático
10         return numero; // não estático
11     }
12     // OK
13     public int getTotal(){ //não estático
14         return totalAgencias; // estático
15     }
16     // ERRO DE SINTAXE
17     public static int retornaNumero(){ //estático
18         return numero; // não estático
19     }
20 }
```



Modificador **final** – para definir constantes

Pode ser usado em atributos, métodos ou em variáveis locais

- Uma variável após ter recebido um valor, esse não poderá ser alterado
- Métodos não poderão ser sobrescritos (conceito de herança)

```
1 public class Celular{  
2     private final String FREQUENCIA = "1800";  
3     private final int SERIAL;  
4  
5     public Celular(int s){  
6         this.SERIAL = s;  
7     }  
8     public final void iniciarChamada(){  
9         // ....  
10    }  
11 }
```



Exercício: classe utilitária `java.lang.Math`

<http://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

- 1 Para obter a raiz quadrada do número 4, basta:
`double d = Math.sqrt(4);`
- 2 `Math.PI` é uma constante que contém o valor aproximado de π .
Para imprimir este valor, basta:
`System.out.println(Math.PI);`

Responda

Dos conceitos apresentados nesta aula, quais deles a classe `java.lang.Math` faz uso?



- Modifique o atributo da classe `Contador` para que seja um membro estático. O que mudou? Afeta o resultado da aplicação?
- E na classe `ContaCorrente`, algum atributo poderia ser estático? O que aconteceria se `saldo` fosse estático?



Agora é Java!

Classe String e StringBuilder

- Igualdade de Strings (?!)
 - O método (equals()) será sempre a melhor garantia
- Classe StringBuilder

```
1
2 String s = "Ciência";
3
4 // String são imutáveis
5 s += " da Computação"; // JVM cria novos objetos na memória
6
7 // Classe mais adequada quando se deseja concatenar strings
8
9 StringBuilder sb = new StringBuilder("Ciência");
10 sb.append(" da Computação");
11 String res = sb.toString();
```



Lendo arquivo texto

```
1 import java.util.Scanner;
2 public class ConverteString{
3     public static void main(String[] args){
4         Scanner entrada = new Scanner(System.in);
5         // Enquanto houver nova linha
6         while(entrada.hasNext()){
7             String linha = entrada.nextLine();
8             System.out.println(linha.toUpperCase());
9         }
10    }
11 }
```

```
1 java ConverteString < entrada.txt
```

