



Laboratório de Linguagens de Programação

Escopo de Bloco e Chamadas Encadeadas

André Pfeiffer
Letícia Bacoccoli
Raphael Americano



Laboratório de Linguagens de Programação

Escopo de bloco

- A variável pode existir em diversos escopos
- Objeto, método e bloco
- Escopo != modificador de acesso



Laboratório de Linguagens de Programação

Escopo de objeto

```
3 public class Dog {  
4     String nome;  
5     int idade;  
6 }  
7
```

Laboratório de Linguagens de Programação

Escopo de método

Escopo de objeto

```
3 public class Dog {  
4     String nome;  
5     int idade;  
6     public void potenciaLatido(){  
7         int potencia = 2;  
8         System.out.println(potencia + idade);  
9     }  
10 }  
11
```

Laboratório de Linguagens de Programação

Escopo de método

Escopo de objeto

```
3 public class Dog {  
4     String nome;  
5     int idade;  
6     public void potenciaLatido(){  
7         int potencia = 2;  
8         System.out.println(potencia + idade);  
9     }  
10    public void potenciaCao(){  
11        System.out.println(nome + " tem potência " + potencia);  
12    }  
13 }  
14
```

Laboratório de Linguagens de Programação

Escopo de bloco

Escopo de método

Escopo de objeto

```
3 public class Dog {  
4     String nome;  
5     int idade;  
6     public void potenciaLatido(){  
7         int potencia = 2;  
8         for(int i = 0; i < (idade + potencia); i++){  
9             System.out.println(potencia + idade + i);  
10        }  
11    }  
12 }  
13
```

Laboratório de Linguagens de Programação

Escopo de bloco

Escopo de método

Escopo de objeto

```
3 public class Dog {  
4     String nome;  
5     int idade;  
6     public void potenciaLatido(){  
7         int potencia = 2;  
8         for(int i = 0; i < (idade + potencia); i++){  
9             System.out.println(potencia + idade + i);  
10        }  
11        System.out.println(potencia + idade - i);  
12    }  
13 }
```

Laboratório de Linguagens de Programação

```
Dog.java
1 package dog;
2
3 public class Dog {
4     public static void latidos(){
5         for (int i = 0; i < 2; i++){
6             System.out.println("au");
7         } // portugues
8         for (int i = 0; i < 2; i++){
9             System.out.println("boj");
10        } // esperanto
11        for (int i = 0; i < 2; i++){
12            System.out.println("bau");
13        } // italiano
14        for (int i = 0; i < 2; i++){
15            System.out.println("woof");
16            for (int j = 0; j < 2; j++){
17                System.out.println("ruff");
18                for (int l = 0; l < 2; l++){
19                    System.out.println("arf");
20                }
21            }
22        } // ingles
23    }
24 }
25
```

```
TestDog.java
1 package dog;
2
3 public class TestDog {
4
5     public static void main(String[] args) {
6         Dog.latidos();
7     }
8 }
9
```

```
Problems @ Javadoc Declaration Console
<terminated> TestDog [Java Application] C:\Program Files\Java
au
au
boj
boj
bau
bau
woof
ruff
arf
arf
ruff
arf
arf
woof
ruff
arf
arf
ruff
arf
arf
```




Laboratório de Linguagens de Programação

Chamadas Encadeadas

- É apenas uma maneira diferente de executar código
- É muito utilizada
- Pode ser difícil de entender sem prática

Laboratório de Linguagens de Programação

Chamada encadeada

```
StringBuffer sb = new StringBuffer("spring");  
sb = sb.delete(3,6).insert(2,"umme").deleteCharAt(1);  
System.out.println("sb = " + sb);  
// o resultado é sb = summer
```

Sem chamada encadeada

```
StringBuffer sb = new StringBuffer("spring");  
sb = sb.delete(3,6);  
sb = sb.insert(2, "umme");  
sb = sb.deleteCharAt(1);  
System.out.println("sb = " + sb);  
// o resultado é sb = summer
```



Laboratório de Linguagens de Programação

Dicas de leitura

- Trabalhe da esquerda para a direita
- Encontre o resultado do método mais à esquerda
- Está confuso? Retire a chamada encadeada!



Laboratório de Linguagens de Programação

```
public class TestDog {  
  
    public static void main(String[] args) {  
        new TestDog().go();  
    }  
    void go(){  
        // nesse local estará o que REALMENTE queremos...  
    }  
  
}
```

Laboratório de Linguagens de Programação

Michel.java

```
1 public class Michel {
2     public static void main(String[] args) {
3         new Michel().nossa().assim().ai().ai().ln().
4         delicia().assim().ai().ai().ln().
5         nossa().assim().ai().ai().ln().
6         delicia().assim().ai().ai();
7     }
8     Michel nossa(){
9         System.out.println("Nossa, nossa");
10        return this;
11    }
12    Michel assim(){
13        System.out.println("Assim você me mata");
14        return this;
15    }
16    Michel ai(){
17        System.out.println("Ai, se eu te pego");
18        return this;
19    }
20    Michel ai().ai(){
21        System.out.println("Ai, ai se eu te pego");
22        return this;
23    }
24    Michel delicia(){
25        System.out.println("Delícia, delícia");
26        return this;
27    }
28    Michel ln(){
29        System.out.println("");
30        return this;
31    }
32 }
```

Problems @ Javadoc Declaration Console

<terminated> Michel (1) [Java Application] C:\Program Files\Ja

Nossa, nossa
Assim você me mata
Ai, se eu te pego
Ai, ai se eu te pego

Delícia, delícia
Assim você me mata
Ai, se eu te pego
Ai, ai se eu te pego

Nossa, nossa
Assim você me mata
Ai, se eu te pego
Ai, ai se eu te pego

Delícia, delícia
Assim você me mata
Ai, se eu te pego
Ai, ai se eu te pego