

Java Basics

Define the scope of variables

O escopo é o que determina em que pontos do código uma variável pode ser usada.

Variáveis locais

Chamamos de locais as variáveis declaradas dentro de métodos ou construtores. Antes de continuar, vamos estabelecer uma regra básica: o ciclo de vida de uma variável local vai do ponto onde ela foi declarada até o fim do bloco onde ela foi declarada.

Mas o que é um bloco? Podemos entender como bloco um trecho de código entre chaves. Pode ser um método, um construtor, o corpo de um **if**, de um **for** etc.:

```
1 public void m1() { // in cio do bloco do m todo
2     int x = 10; // vari vel local do m todo
3
4     if (x >= 10) { // in cio do bloco do if
5         int y = 50; // vari vel local do if
6         System.out.print(y);
7
8     } // fim do bloco do if
9
10 } // fim do bloco do m todo
```

Analisando esse código, temos uma variável **x**, que é declarada no começo do método. Ela pode ser utilizada durante todo o corpo do método. Dentro do **if**, declaramos a variável **y**. **y** só pode ser utilizada dentro do corpo do **if**, delimitado pelas chaves. Se tentarmos usar **y** fora do corpo do **if**, teremos um erro de compilação, pois a variável saiu do seu escopo.

Tome cuidado especial com loops **for**. As variáveis declaradas na área de inicialização do loop só podem ser usadas no corpo do loop:

```
1
2     for (int i = 0, j = 0; i < 10; i++)
3         j++;
4
5     System.out.println(j); // erro, j    n o  est  mais
                           no escopo
```

Define the structure of a Java class

Create executable Java applications with a main method; run a Java program from the command line; produce console output

Import other Java packages to make them accessible in your code

Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc.