

# Estrutura de Dados I

## Tipos de Dados em Java

Matheus Gabriel

Agosto 2024

## 1 Tipos de variáveis

### 1.1 Tipos primitivos

São os tipos criados pela própria linguagem de programação.

Cada tipo ocupa uma quantidade de bytes diferente:

Tipo de Dados	Tamanho (Bytes)	Descrição simplificada
byte	1	Número inteiro pequeno
short	2	Número inteiro médio
int	4	Número inteiro
long	8	Número inteiro grande
float	4	Número real de precisão simples
double	8	Número real de precisão dupla
char	2	Carácter Unicode
boolean	1 (ou 1 bit, em certos casos)	Valor lógico (true/false)

Ou use a tabela do W3Schools como referência.

### 1.2 Tipos estruturados (classes)

São criados a partir da composição de tipos primitivos. O maior exemplo são as próprias classes.

```
1 // Estudante.java
2 public class Estudante {
3     private String matricula;
4     private String nome;
5     private float n1;
6     private float n2;
7     private float sub;
8     private float pf;
9 }
```

Ao invés de armazenar as variáveis separadamente eles ficam juntas dentro da classe **Estudante**, o que ajuda na organização e manutenção do código.

#### Aviso

O conceito de visibilidade utilizado por **public** e **private** será explicado em anotações futuras, por enquanto eles podem ser ignorados.

## 2 Declaração de classes

O código a seguir declara os objetos `estudante` e `st`, é importante que o programa saiba aonde essa classe está armazenada, então por conveniência todos os arquivos `.java` desse exemplo estão no mesmo diretório.

```
1  // Main.java
2  public class Main {
3      public static void main(String[] args) {
4          Estudante estudante = new Estudante();
5          Estudante st;
6          st = new Estudante();
7      }
8  }
```

### 3 Getters e Setters

Os métodos *getters* e *setters* são usados para acessar um valor e definir um valor respectivamente. A norma para a nomenclatura desses tipos de métodos segue o estilo do exemplo a seguir:

```
1 // Estudante.java
2 public class Estudante {
3     private String matricula;
4     private String nome;
5     private float n1;
6     private float n2;
7     private float sub;
8     private float pf;
9
10    public String getMatricula() {
11        return matricula;
12    }
13
14    public void setMatricula(String matricula) {
15        this.matricula = matricula;
16    }
17
18 }
19
20 // Main.java
21 public class Main {
22     public static void main(String[] args) {
23         Estudante estudante = new Estudante();
24         Estudante st;
25         st = new Estudante();
26
27         estudante.setMatricula("PRIMEIRO");
28
29         System.out.println(estudante.getMatricula());
30     }
31 }
32
```

#### Aviso

Métodos são "praticamente" semelhantes às funções, mas elas estão sempre presentes dentro de uma classe.

### 3.1 Uso do this.

O `this`. é usado para alterar a variável da instância criada (`estudante` no caso do exemplo anterior), se usarmos `matricula` diretamente alteramos o valor do próprio parâmetro, o que pode causar erros em operações futuras com essa mesma classe.

### 3.2 Exemplo de setter modificado

O código interno dos getters e setters não precisam seguir o mesmo modelo, no exemplo a seguir, alteramos o setter para que o valor (da instância) nunca seja negativo.

```
1 // InteiroPositivo.java
2 public class InteiroPositivo {
3     private int valor;
4
5     public int getValor() {
6         return valor;
7     }
8
9     public void setValor(int valor) {
10         if (valor < 0) {
11             this.valor = -valor;
12         }
13         else {
14             this.valor = valor;
15         }
16     }
17 }
```