

Getters e Setters

Seguindo a convenção Java Beans:

Os métodos "Getters" e "Setters" são utilizados para buscar valores de atributos ou definir novos valores de atributos, de instâncias de classes.

O método **Getter**, retorna o valor do atributo especificado.

O método **Setter**, define outro novo valor para o atributo especificado.

Vemos o código abaixo, da criação de um objeto Aluno com nome e idade:

```
// arquivo Aluno.java
public class Aluno {
    String nome;
    int idade;
}

// arquivo Escola.java
public class Escola {
    public static void main(String[] args) {
        Aluno felipe = new Aluno();
        felipe.nome="Felipe";
        felipe.idade = 8;

        System.out.println("O aluno " + felipe.nome + " tem " + felipe.idade + " anos ");
        //RESULTADO NO CONSOLE
        //O aluno Felipe tem 8 anos
    }
}
```

Seguindo a convenção Java Beans, uma classe que contém esta estrutura de estados deverá seguir as regras abaixo:

- Os atributos precisam ter o modificador de acesso `private`. Ex.: `private String nome;`
- Como agora os atributos estarão somente a nível de classe, precisaremos dos métodos **getX** e **setX**, Ex.: `getNome()` e `setNome(String novoNome);`
- O método **get**, é responsável por obter o valor atual do atributo, logo ele precisa ser `public`, retornar um tipo correspondente ao valor, Ex.: `public String getNome() {};`
- O método **set**, é responsável por definir ou modificar o valor de um atributo em um objeto, logo, ele também precisa ser `public`, receber um parâmetro do mesmo tipo da variável, mas não retorna nenhum valor `void`. Ex.: `public void setNome(String newNome);`

```
//arquivo Aluno.java
public class Aluno {
    private String nome;
    private int idade;

    public String getNome() {
```

```
        return nome;
    }
    public void setName(String newName) {
        nome = newName;
    }
    public int getIdade() {
        return idade;
    }
    public void setIdade(int newIdade) {
        this.idade = newIdade;
    }
}
//arquivo Escola.java
public class Escola {
    public static void main(String[] args) {
        Aluno felipe = new Aluno();
        felipe.setName("Felipe");
        felipe.setIdade(8);

        System.out.println("O aluno " + felipe.getName() + " tem " + felipe.getIdade() + "
    }
}
```

- ❗ A proposta do código acima é a mesma que o código anterior, a diferença é que adotamos a convenção Java Beans, para definir e obter as características dos nossos objetos.

Uso do `this` no método `set`.

É muito comum vermos nossos métodos de definição ter a seguinte sintaxe:

```
//arquivo Aluno.java
private String nome;

public void setName(String nome) {
    this.nome = nome;
}
```

- ⚠ Observe que a descrição do nosso atributo `nome` é igual a descrição do parâmetro, logo, utilizamos mais uma palavra reservada `this`, para distinguir um do outro. Para mais detalhes veja [Palavras Reservadas](#).



Programação Orientada a Objetos - Previous
Visibilidade dos recursos

Next - Programação Orientada a Objetos
Construtores



Last modified 1mo ago