



# Modificadores de Acesso

## Encapsulamento

### Definição

Encapsular os dados de uma aplicação significa evitar que estes sofram acessos indevidos. Para isso, é criada uma estrutura que contém métodos que podem ser utilizados por qualquer outra classe, sem causar inconsistências no desenvolvimento de um código.

Na prática, isso é feito por meio de dois métodos: os *getters* e os *setters*.

- *Getters* têm por objetivo retornar o valor que lhe foi pedido, mas de forma a não prejudicar a integridade do dado em si.
- *Setters* recebem como argumento uma informação, que pode ser qualquer tipo de dados suportados pela linguagem. Dessa forma, não haverá o risco de ocorrerem acessos indevidos.

**public**, **private** e **protected**

Modificador	Uso Interno (Dentro da Classe)	Uso em Classes <u>Derivadas</u>	Uso Externo (Fora da Classe)
<b>public</b>	●	●	●
<b>protected</b>	●	●	●
<b>private</b>	●	●	●

### Exemplo em Java

Segue o exemplo abaixo:

```
public class Funcionario {  
    // Atributo Privado  
    private double salario;  
  
    public double getSalario() {  
        // Getter (Método Público)  
    }  
}
```

```
        return salario;
    }

    public void setSalario(double salario) {
        // Setter (Método Público)
        this.salario = salario;
    }
}
```

## Modificador Padrão ("*Default*")

Quando o acesso do modificador não é especificado, ele é tratado como um modificador padrão. Seu escopo é limitado ao *package*.