

Encapsulamento

Grupo 3

Patrick, Leonardo, João, Gabriel, Eduarda, Igor, Pablo

O que é?

Encapsular seria o mesmo que esconder todos os membros de uma classe além de esconder como funcionam as rotinas (no caso métodos) do nosso sistema. Seria uma espécie de proteção.

Para protegemos os atributos métodos, portanto, podemos mudar seus modificadores. O principal deles é o modificador `private` que faz com que ninguém consiga modificar, nem mesmo ler, o atributo em questão

```
— class Funcionario {  
    private double salario  
}
```

Para que serve?

Uma das ideias mais importantes da orientação a objetos é o encapsulamento. Encapsular significa esconder a implementação dos objetos. O encapsulamento favorece principalmente dois aspectos de um sistema: a manutenção e o desenvolvimento

A **manutenção** é favorecida pois uma vez aplicado o encapsulamento, quando o funcionamento de um objeto deve ser alterado, em geral, basta modificar a classe do mesmo.

O **desenvolvimento** é favorecido pois, uma vez aplicado o encapsulamento, conseguimos determinar precisamente as responsabilidades de cada classe da aplicação.

Como funciona?

|O encapsulamento funciona com a utilização de modificadores de acesso para restringir o acesso aos atributos e aos métodos de um objeto. Garante que somente as classes apropriadas possam acessar as informações e ajuda a assegurar que os atributos e os métodos sejam usados de forma consistente e previsível.

Modificadores de acesso

private — os atributos e os métodos marcados como “private” só podem ser acessados dentro da própria classe em que foram declarados;

```
— public class Funcionario {  
    private String nome;  
}
```

protected — os atributos e os métodos marcados como “protected” podem ser acessados dentro da própria classe e das subclasses;

```
— public class Funcionario {  
    protect String cpf;  
}
```

public — os atributos e os métodos marcados como “public” podem ser acessados por qualquer classe.

— `public class Funcionario {`
`}`

Exemplos de Getters and Setters

```
public class Vehicle {  
    private String color;
```

```
        public String getColor(){  
            return color;  
        }
```

```
        public void setColor(String c) {  
            —    this.color = c;  
        }  
    }
```

Exemplos encapsulamento em atributos


```
public class Conta {
```

```
    private double saldo;
```

```
    void deposita(double valor) {  
        this.saldo += valor;  
        this.descontaTarifa();  
    }
```

```
    void saca(double valor) {  
        this.saldo -= valor;  
        this.descontaTarifa();  
    }
```

```
    private void descontaTarifa() {  
        this.saldo -= 0.1;  
    }
```

```
}
```