

JAVA



ENCAPSULAMENTO

HORA DA ANALISE



```
Main.java x Conta.java x
1 package com.company;
2
3 public class Conta {
4
5     int numero;
6     String dono;
7     double saldo;
8
9     public boolean sacar(int valor) {
10         if (valor <= this.saldo) {
11             this.saldo = saldo - valor;
12             return true;
13         }
14         return false;
15     }
16 }
```

```
Main.java x Conta.java x
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         Conta conta = new Conta();
7         conta.dono = "Taina";
8         conta.numero = 123456789;
9         conta.saldo = -50;
10
11         System.out.println(conta.sacar(valor: 10));
12         System.out.println(conta.saldo);
13     }
14 }
```

```
Run: Main x
▶ ↑ /Library/
■ ↓ false
📷 ↺ -50.0
⚙️ ⬇
```

QUESTIONAMENTO?

Se no meu requisito do sistema, eu tenho alguma restrição, por exemplo.

- × O saldo da conta só poderá ser positivo.

```
Conta conta = new Conta();  
conta.dono = "Taina";  
conta.numero = 123456789;  
conta.saldo = -50;
```

ENCAPSULAMENTO

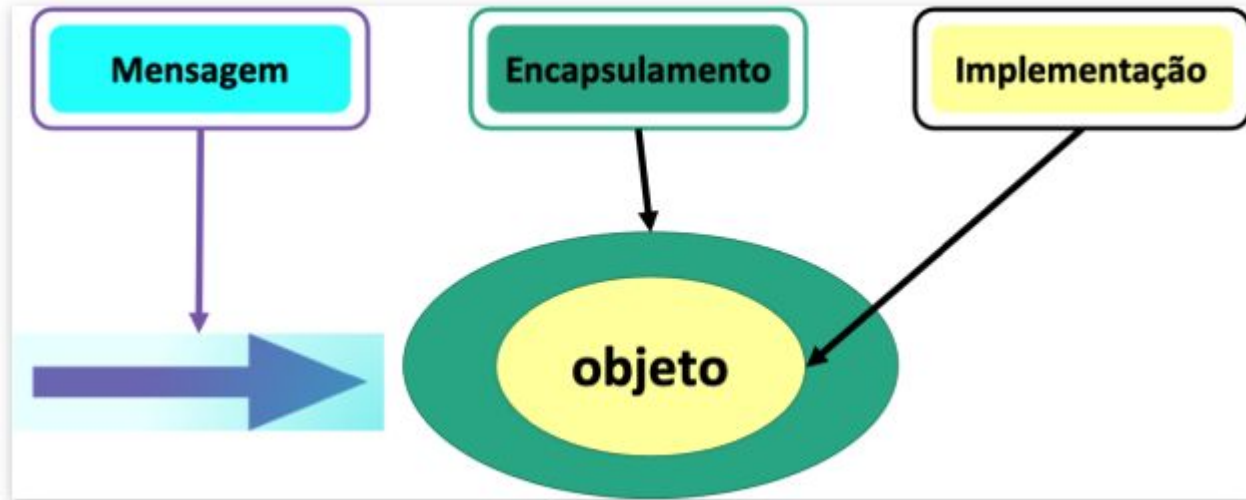


ENCAPSULAMENTO

Encapsulamento é a característica da OO capaz de ocultar partes (dados e detalhes), de implementação interna de classes, do mundo exterior.

- × Acesso deve-se dar a quem e por quem?
- × Acesso a que?

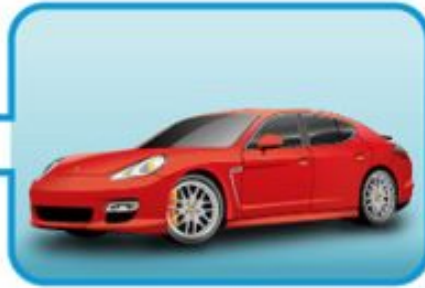
ENCAPSULAMENTO



HAM?



ENTENDENDO MELHOR...



***COMO ISSO
ACONTECE NA
PRÁTICA?***



MODIFICADORES DE ACESSO

Public: Garante que o atributo ou método da classe seja acessado ou executado a partir de qualquer outra classe;

Private: Pode ser acessado, modificado ou executado apenas por métodos da mesma classe;

MODIFICADORES DE ACESSO

Readonly: Atributo somente leitura, inicializados em sua declaração ou no construtor;

Protected: Funciona como o private, exceto que as classes filhas ou derivadas também terão acesso ao atributo ou método (Herança).



Main.java



Conta.java



1

```
package com.company;
```

2

3

```
public class Conta {
```

4

5

```
    private int numero;
```

6

```
    private String dono;
```

7

```
    private double saldo;
```

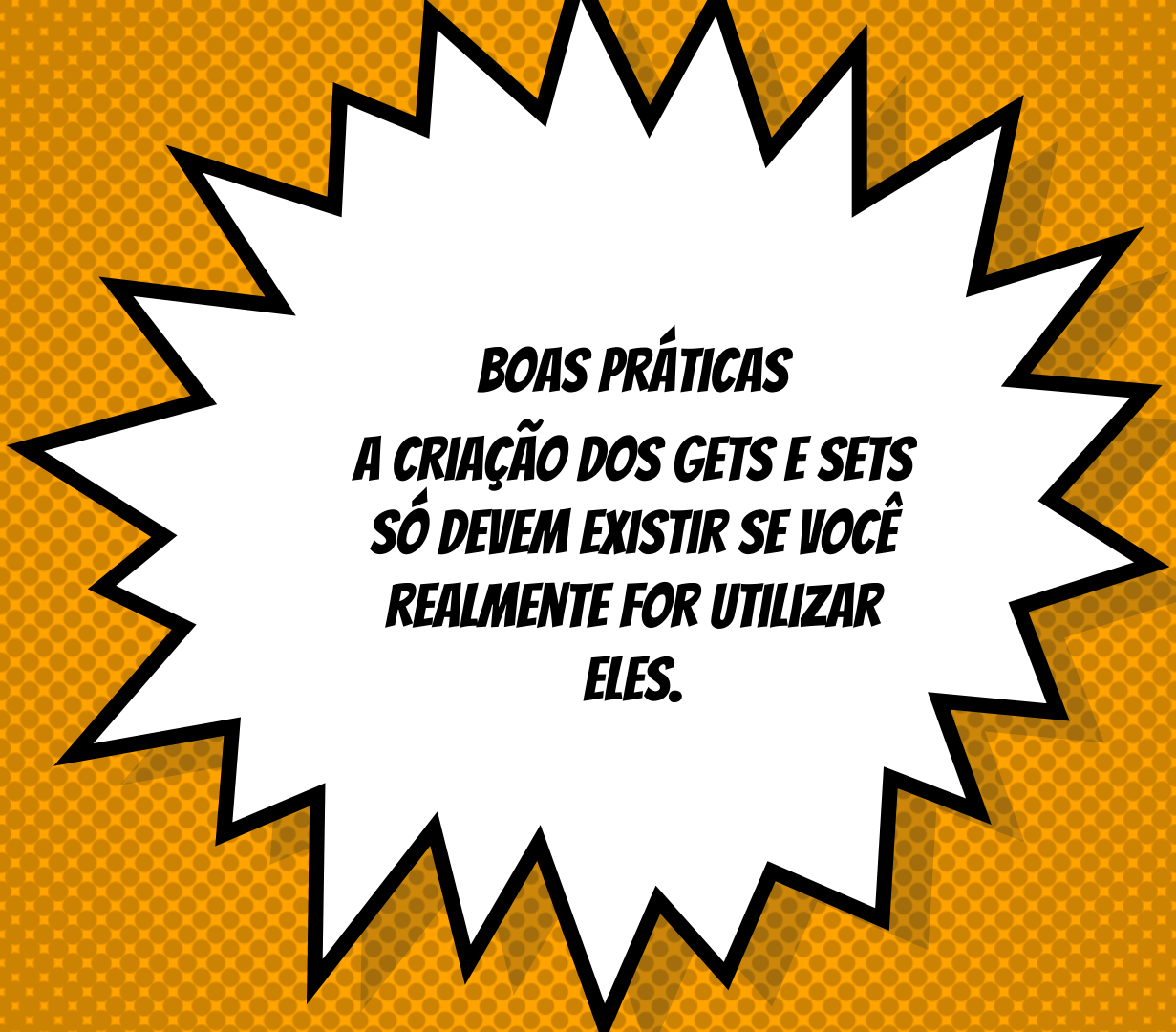
8

***O QUE PODERÍAMOS
FAZER, COM O QUE
VIMOS ATÉ AGORA PARA
SOLUCIONAR ESSA
BRONCA?***




```
Main.java x Conta.java x
9      public int getNumero() {
10          return numero;
11      }
12
13      public void setNumero(int numero) {
14          this.numero = numero;
15      }
16
17      public String getDono() {
18          return dono;
19      }
20
21      public void setDono(String dono) {
22          this.dono = dono;
23      }
24
25      public double getSaldo() {
26          return saldo;
27      }
28
29      public void setSaldo(double saldo) {
30          this.saldo = saldo;
31      }
```

```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         Conta conta = new Conta();
7         conta.setDono("Taina");
8         conta.setNumero(123456789);
9         conta.setSaldo(-50);
10
11         System.out.println(conta.sacar( valor: 10));
12         System.out.println(conta.getSaldo());
13     }
14 }
```



***BOAS PRÁTICAS
A CRIAÇÃO DOS GETS E SETS
SÓ DEVEM EXISTIR SE VOCÊ
REALMENTE FOR UTILIZAR
ELES.***

```
Main.java x Conta.java x
9      public int getNumero() {
10          return numero;
11      }
12
13      public void setNumero(int numero) {
14          this.numero = numero;
15      }
16
17      public String getDono() {
18          return dono;
19      }
20
21      public void setDono(String dono) {
22          this.dono = dono;
23      }
24
25      public double getSaldo() {
26          return saldo;
27      }
28
29      public void setSaldo(double saldo) {
30          this.saldo = saldo;
31      }
```

```
Main.java x Conta.java x
9      public void setNumero(int numero) {
10          this.numero = numero;
11      }
12
13      public void setDono(String dono) {
14          this.dono = dono;
15      }
16
17      public double getSaldo() {
18          return saldo;
19      }
20
21      public void setSaldo(double saldo) {
22          this.saldo = saldo;
23      }
```

***EITA, E COMO FICA
AGORA O
RELACIONAMENTO ENTRE
OBJETOS?***



RELACIONAMENTO ENTRE OBJETOS

Então, se quisermos representar um cliente em nosso sistema, com as seguintes informações:

- × Nome
- × CPF
- × Endereço

E para fazer
com que o
cliente criado
possua uma
conta?



```
1 package com.company;  
2  
3 public class Pessoa {  
4     private String nome;  
5     private String cpf;  
6     private String endereco;  
7 }
```

```
1 package com.company;
2
3 public class Pessoa {
4     private String nome;
5     private String cpf;
6     private String endereco;
7
8     private Conta conta = null;
9
10    public void adicionarContar(Conta conta) {
11        this.conta = conta;
12    }
13 }
```

```
3 public class Pessoa {
4     private String nome;
5     private String cpf;
6     private String endereco;
7     private Conta conta = null;
8
9     public String getNome() {
10         return nome;
11     }
12
13     public void setNome(String nome) {
14         this.nome = nome;
15     }
16
17     public String getCpf() {
18         return cpf;
19     }
20
21     public void setCpf(String cpf) {
22         this.cpf = cpf;
23     }
24
25     public String getEndereco() {
26         return endereco;
27     }
28
29     public void setEndereco(String endereco) {
30         this.endereco = endereco;
31     }
32
33     public Conta getConta() {
34         return conta;
35     }
36
37     public void adicionarConta(Conta conta) {
38         this.conta = conta;
39     }
40 }
```

```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         //Criando Cliente
7         Pessoa cliente = new Pessoa();
8         cliente.setNome("Taina");
9
10        //Criando conta
11        Conta conta = new Conta();
12
13        //Setando valores
14        cliente.adicionarContar(conta);
15        cliente.getConta().setNumero(123456789);
16        cliente.getConta().setSaldo(100);
17        cliente.setNome("Taina");
18        cliente.setCpf("123.456.789-00");
19        cliente.setEndereco("Rua bla bla, n 123, bairro tal");
20
21        //Realizando um saque
22        cliente.getConta().sacar( valor: 10);
23
24        //Imprimindo os valores
25        System.out.println("O saldo da conta do cliente é: "+ cliente.getConta().getSaldo());
26        System.out.printf("Cliente %s de cpf %s mora no endereço %s", cliente.getNome(), cliente.getCpf(), cliente.getEndereco());
27    }
28 }
```

Run: Main x



```
↑ /Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home/bin/java -javaagent
↓ 0 saldo da conta do cliente é: 90.0
⇌ Cliente Taina de cpf 123.456.789-00 mora no endereço Rua bla bla, n 123, bairro tal
⌕ Process finished with exit code 0
```

```
21 //Realizando um saque
```

```
22 boolean podeSacar = cliente.getConta().sacar( valor: 10);
```

```
23  
24 //Imprimindo os valores
```

```
25 if (podeSacar == true) {
```

```
26     System.out.println("O saldo da conta do cliente é: "+ cliente.getConta().getSaldo());
```

```
27     System.out.printf("Cliente %s de cpf %s mora no endereço %s", cliente.getNome(), cliente.getCpf(), cliente.getEndereco());
```

```
28 } else {
```

```
29     System.out.println("Impossível realizar o saque!!!");
```

```
30 }
```

```
31  
32 }
```



***MÃO NA MASSA
COM VOCÊ***

PRÁTICA 01

Na main, crie três objetos Pessoa e para cada uma crie uma Conta.

Faça operações de sacar e transferir dinheiro entre essas contas.

No final, mostre as informações de cada cliente e sua respectiva conta.

PRÁTICA 02

Crie duas classes, uma chamada professor e outra aluna. Com os seguintes atributos:

- × Professor: Nome, curso, cpf, salario, lista de alunos.
- × Aluna: Nome, nota1, nota2.

Imprima na Main o nome do professor e seu curso.

PRÁTICA 03

Na classe Professor faça um método para calcular o salário do professor. Para esse cálculo é necessário saber o valor da hora aula do professor e o número de aulas ministradas).

Salario = Valor hora aula * aula ministradas

PRÁTICA 03 (CONTINUAÇÃO)...

O valor da aula é um valor fixo de R\$29,00, faça com que a pessoa usuária entrar com o valor de número de aulas ministradas.

Dica: Utilize um método "setSalarioProfessor", este método será chamado na Main.

Imprima na Main o salário do professor, lembrando que o atributo salário é private.

PRÁTICA 04

Agora, crie um método de calcularMedia na classe Aluna. Os valores da nota1 e nota2 serão informados pela pessoa usuária, juntamente com o nome da aluna.

Na main, imprima o nome da aluna com a média final, e uma atribuição informando se o aluno foi "Aprovado" ou "Reprovado"

(media \geq 7, a aluna está aprovada)



THE END