

Orientação a Objetos

Prof. Me. Fabiano Fernandes

Conteúdo da aula

- Conteúdo da aula
 - Definição de OOP
 - Pilares da OOP
 - Abstração
 - Herança
 - Polimorfismo
 - Encapsulamento
 - Métodos e Atributos
 - Classes

Definição de OOP

“Uma nova maneira de pensar os problemas utilizando conceitos do Mundo Real.”

Consiste em utilizar estruturas de dados que simulem o comportamento da vida real.

Pilares de OOP

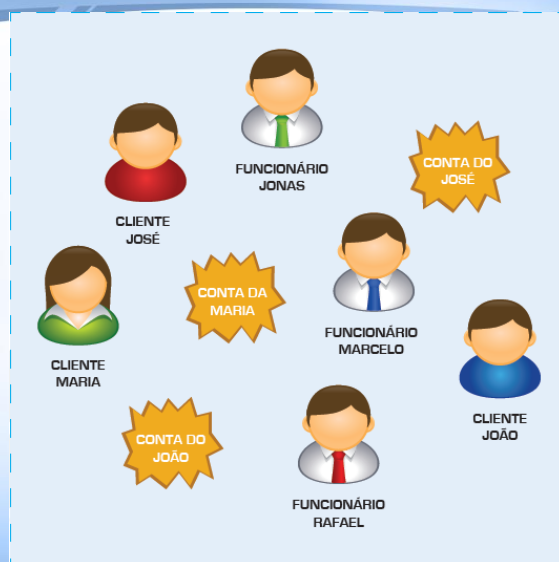


Abstração

Visa abstrair algo do mundo real e transformá-lo em um objeto na programação.

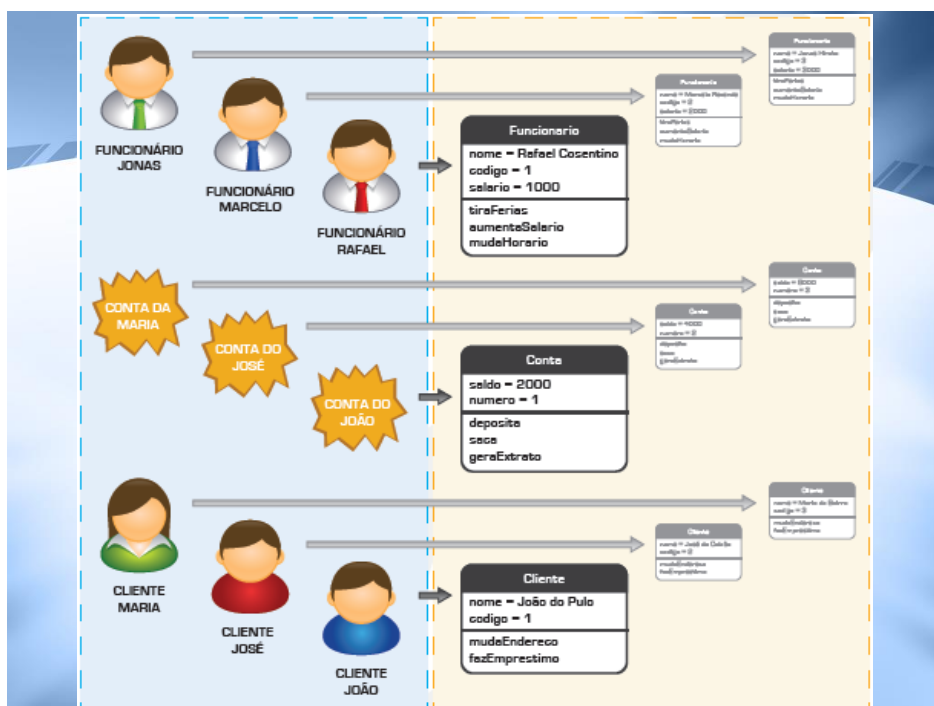
Habilidade de se concentrar nos aspectos essenciais do sistema, ou um contexto qualquer, ignorando o que é supérfluo.

Princípios da POO

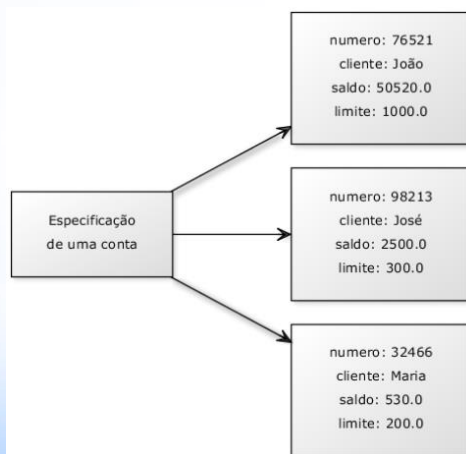


Objetos, Atributos e Métodos

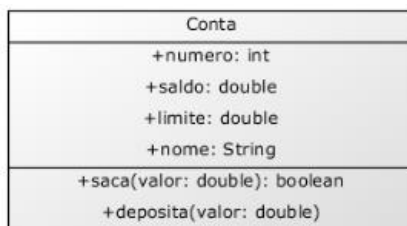
- Objeto
- Atributos
- Métodos



Objeto...



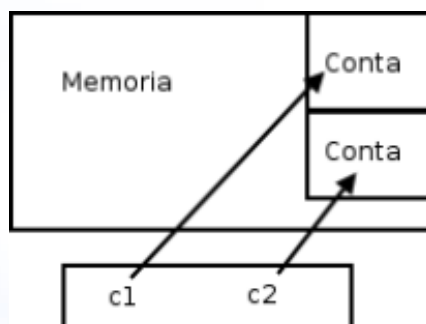
Implemente a classe abaixo...



Referência a objetos...

```
public static void main(String args[]) {  
    Conta c1;  
    c1 = new Conta();  
    Conta c2;  
    c2 = new Conta();  
}
```

Referência a objetos...



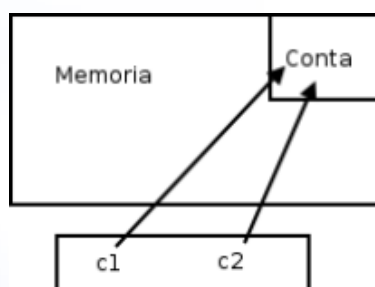
Referência a objetos...

```
Conta c1 = new Conta();  
c1.deposita(100);
```

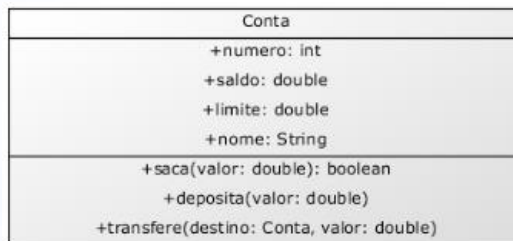
```
Conta c2 = c1;  
c2.deposita(200);
```

```
System.out.println(c1.saldo);  
System.out.println(c2.saldo);
```

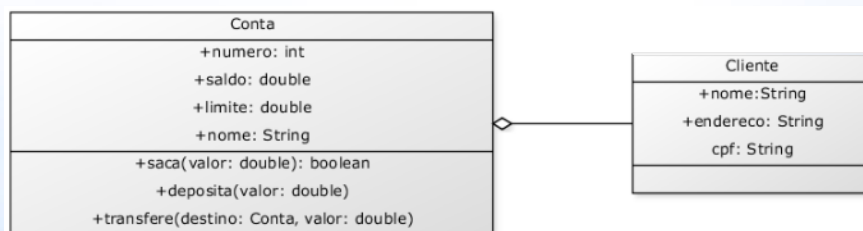
Referência a objetos...



Implemente a classe abaixo...



Implemente as classes abaixo...



Exercício

1) Modele um funcionário. Ele deve ter o nome do funcionário, o departamento onde trabalha, seu salário (double), a data de entrada no banco (String) e seu RG (String).

Você deve criar alguns métodos de acordo com sua necessidade.

Além deles, crie um método `recebeAumento` que aumenta o salário do funcionário de acordo com o parâmetro passado como argumento.

Crie também um método `calculaGanhoAnual`, que não recebe parâmetro algum, devolvendo o valor do salário multiplicado por 12.

A ideia aqui é apenas modelar, isto é, só identifique que informações são importantes e o que um funcionário faz. Desenhe no papel tudo o que um `Funcionario` tem e tudo que ele faz.

Exercício

2) Transforme o modelo do exercício anterior em uma classe Java.

Teste-a, usando uma outra classe que tenha o `main`. Você deve criar a classe do funcionário com o nome `Funcionario`, mas pode nomear como quiser a classe de testes, contudo, ela deve possuir o método `main`.

Lembre-se da dica: Você pode (e deve) compilar seu arquivo java sem que você ainda tenha terminado sua classe `Funcionario`. Isso evitará que você receba dezenas de erros de compilação de uma vez só.

Exercício

- Programa 1

Classe: Pessoa

Atributos: nome, idade.

Método: void fazAniversario()

Crie uma pessoa, coloque seu nome e idade iniciais, faça alguns aniversários (aumentando a idade) e imprima seu nome e sua idade.

Exercício

- Programa 2

Classe: Porta

Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ

Métodos: void abre()

void fecha()

void pinta(String s)

boolean estaAberta()

Crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método estaAberta para verificar se ela está aberta.

Exercício

- Classe: Casa

Atributos: cor, porta1, porta2, porta3

Método: void pinta(String s),
int quantasPortasEstaoAbertas()

Crie uma casa e pinte-a. Crie três portas e coloque-as na casa; abra e feche as mesmas como desejar. Utilize o método *quantasPortasEstaoAbertas* para imprimir o número de portas abertas.