



FIESC SENAI

A FORÇA DA INDÚSTRIA CATARINENSE

Orientação a objetos básica

Rodrigo Fortes

Veremos

- O que é e para que serve orientação a objetos;
- Classes, atributos e comportamentos;
- Variáveis e objetos na memória.

O que toda conta bancária tem e é importante?

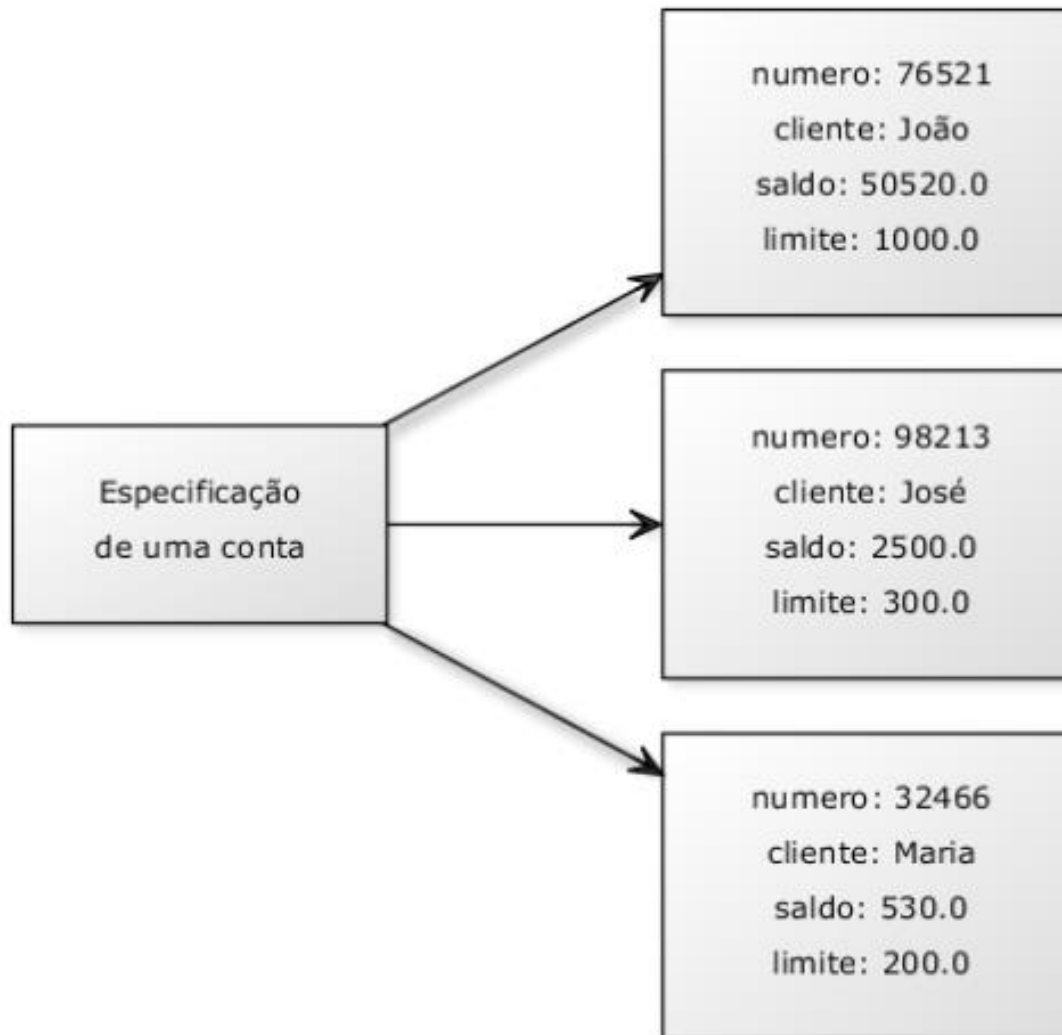
- número da conta
- nome do dono da conta
- saldo
- limite

Operações com Inteiros

O que toda conta bancária faz e é importante?

- saca uma quantidade x
- deposita uma quantidade x
- imprime o nome do dono da conta
- devolve o saldo atual
- transfere uma quantidade x para uma outra conta y
- devolve o tipo de conta

Declaração de Variáveis



```
class Conta {  
    int numero;  
    String dono;  
    double saldo;  
    double limite;  
  
    // ..  
}
```

Objeto

```
class Programa {  
    public static void main(String[] args) {  
        new Conta();  
    }  
}
```


Objeto

```
class Programa {  
    public static void main(String[] args) {  
        Conta minhaConta;  
        minhaConta = new Conta();  
    }  
}
```

Objeto

```
class Programa {  
    public static void main(String[] args) {  
        Conta minhaConta;  
        minhaConta = new Conta();  
  
        minhaConta.dono = "Rodrigo";  
        minhaConta.saldo = 1000.0;  
  
        System.out.println("Saldo atual: " + minhaConta.saldo);  
    }  
}
```

Métodos sem Retorno

```
class Conta {  
    double salario;  
    // ... outros atributos ...  
  
    void saca(double quantidade) {  
        double novoSaldo = this.saldo - quantidade;  
        this.saldo = novoSaldo;  
    }  
}
```

Métodos sem Retorno

```
class Conta {  
    // ... outros atributos e métodos ...  
  
    void deposita(double quantidade) {  
        this.saldo += quantidade;  
    }  
}
```

Métodos sem Retorno

```
class TestaAlgunsMetodos {  
    public static void main(String[] args) {  
        // criando a conta  
        Conta minhaConta;  
        minhaConta = new Conta();  
  
        // alterando os valores de minhaConta  
        minhaConta.dono = "Rodrigo";  
        minhaConta.saldo = 1000;  
  
        // saca 200 reais  
        minhaConta.saca(200);  
  
        // deposita 500 reais  
        minhaConta.deposita(500);  
        System.out.println(minhaConta.saldo);  
    }  
}
```

Qual o retorno?

Métodos com Retorno

```
class Conta {  
    // ... outros métodos e atributos ...  
  
    boolean saca(double valor) {  
        if (this.saldo < valor) {  
            return false;  
        }  
        else {  
            this.saldo = this.saldo - valor;  
            return true;  
        }  
    }  
}
```

Métodos com Retorno

Conta
+numero: int
+saldo: double
+limite: double
+nome: String
+saca(valor: double): boolean
+deposita(valor: double)

Métodos com Retorno

```
minhaConta.saldo = 1000;  
boolean consegui = minhaConta.saca(2000);  
if (consegui) {  
    System.out.println("Consegui sacar");  
} else {  
    System.out.println("Não consegui sacar");  
}
```


Objetos Acessados por Referência

Quando declaramos uma variável para associar a um objeto, na verdade, essa variável não guarda o objeto, e sim uma maneira de acessá-lo, chamada de **referência**.

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

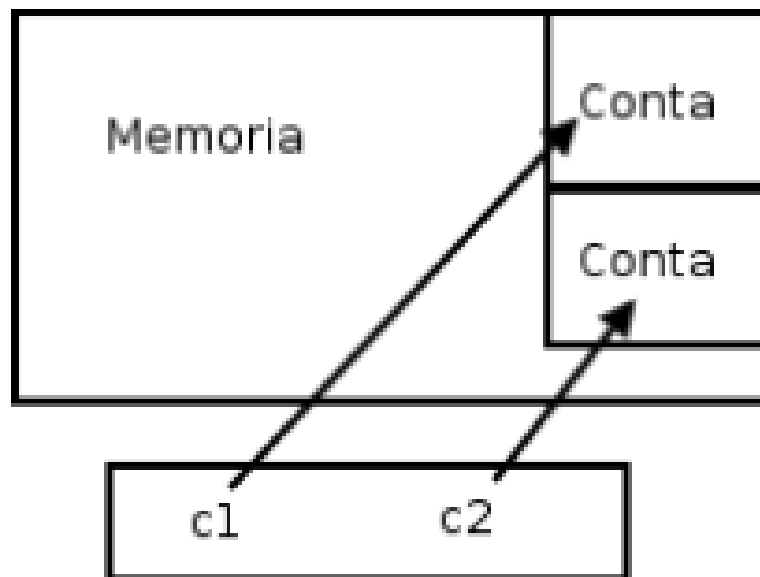
Objetos Acessados por Referência

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

c1 se refere a um objeto. **Não é correto** dizer que c1 é um objeto, pois c1 é uma variável de referência, apesar de, depois de um tempo, os programadores

Geralmente falamos "Tenho um **objeto c** do tipo **Conta**", mas apenas para não precisar falar "Tenho uma **referência c** a um **objeto** do tipo **Conta**".

Objetos Acessados por Referência



Objetos Acessados por Referência

```
class TestaReferencias {  
    public static void main(String args[]) {  
        Conta c1 = new Conta();  
        c1.deposita(100);  
  
        Conta c2 = c1; // linha importante!  
        c2.deposita(200);  
  
        System.out.println(c1.saldo);  
        System.out.println(c2.saldo);  
    }  
}
```

Método Transfere

```
boolean transfere(Conta destino, double valor) {  
    boolean retirou = this.saca(valor);  
    if (retirou == false) {  
        // não deu pra sacar!  
        return false;  
    }  
    else {  
        destino.deposita(valor);  
        return true;  
    }  
}
```

Método Transfere

Conta
+numero: int
+saldo: double
+limite: double
+nome: String
+saca(valor: double): boolean
+deposita(valor: double)
+transfere(destino: Conta, valor: double): boolean

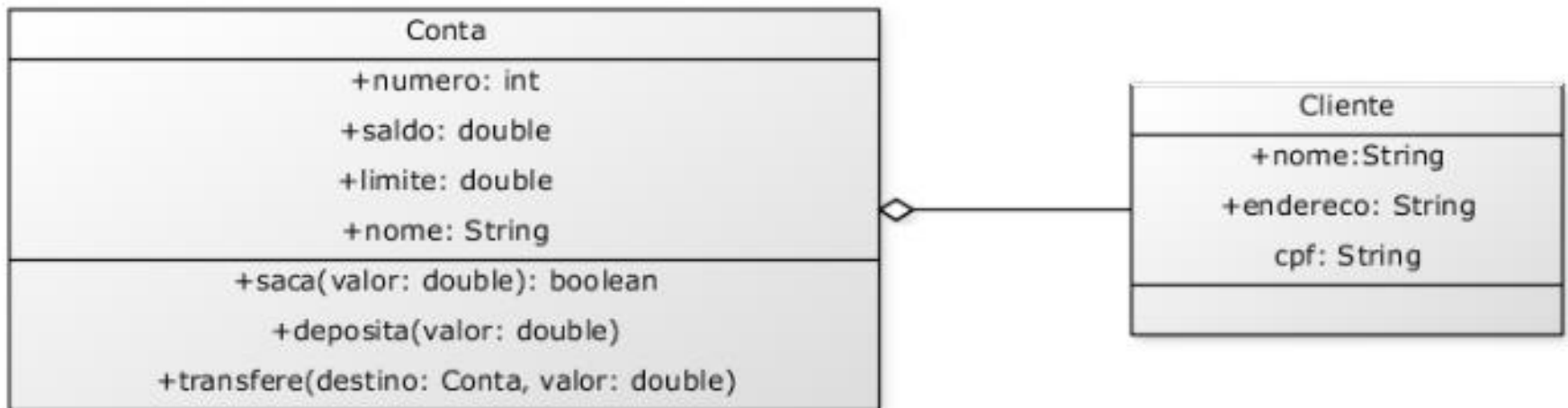
Atributos com Referência

```
class Cliente {  
    String nome;  
    String sobrenome;  
    String cpf;  
}
```

```
class Conta {  
    int numero;  
    double saldo;  
    double limite;  
    Cliente titular;  
    // ..  
}
```

```
class Teste {  
    public static void main(String[] args) {  
        Conta minhaConta = new Conta();  
  
        minhaConta.titular.nome = "Rodrigo";  
        // ...  
    }  
}
```


Atributos com Referência



Perguntas?