

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Programação Orientada a Objetos

Ciências da Computação

Abstração e
Modelagem

O que é a Orientação a Objetos ? ...

Classe → **tipo** ou **molde**



Objeto → **instância** ou **exemplar**

Mensagem → **comunicação** entre objetos

vocabulário básico

Abstração

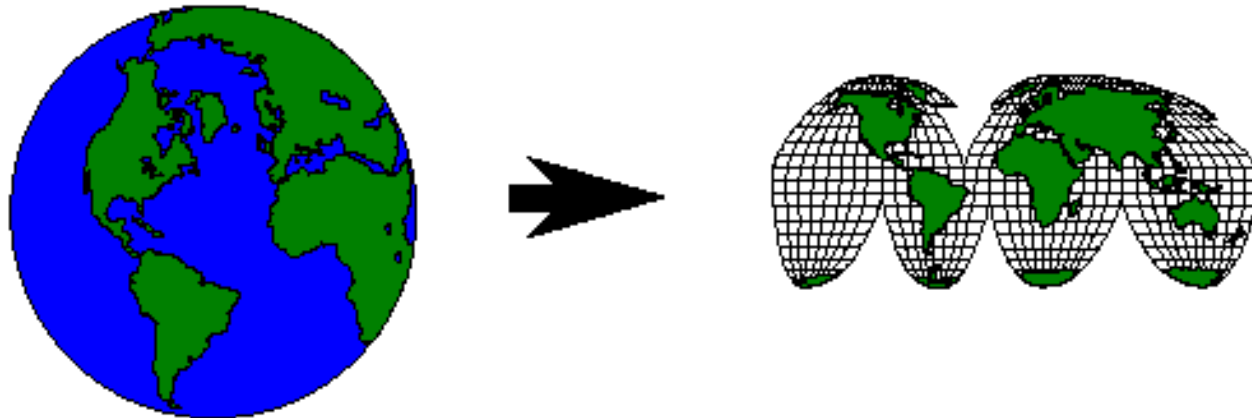
Habilidade de se concentrar nos aspectos essenciais de um Contexto, ignorando características menos importantes

Na OO, uma classe é uma abstração de entidades existentes do sistema de software

Tecnologia de Orientação a Objetos

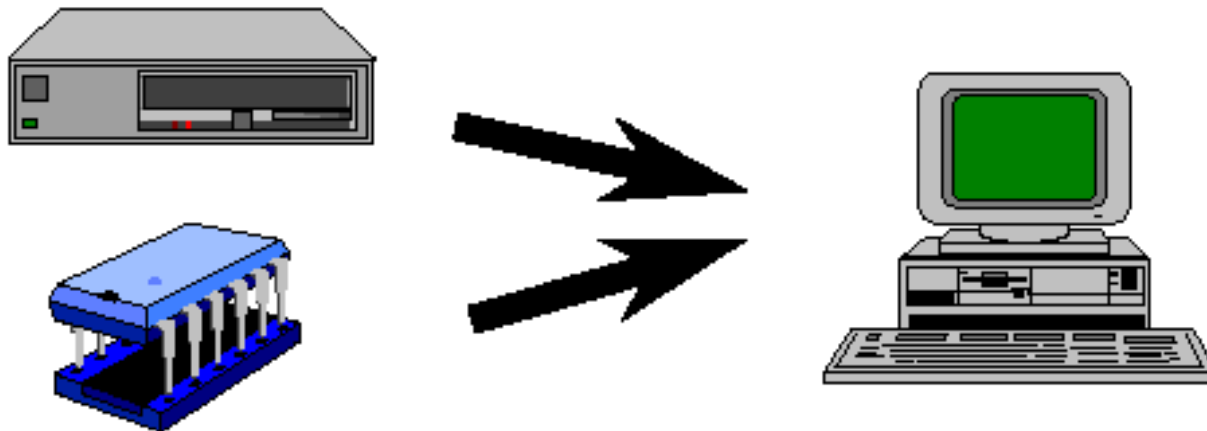
Abstração

- A OO facilita a abstração por representar mais intuitivamente objetos do mundo real.



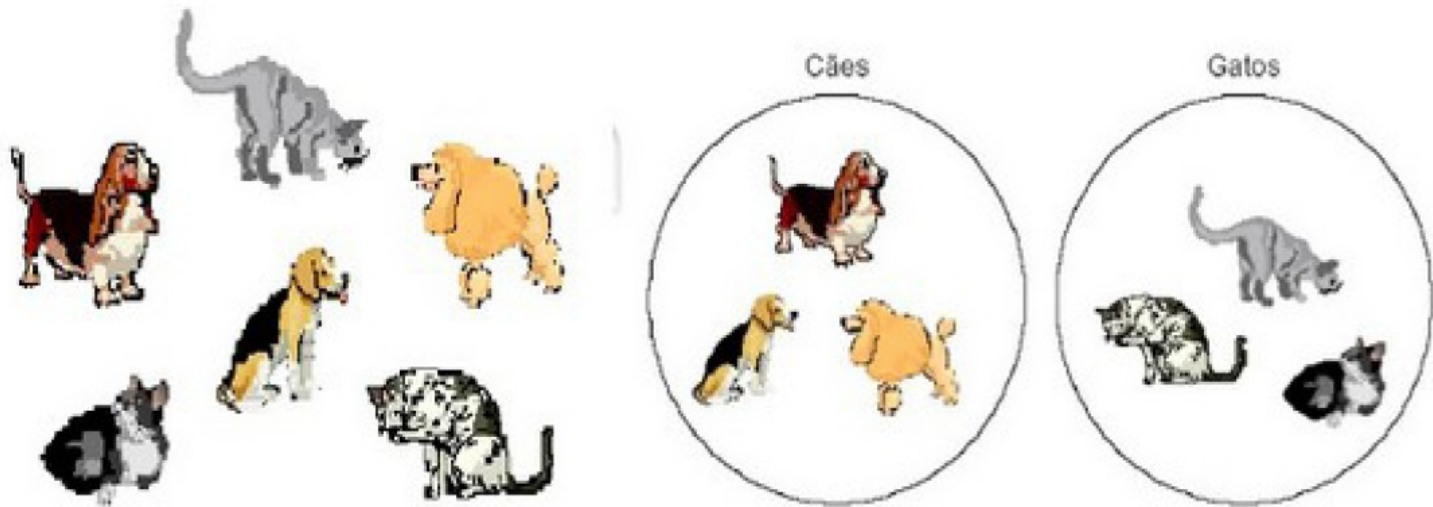
Tecnologia de Orientação a Objetos

- O programador trabalha sobre componentes já desenvolvidos e testados, diminuindo a complexidade e aumentando a abstração.

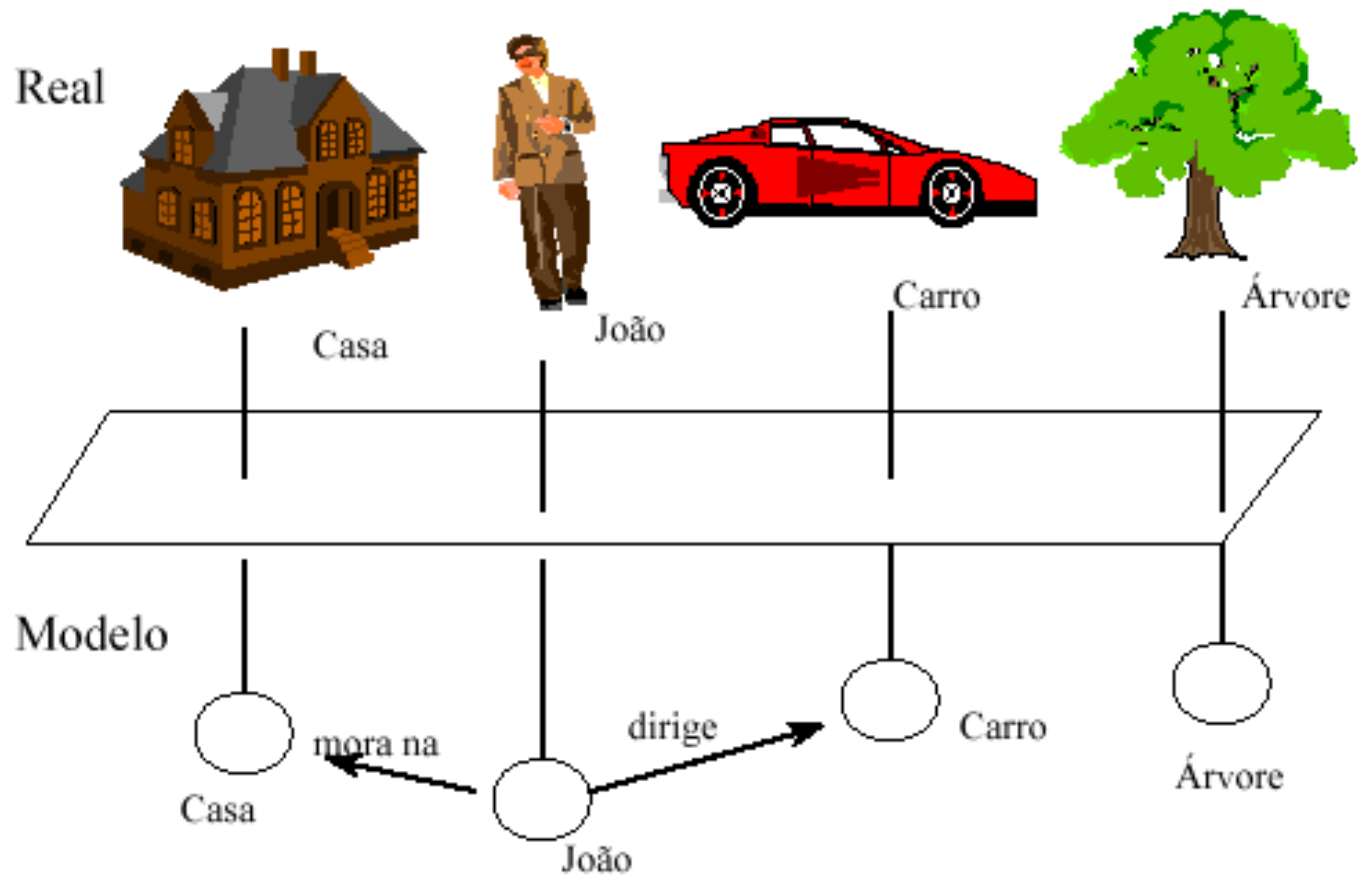


Orientação a Objetos

Classificando

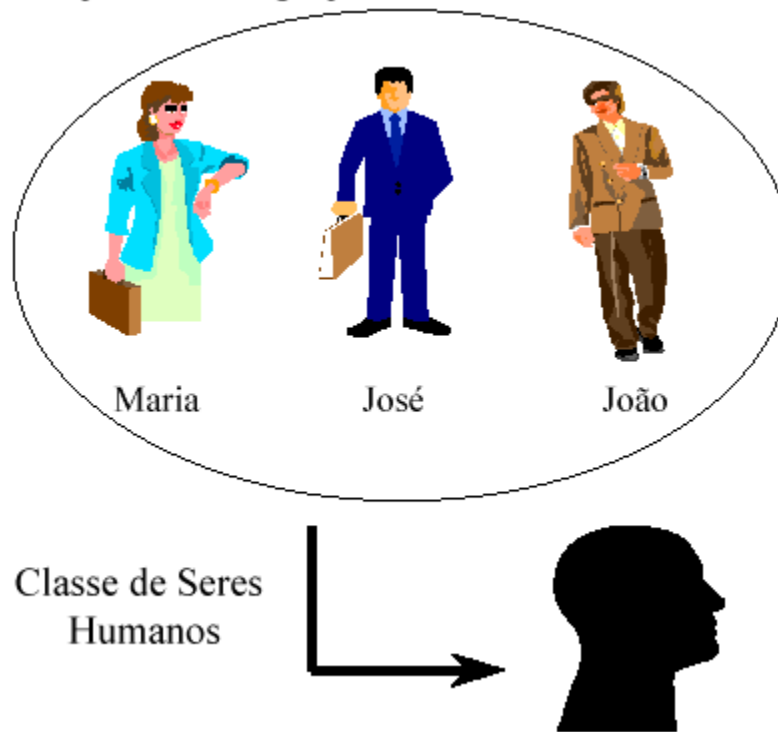


Orientação a Objetos



Orientação a Objetos

Classe: representa um “gabarito” para muitos objetos e descreve como estes objetos estão estruturados internamente.



Orientação a Objetos

Objetos: “coisas” do mundo real



Uma **instância** é um **objeto** criado a partir de uma classe.
A **classe** descreve a estrutura da instância, enquanto que o estado da **instância** é definido pelas **operações** realizadas sobre ela.

Orientação a Objetos

Propriedade e Atributos: objetos do mundo real possuem propriedades e valores para estas propriedades

	<u>Maria</u>	<u>José</u>	<u>João</u>
• idade	31	28	43
• endereço	Rua xx	Rua yy	Av zz
• sexo	Fem	Masc	Masc
• etc			



Maria



José



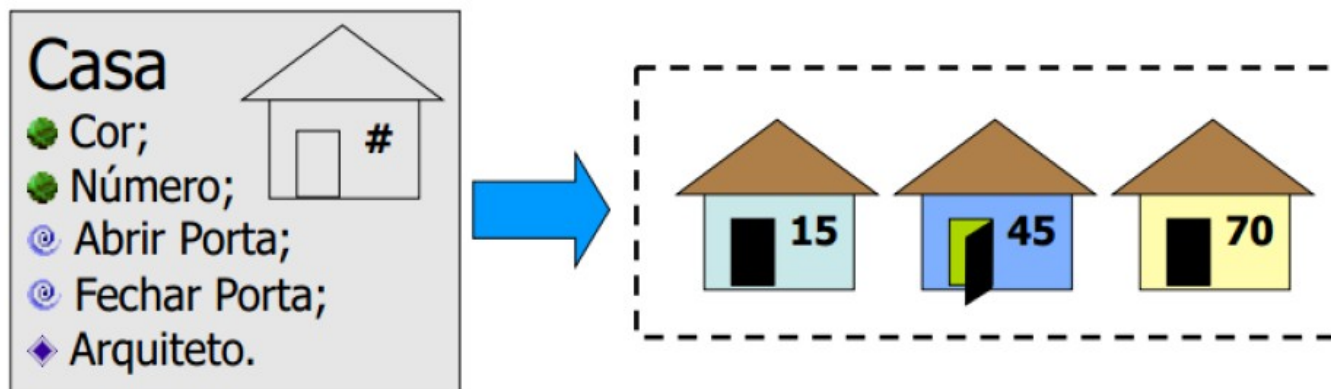
João

Orientação a Objetos

Diferenciando Classes de Objetos

Classe

Objetos



Orientação a Objetos

Diferenciando Classes de Objetos

Classe Carro		Objeto Carro A	Objeto Carro B
<i>Atributos de Objeto</i>	Número da placa	ABC 111	XYZ 123
	Cor	Azul	Vermelha
	Fabricante	Mitsubishi	Toyota
	Velocidade	50 km/h	100 km/h
<i>Métodos de Objeto</i>	Método Acelerar		
	Método Girar		
	Método Frear		

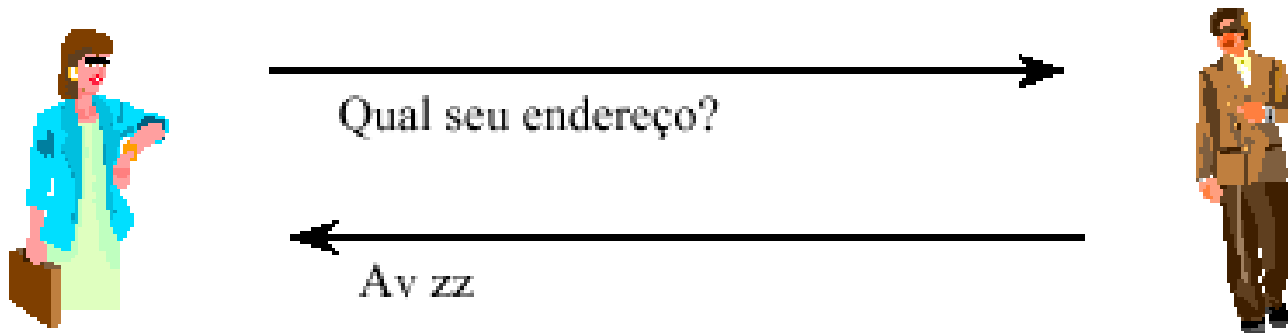
Classe Carro e seus objetos

Cada objeto tem seu conjunto de atributos – representam o estado do objeto;

A implementação dos métodos é compartilhada por todos os objetos da classe;

Orientação a Objetos

Mensagens e Métodos: um objeto exibe algum comportamento (executa alguma operação) quando recebe um estímulo de outro objeto



Um objeto requisita a ação de algum outro objeto enviando uma Mensagem para ele.

Orientação a Objetos

Uma **mensagem** contém:

- o nome do objeto receptor;
- o nome da mensagem;
- argumentos (opcional) que podem ser objetos;

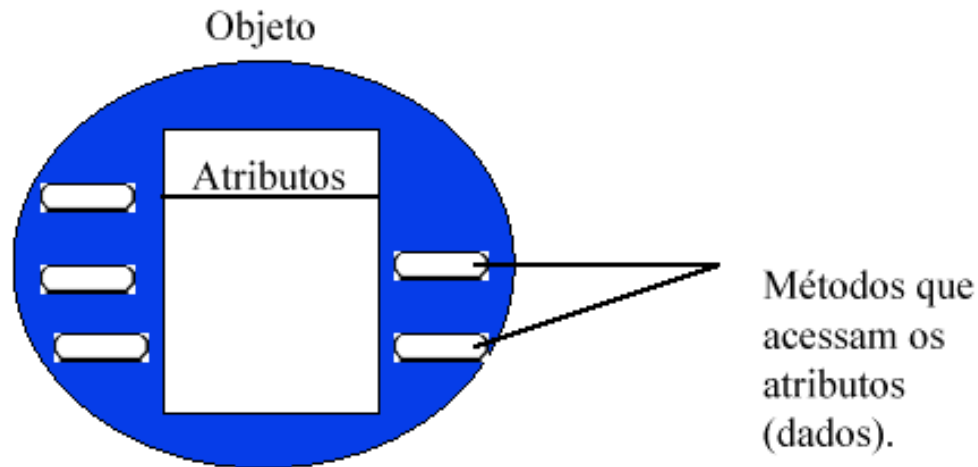
Uma **mensagem** é uma solicitação a um objeto para que seja executada uma rotina denominada **método**

Os **métodos** são responsáveis por acessar ou alterar os **atributos** de um objeto

Orientação a Objetos

Encapsulamento:

termo formal que define o empacotamento de dados de um objeto, permitindo o acesso aos dados somente através dos métodos deste mesmo objeto.

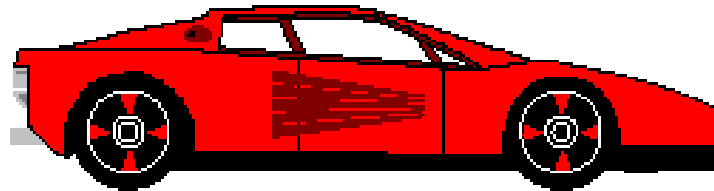


Orientação a Objetos

Com o **encapsulamento**, o objeto se comporta como uma caixa-preta, aumentando a **abstração**

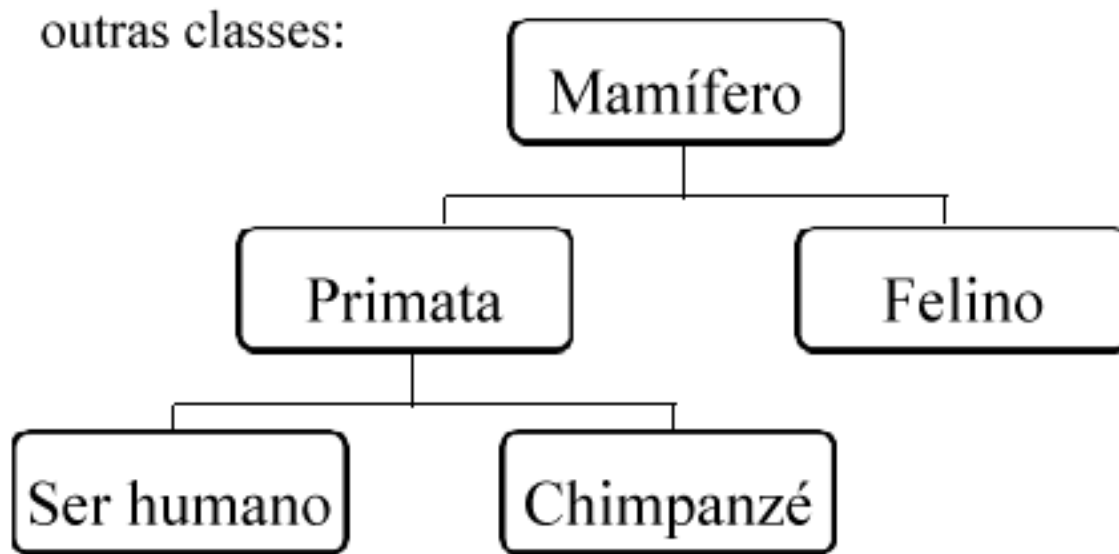


Motorista dirige um carro através dos pedais, alavanca de marchas e volante. Questões a respeito de motor estão escondidas para ele.



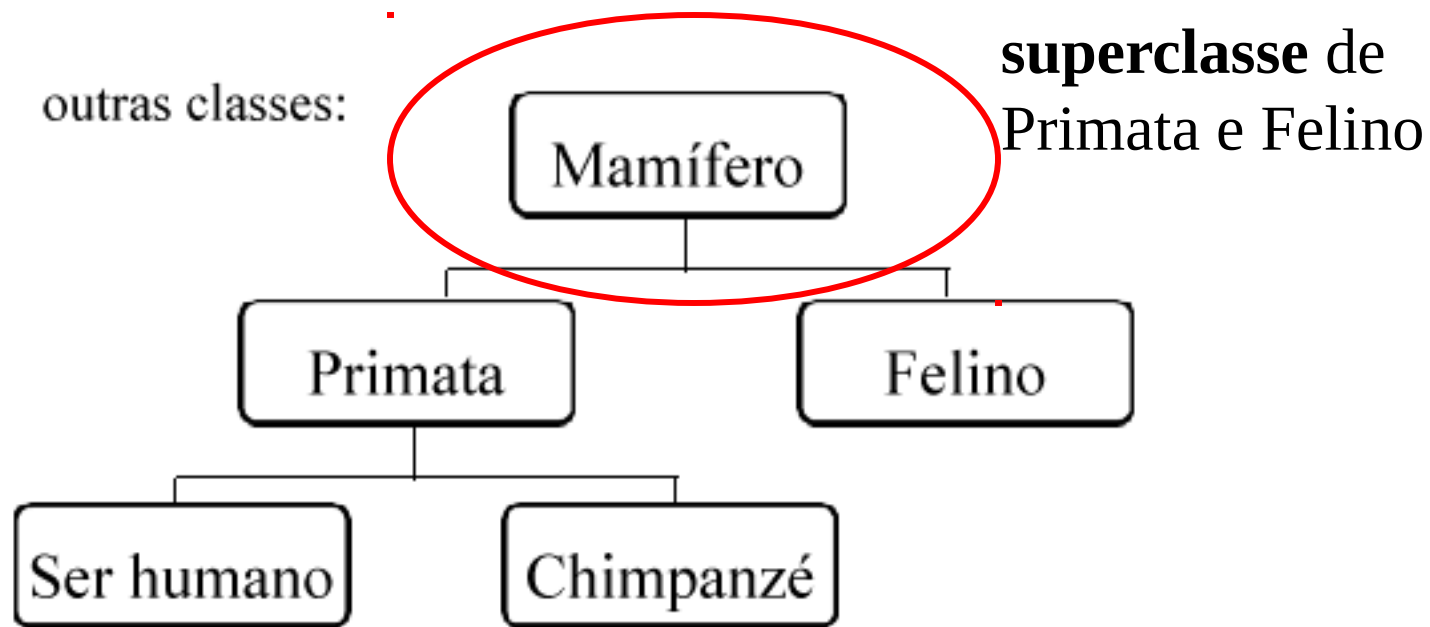
Orientação a Objetos

Uma **classe** pode também resumir elementos comuns de outras classes ...



Surge então o conceito de **subclasse** e **superclasse**.

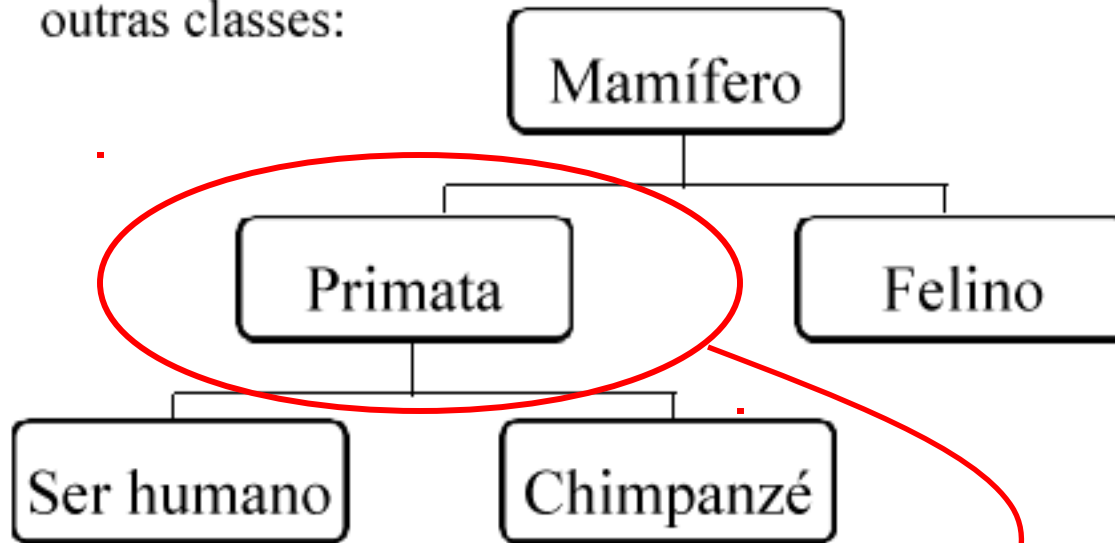
Orientação a Objetos



Orientação a Objetos

Conceito de Herança

outras classes:

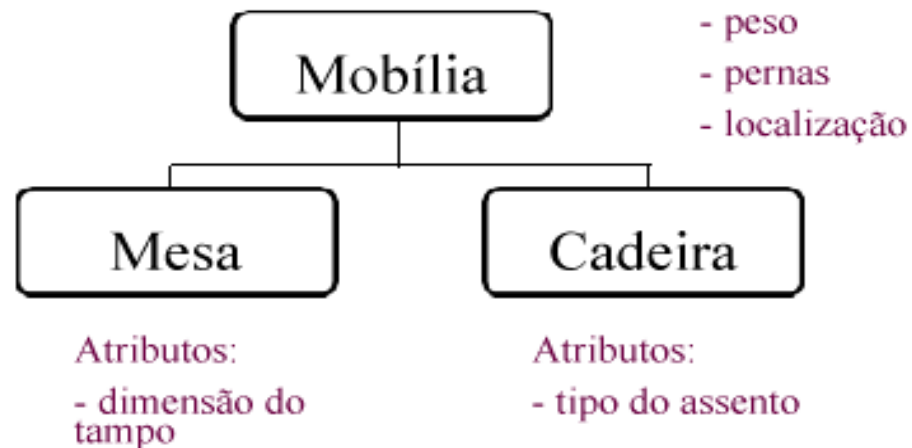


superclasse para Ser humano e Chimpanzé, as quais são subclasses de Primata

Orientação a Objetos

13

Outro exemplo de herança:



Um objeto, com estado:

- marrom
- 17 kg
- 4
- sala A200
- 1 x 1,5 m

Um objeto, com estado:

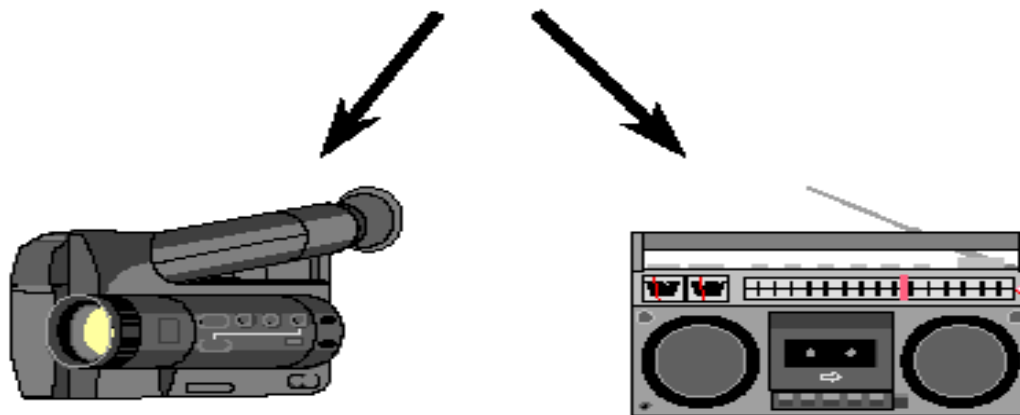
- preta
- 5 kg
- 4
- sala A200
- ondulado

Orientação a Objetos

Polimorfismo

- É o nome dado à capacidade que objetos diferentes têm de responder a uma mesma mensagem.
- Mesmo nome (mensagem), formas de execução diferentes, próprias de cada objeto.

Gravar



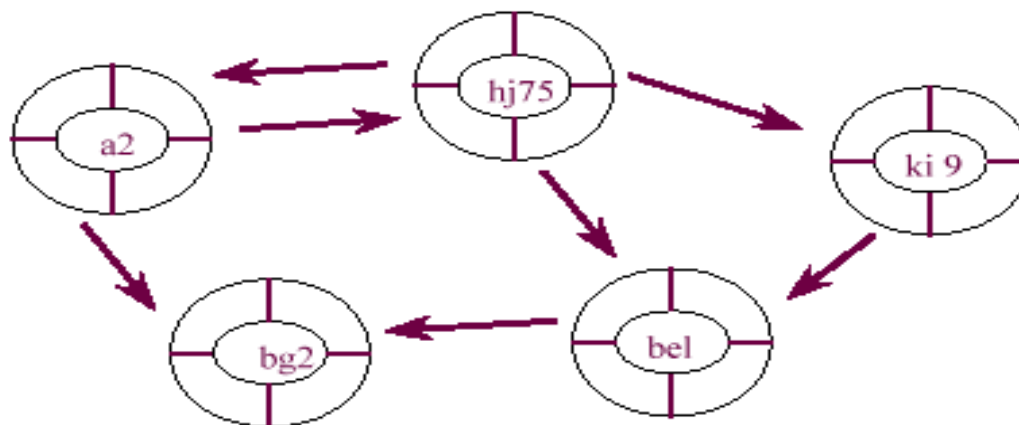
Com o polimorfismo o usuário pode enviar uma mensagem genérica e abandonar detalhes sobre a exata implementação sobre o objeto receptor.

Orientação a Objetos

Portanto,

O que é Orientação a Objetos ?

- É uma maneira de organizarmos o software como uma coleção de objetos discretos que incorporam :
 - uma Estrutura de Dados; e
 - um Comportamento Associado.



Modelagem com UML

- A Unified Modeling Language (UML) é uma linguagem para especificar, visualizar e documentar modelos de software orientados a objeto
- A UML é composta por muitos elementos que representam as diferentes partes de um sistema.
- Os elementos são usados para criar diagramas

Orientação a Objetos

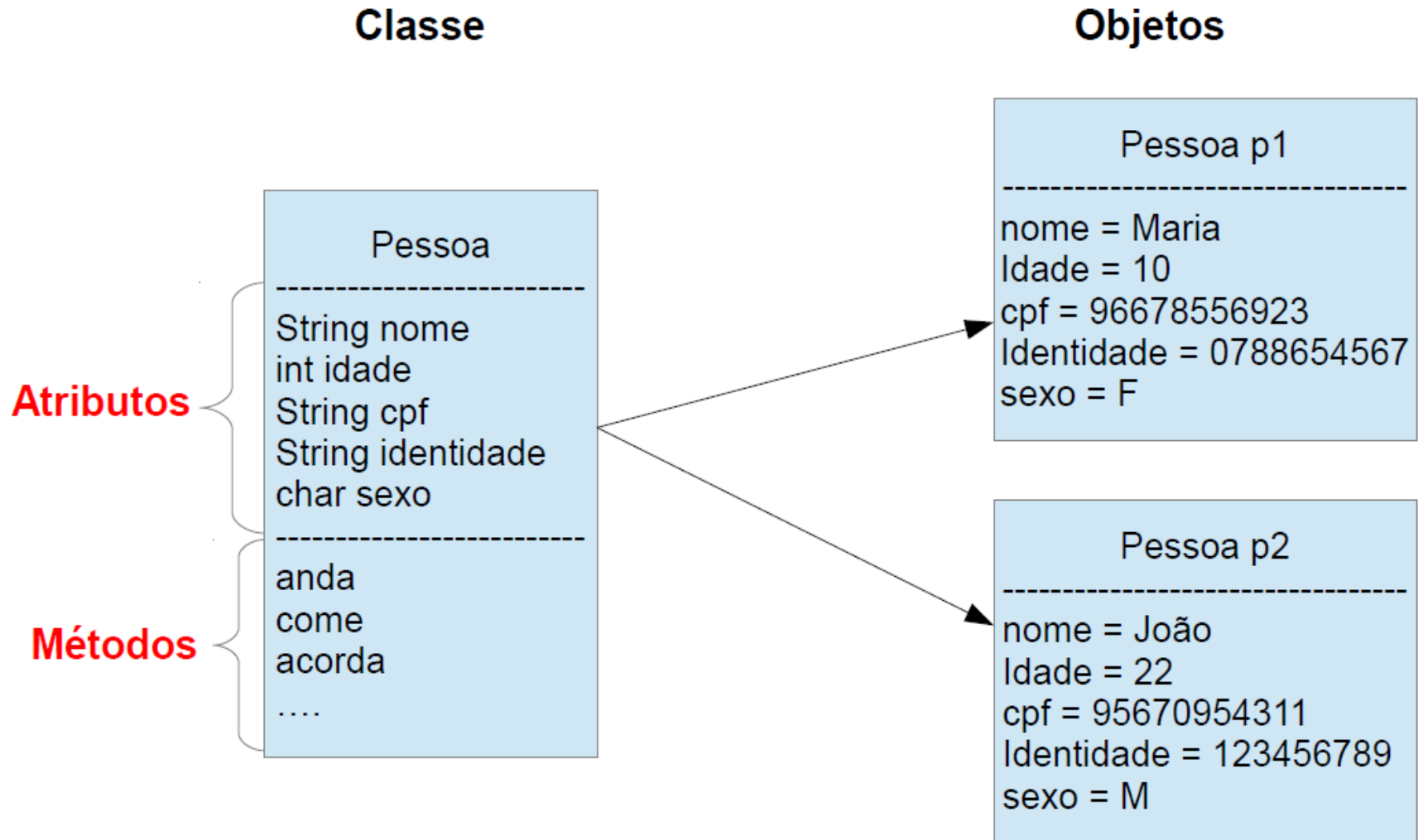
Alguns Diagramas da UML

- **Diagrama de Caso de Uso** mostra atores (pessoas ou outros usuários do sistema), casos de uso (os cenários onde eles usam o sistema), e seus relacionamentos
- **Diagrama de Classe** mostra classes e os relacionamentos entre elas
- **Diagrama de Sequência** mostra objetos e uma sequência das chamadas do método feitas para outros objetos.
- **Diagrama de Colaboração** mostra objetos e seus relacionamentos, colocando ênfase nos objetos que participam na troca de mensagens
- **Diagrama de Estado** mostra estados, mudanças de estado e eventos num objeto ou uma parte do sistema
- **Diagrama de Atividade** mostra atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema
- ...

Criando uma Classe - Modelo

- Pessoa
 - O que uma pessoa tem? **(Atributos)**
 - Nome
 - Idade
 - CPF
 - Identidade
 - Sexo
 - O que uma pessoa faz? **(Métodos)**
 - Acorda
 - Anda
 - Come
 - ...

Representação UML



Criando Tipos de Dados (abstração)

- Domínio *Banco*

- Cliente
- Conta
 - Conta corrente
 - Conta poupança
 - Conta salário
- Investimento
- Contrato
- Convênio
- Boleto Bancário
- Cheque
- Funcionário
 - Gerente
 - Caixa
- Agência
- Pagamento
- Financiamento
- Etc.

Criando um tipo de dado – Uma Classe

– Modelo

- Conta

- O que uma conta tem? **(Atributos)**

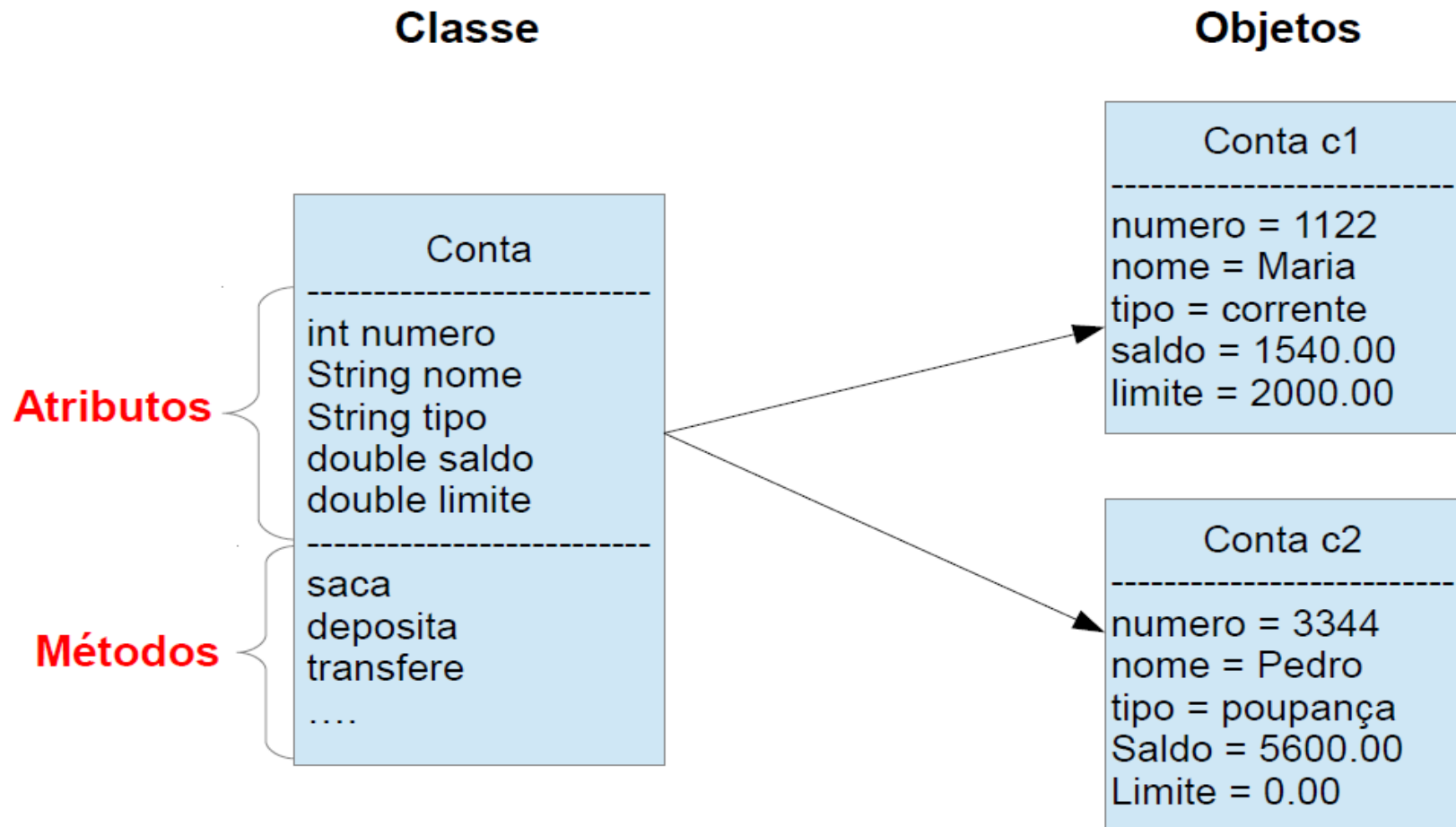
- Número da conta
 - Nome do dono
 - Tipo da conta
 - Saldo
 - Limite

- O que uma conta faz? **(Métodos)**

- Saca um valor x
 - Deposita um valor x
 - Transfere um valor x para outra conta c2

Orientação a Objetos

Representação UML



Orientação a Objetos

Classe em Java

```
class Conta {  
    // definição dos atributos  
    int numero;  
    String nome;  
    String tipo;  
    double saldo;  
    double limite;  
  
    //definição dos métodos  
    void saca(double valor) {  
        saldo = saldo - valor;  
    }  
    void deposita(double valor) {  
        saldo = saldo + valor;  
    }  
    void transfere(double valor, Conta c2) {  
        saldo = saldo - valor;  
        c2.saldo = c2.saldo + valor;  
    }  
}
```

Orientação a Objetos

Testando a Classe Conta

```
class Programa {
```

```
    public void main (String[] args) {
```

```
        Conta minhaConta;
```

```
        minhaConta = new Conta();
```

```
        minhaConta.saldo = 200.00;
```

```
        minhaConta.nome = "Maria";
```

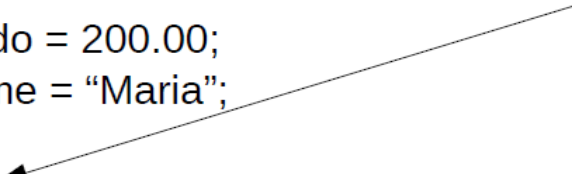
```
        minhaConta.deposita(450.00);
```

```
        MinhaConta.saca(100.00)
```

```
        System.out.println("Saldo Atual = " + minhaConta.saldo);
```

```
    }
```

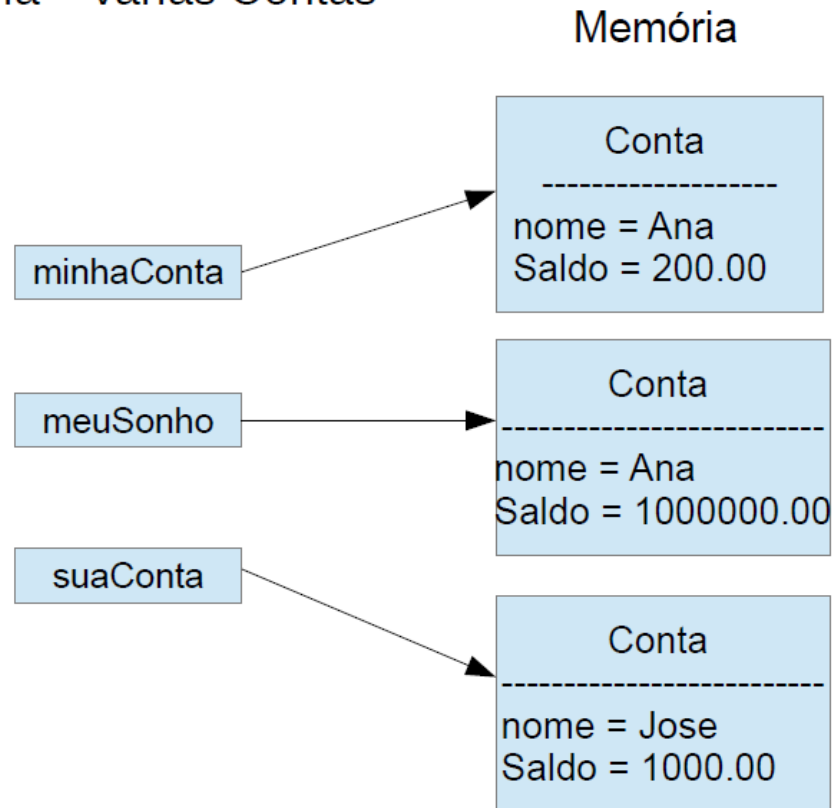
Invocando métodos



Orientação a Objetos

Criando vários Objetos em Memória – várias Contas

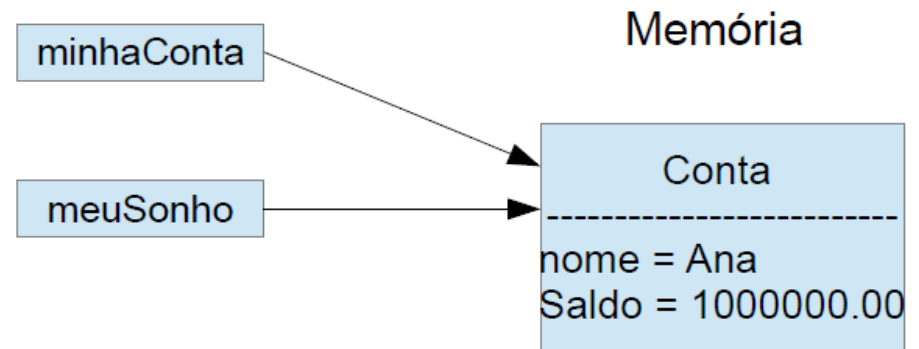
```
class Programa {  
    public void main (String[] args) {  
        Conta minhaConta = new Conta();  
        minhaConta.saldo = 200.00;  
        minhaConta.nome = "Ana";  
  
        Conta meuSonho = new Conta();  
        meuSonho.saldo = 1000000.00;  
        meuSonho.nome = "Ana";  
  
        Conta suaConta = new Conta();  
        suaConta.saldo = 1000.00;  
        suaConta.nome = "José";  
    }  
}
```



Orientação a Objetos

Atribuindo variáveis à mesma referência

```
class Programa {  
    public void main (String[] args) {  
        Conta minhaConta = new Conta();  
        minhaConta.saldo = 200.00;  
        minhaConta.nome = "Ana";  
  
        Conta meuSonho = new Conta();  
        meuSonho.saldo = 1000000.00;  
        meuSonho.nome = "Ana";  
  
        minhaConta = meuSonho;  
        System.out.println(minhaConta.saldo);  
        System.out.println(meuSonho.saldo);  
    }  
}
```



Orientação a Objetos

Método Transfere para outra Conta

```
class Conta {  
    ...  
    // atributos e métodos;  
  
    void transferePara(Conta destino, double valor) {  
        this.saldo = this.saldo - valor;  
        destino.saldo = destino.saldo + valor;  
    }  
}
```

```
class Programa {  
    public void main (String[] args) {  
        Conta minhaConta = new Conta();  
        minhaConta.saldo = 1000.00;  
  
        Conta contaDestino = new Conta();  
        contaDestino.saldo = 200.00;  
  
        minhaConta.transferePara(contaDestino,100.00);  
        System.out.println(minhaConta.saldo);  
        System.out.println(contaDestino.saldo);  
    }  
}
```

Orientação a Objetos

Modelo da Conta

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

Acrescentando Atributos à Conta

Informações necessárias para abrir uma conta:

- Nome
- Sobrenome
- Data de Nascimento
- CPF
- Endereço
- Telefone
- Etc..

São atributos de Cliente e não de Conta

Um Cliente pode ter mais de uma Conta

Orientação a Objetos

Associando Classes

```
class Cliente {  
    String nome;  
    String endereco;  
    String cpf; ...  
}  
  
class Conta {  
    int numero;  
    double saldo;  
    double limite;  
    String nome;  
    Cliente titular;  
}
```

```
class Programa {  
    public void main (String[] args) {  
        Conta minhaConta = new Conta();  
  
        Cliente c = new Cliente();  
        minhaConta.titular = c;  
  
        minhaConta.titular.nome = "Ana";  
  
    }  
}
```

minhaConta agora tem uma referência para o Cliente correspondente a c

Forma de Acessar o Cliente

Orientação a Objetos

Modelo da Conta

