ORIENTAÇÃO A OBJETOS

Professor: Katyusco de Farias Santos

Motivação

- Até agora, vimos programação procedural em Java.
  - Várias funções para validação podem estar soltas;
  - A lógica de negócios pode estar fragmentada em vários arquivos;

```
String cpf = "123";

validarCpf(cpf);
```

```
int anoNascimento = 1988;

calcularIdade(anoNascimento);
```

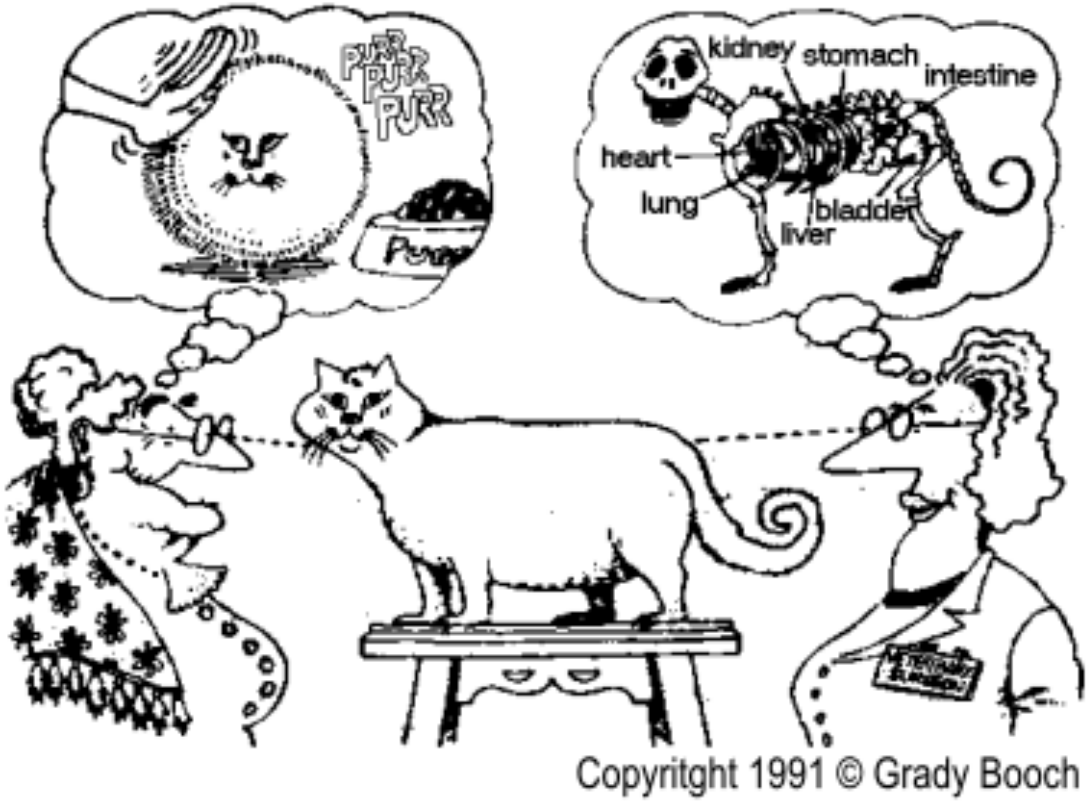
  - Muito código pode ser escrito desnecessariamente em um paradigma procedural.

Motivação

- Orientação a objetos:
  - Concentra responsabilidades nos pontos certos;
    - Evita replicação de código para métodos e funções.
  - Melhora o encapsulamento da lógica de negócio;
  - Facilita a manutenção e expansão do código.

Abstração

- Conceito importante para OO.



Abstração

- Abstração:
  - Técnica para lidar com a complexidade de um problema;
  - Destaca os aspectos importantes do objeto real, abstraído segundo o observador;
  - Ignora os detalhes não relevantes para o observador.
- Exemplos diários:
  - Dirigir um carro;
  - Retirar dinheiro no caixa eletrônico.

Abstração

- Exemplos:
  - Um mapa mundi;
  - Um modelo de avião;
  - Uma maquete de um edifício;
  - Um programa de computador;
  - Uma planta baixa de uma casa.

Abstração

- Abstração de processos ou sistemas:
  - Dividimos um programa em subprogramas menores e mais fáceis de escrever e compreender;
  - Para usar um subprograma escrito por terceiros, abstraímos a sua implementação e nos concentramos na forma como ativaremos o subprograma (interface).
- Abstração de dados:
  - Serve para representação de entidades do mundo real, dependendo do domínio do problema:
    - Focar nas propriedades e operações que interessam ao sistema.

Classes e Objetos

- Abstração de uma conta do mundo real.

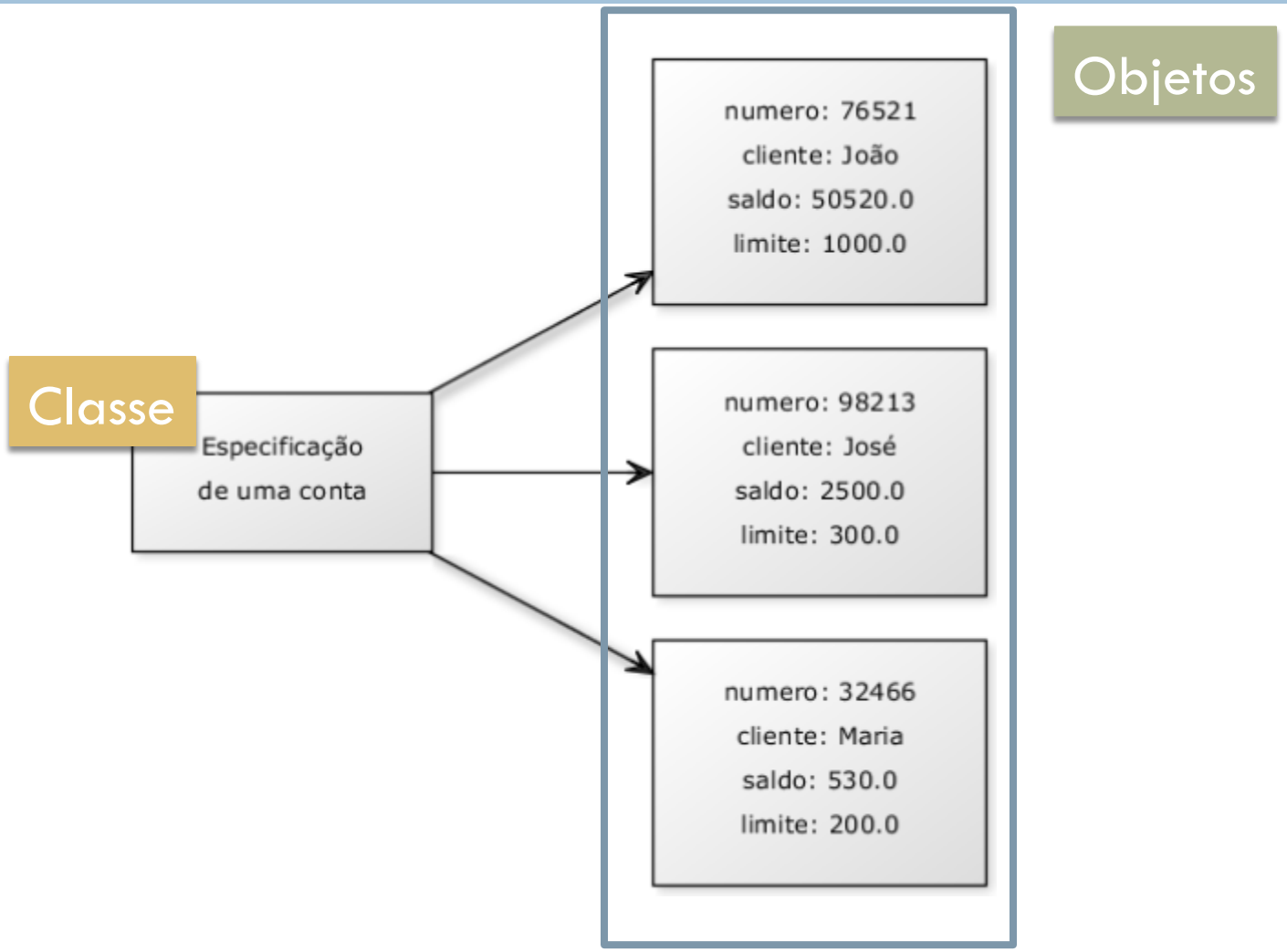
O que toda conta tem e é importante?
- Número da conta;
- Nome do dono da conta;
- Saldo;
- Limite.

Atributos ou propriedades de uma *Conta*

Quais operações são importantes em uma conta?
- Sacar;
- Depositar;
- Ver saldo;
- Transferir.

Comportamento de uma *Conta*

Classes e Objetos



Classes e Objetos

- *Classe* vem da taxonomia de biologia;
  - Todos os seres vivos de uma mesma **classe** biológica têm uma séria de **atributos** e **comportamentos** em comum, mas com valores diferentes.

- Exemplos.
  - Receita de bolo (classe); Bolo feito (objeto).
  - Planta de uma casa (classe); Casa construída (objeto).

Objetos

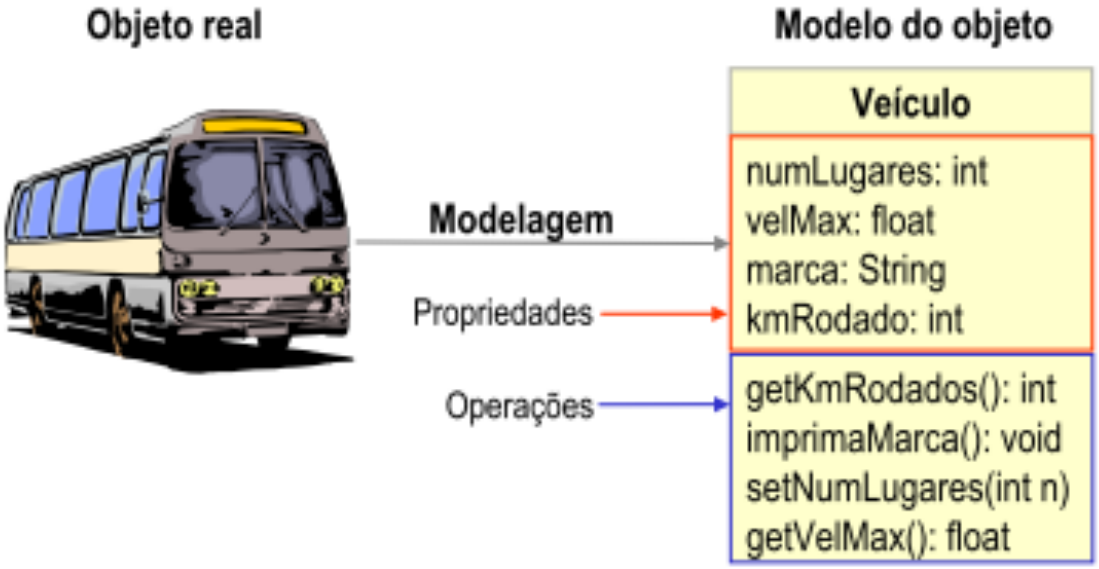
- Objeto:
  - Modela alguma parte da realidade e é algo que existe no tempo e no espaço;
  - Conceito de objetos para entender melhor o mundo;
  - Significado em programação:
    - "Um objeto representa uma entidade, unidade ou item identificável, individual, real ou abstrato, com um papel bem definido no domínio do problema". (Grady Booch, 91)
  - Exemplos:
    - Cadeira, bicicleta, pessoa, carro, número complexo, reação química, etc.

Objetos

- Objetos:
  - Nem tudo é um objeto:
    - Tempo, métricas, sensações e sentimentos humanos não são, em geral, modelados como objetos;
    - São, em geral, tratados como atributos de objetos.
  - Descobrir as entidades importantes do domínio do problema, bem como os atributos e os comportamentos, é a parte mais importante:
    - Papéis envolvidos: analista de sistemas e programador.

Objetos

- Como entidades reais do domínio do problema se transformam em objetos de software?



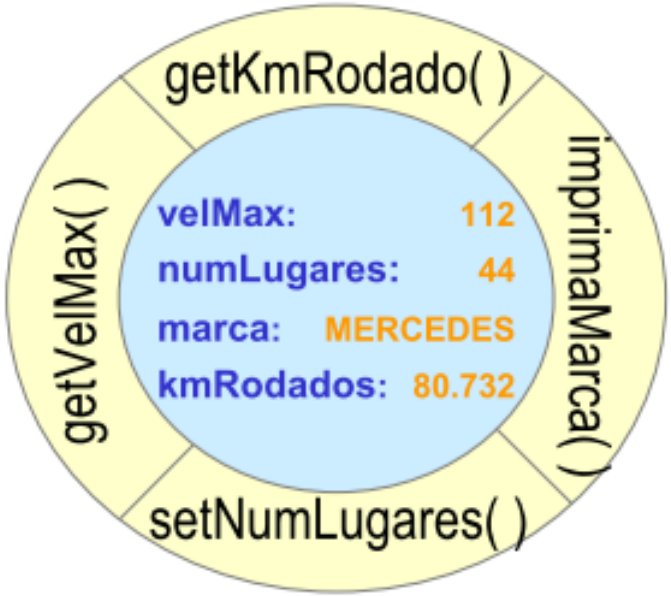
Abstração de dados

Objetos

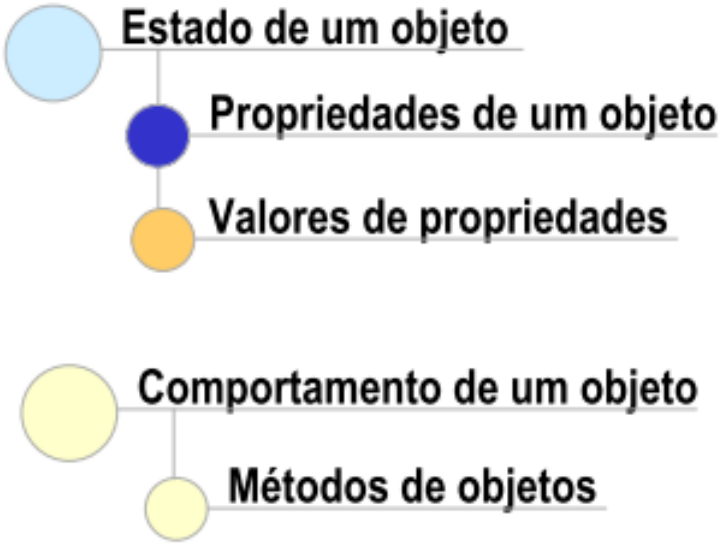
- Características chaves de um objeto:
  - **Estado**: conjunto de atributos (ou propriedades) de um objeto e seus respectivos valores;
  - **Identidade**: cada objeto criado é único e ocupa seu espaço próprio na memória;
  - **Comportamento**: conjunto de operações que o objeto está apto a executar.

Objetos

- Estado e comportamento.

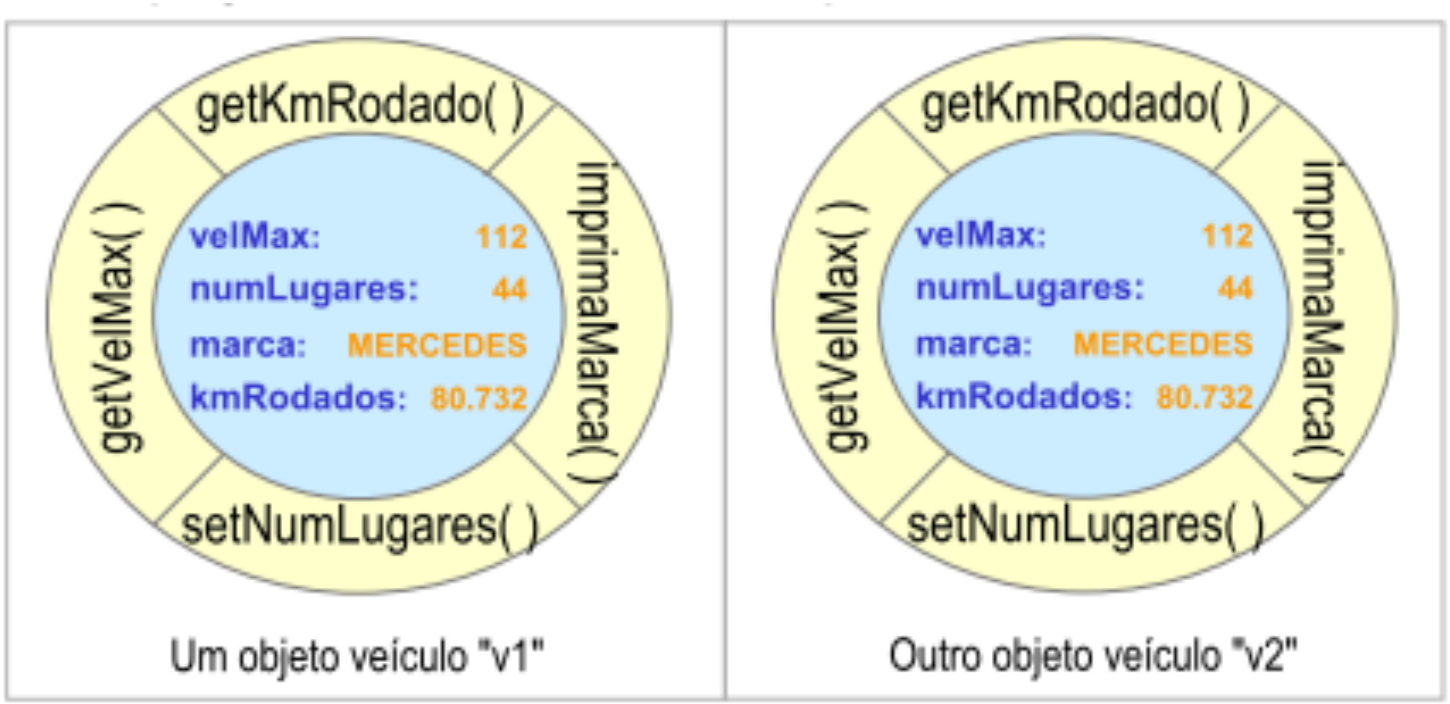


Um objeto veículo



Objetos

- Todo objeto criado ocupa seu próprio espaço na memória do computador (*Identidade*).

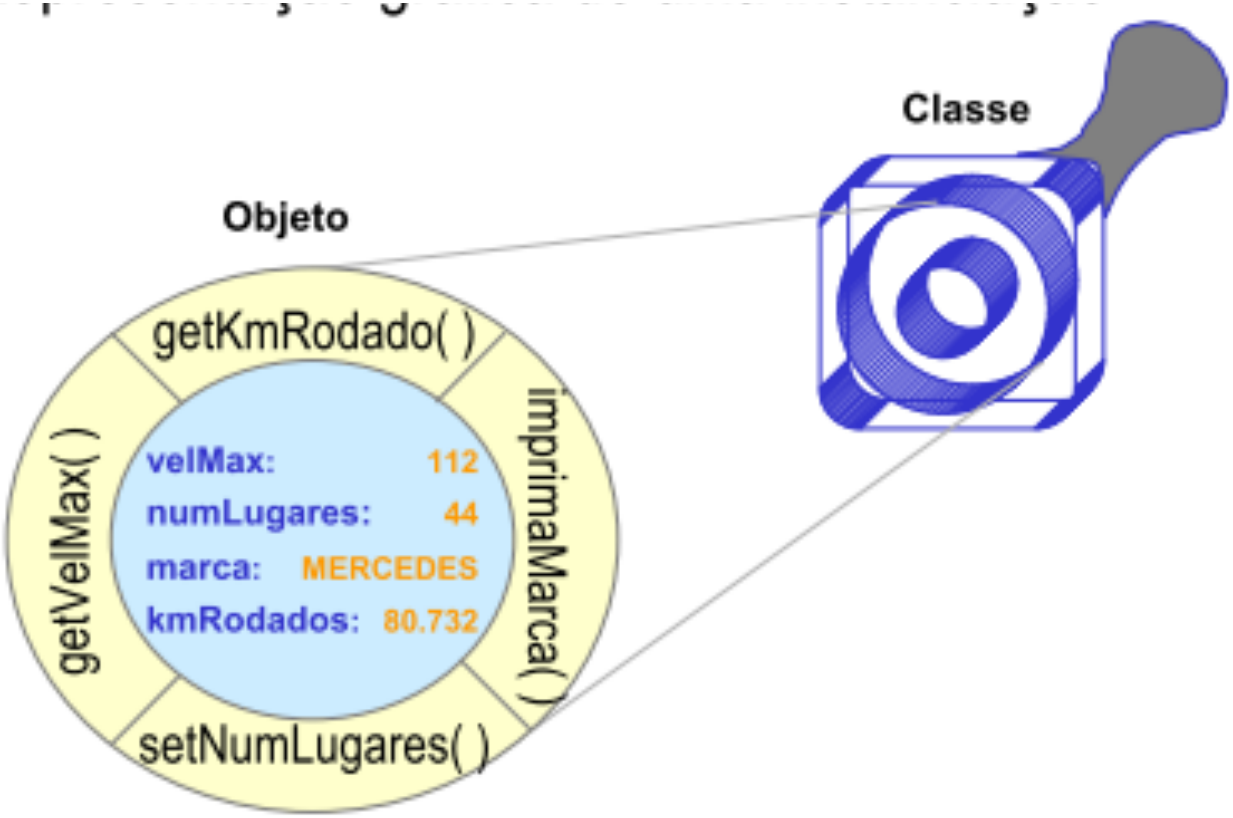


Classes

- Um objeto é criado por uma operação de ***instanciação*** sobre uma ***classe***.
- Classe:
  - Unidade lógica (geralmente, um arquivo) do paradigma OO onde são definidos os atributos e métodos que uma categoria de objetos terá;
  - A classe é uma especificação (uma "fôrma") pelo qual os objetos serão **criados**, isto é, **instanciados**;
  - Programar OO é, em grande parte do tempo, escrever classes.

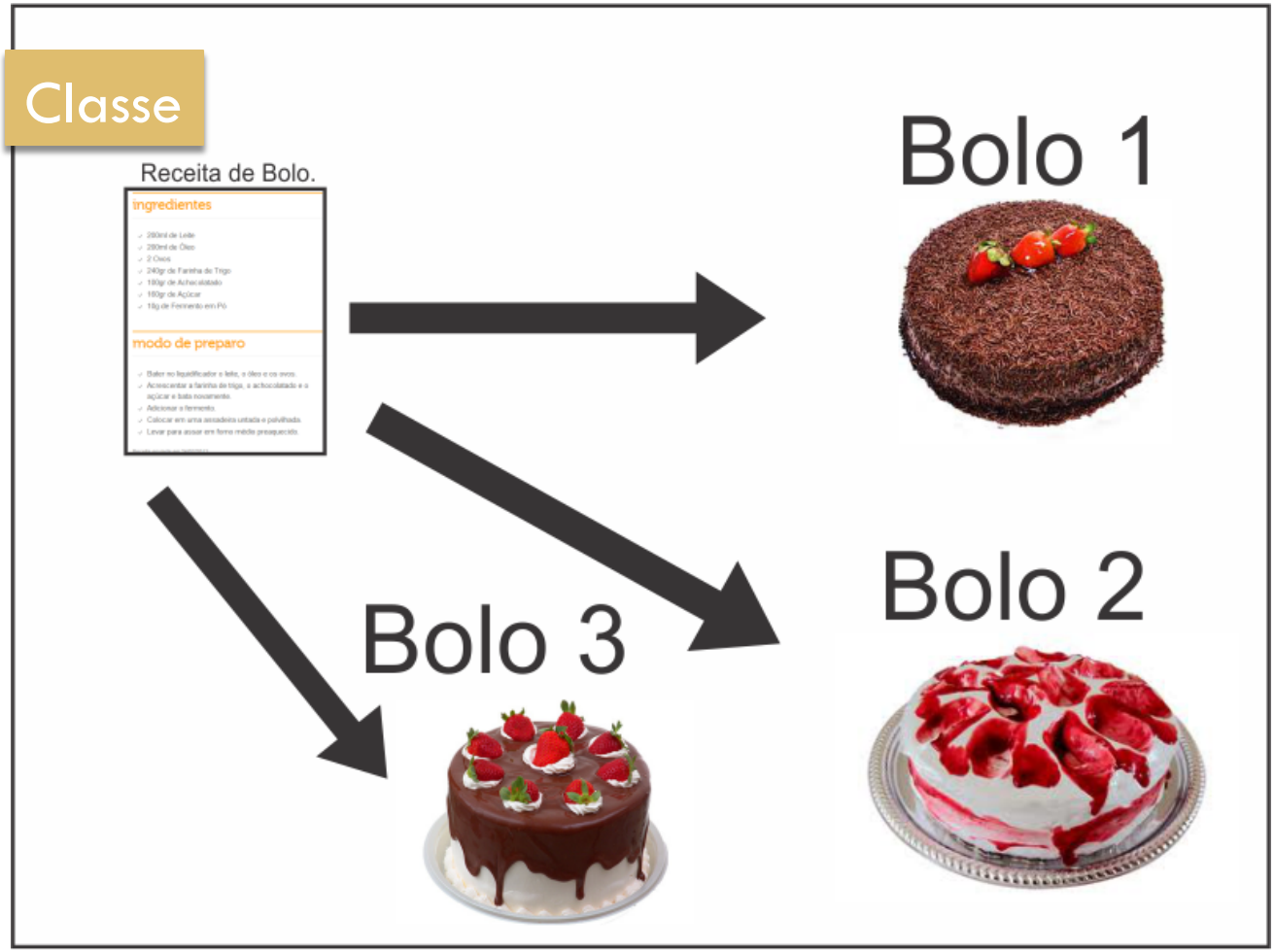
Classes

- Representação gráfica de uma instanciação.



Classes

- Representação gráfica de uma instanciação.



Classes

- Exemplo de código para uma Conta.

Classe

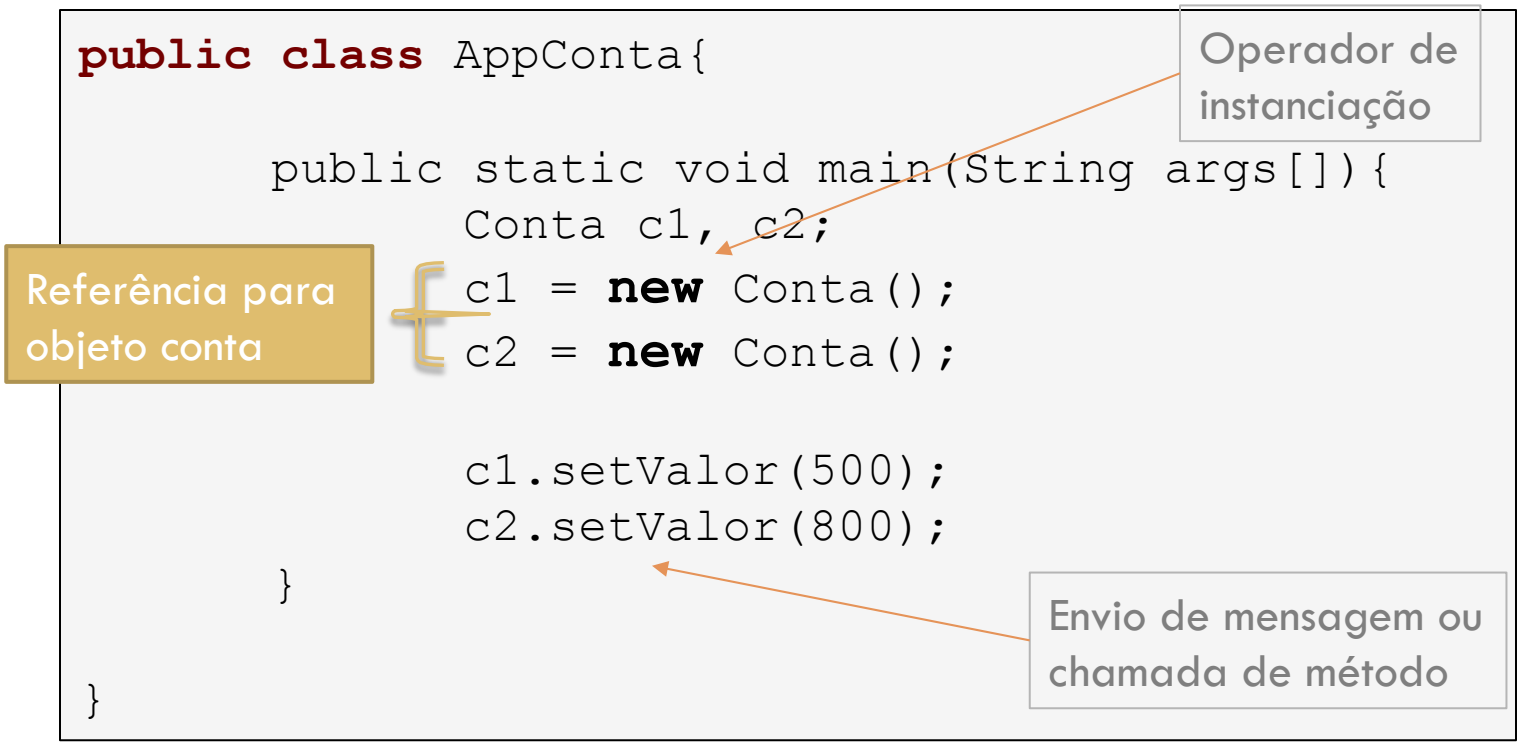
```
class Conta{
        private int numero;
        private String nomeDono;
        private double saldo;
        private double limite;

        public void depositar(double valor){
                saldo = saldo + valor;
        }
        public double sacar(double valor){
                if(valor <= saldo){
                        saldo = saldo - valor;
                } else{ return -1; }
                return saldo;
        }
}
```

Atributos

Métodos

Classes

- Instanciação de objetos.



Referências para Objetos

- A maioria dos objetos em um programa Java são acessados por variáveis chamadas **referências.**
  - As variáveis devem ser declaradas e tipificadas em tempo de compilação.

```
String str;

Pessoa pessoa;

Conta conta;
```

Referências para Objetos

- A declaração indica o tipo da referência.

```
Veiculo v;                    //declaração
v = new Veiculo();            //instanciação
v.setNumLugares(50);          //uso
v = null;                     //desprezo
```

Referência para Objetos

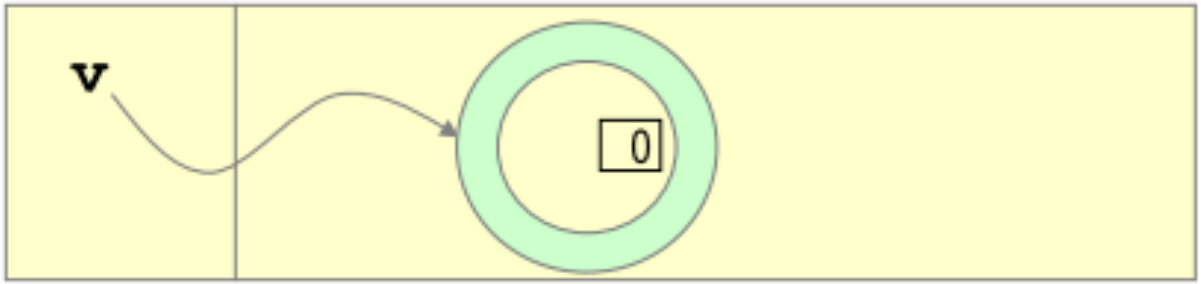
- A declaração indica o tipo da referência.

```
Veiculo v;                    //declaração
v = new Veiculo();            //instanciação
v.setNumLugares(50);          //uso
v = null;                     //desprezo
```

Referência para Objetos

- A instanciação cria objetos na memória.

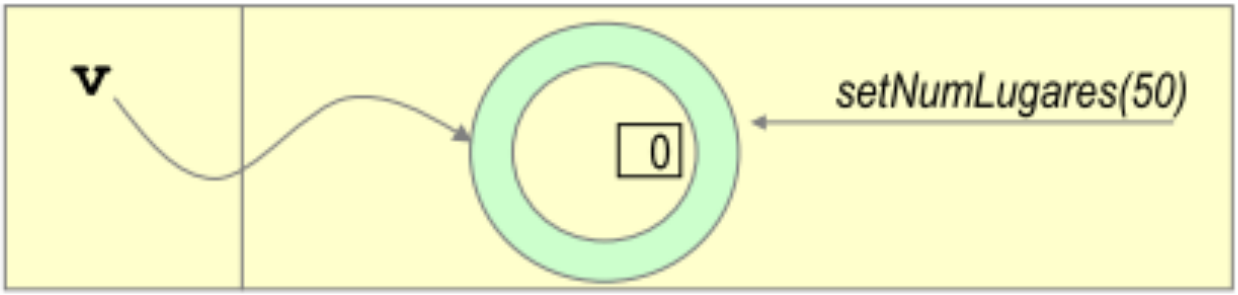
```
Veiculo v;                    //declaração
v = new Veiculo();            //instanciação
v.setNumLugares(50);          //uso
v = null;                     //desprezo
```



Referência para Objetos

- Uma mensagem é enviada ao objeto.

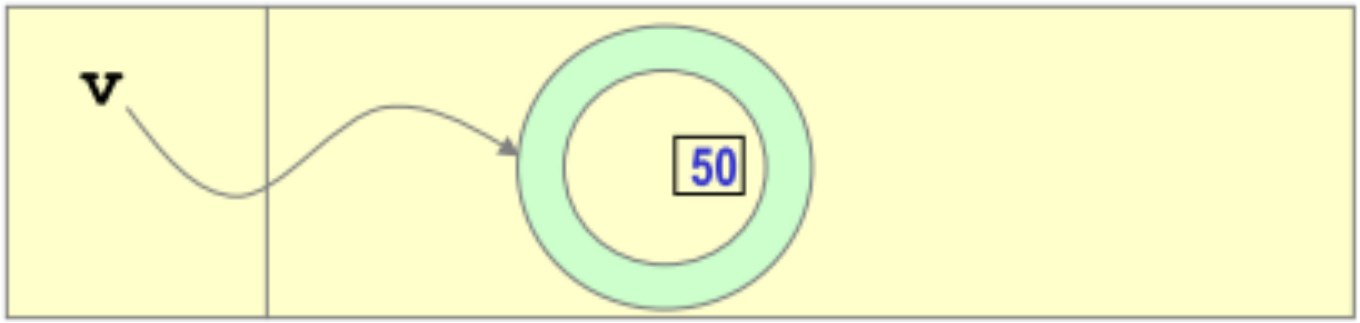
```
Veiculo v;                    //declaração
v = new Veiculo();            //instanciação
v.setNumLugares(50);          //uso
v = null;                     //desprezo
```



Referência para Objetos

- E faz o método ser executado.

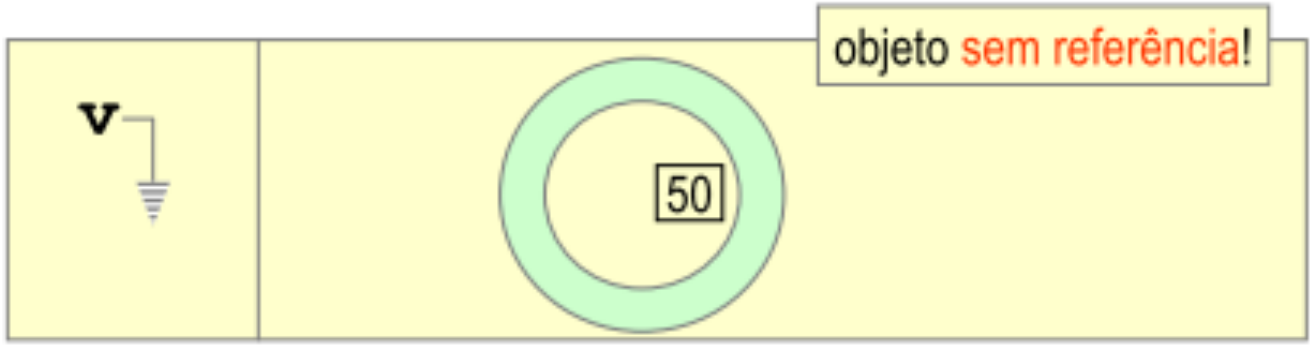
```
Veiculo v;                      //declaração
v = new Veiculo();              //instanciação
v.setNumLugares(50);            //uso
v = null;                       //desprezo
```



Referência para Objetos

- A referência pode deixar de apontar para o objeto.

```
Veiculo v;                    //declaração
v = new Veiculo();            //instanciação
v.setNumLugares(50);          //uso
v = null;                     //desprezo
```



Mensagens e Métodos

- Um método é a implementação de uma operação que o objeto é capaz de executar;
- O conjunto de métodos de um objeto faz parte do **comportamento** deste objeto (ou **interface**);
- Métodos são codificados dentro da classe do objeto;
- Métodos podem receber parâmetros e retornar valores:
  - Um método é semelhante a uma função.

Mensagens e Métodos

- Um sistema orientado a objetos consiste de um conjunto de objetos se comunicando entre si, para resolver algum problema;

- A forma de ativar um método de um objeto é enviando-lhe uma **mensagem:**
  - Faz o método correspondente ser executado.

Mensagens e Métodos

- Comparação Orientação a Objetos x Estruturado.

Paradigma OO	Paradigma Estruturado
Declaração de métodos em classe	Declaração de funções ou procedimentos em módulos
Envio de uma mensagem para um objeto	Chamada de uma função ou procedimento em um outro ponto do código