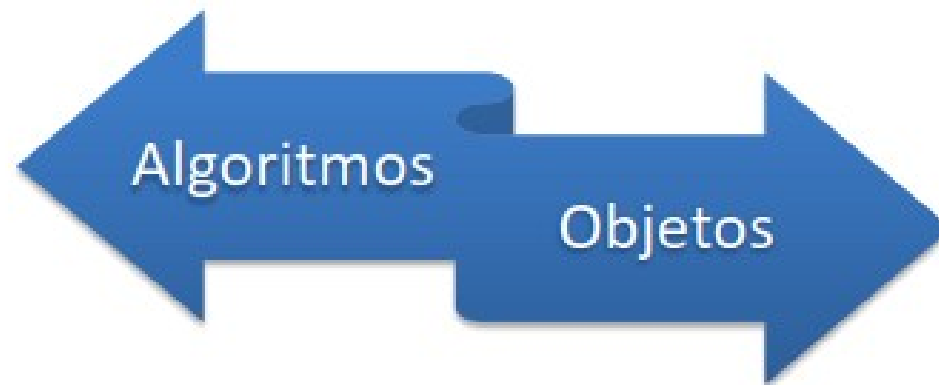


Conceitos de Orientação a Objetos

Conteúdo extraído dos slides de Leonardo Murta

Paradigma procedimental versus OO

- O **paradigma procedimental** organiza o programa em termos de **algoritmos**
- O **paradigma OO** organiza o programa em termos de **objetos**



~~Algoritmos~~ Objetos

- Podemos criar programa pensando em termos de **objetos ao invés de algoritmos?**
- O mundo é composto de objetos
 - Uma loja tem produtos, pedidos, estoque, etc.
 - Um restaurante tem mesas, garçons, comidas, bebidas, etc.
 - Uma universidade tem professores, alunos, disciplinas, etc.
 - Uma rodoviária tem ônibus, passageiros, bagagens, etc.
- E se **criarmos programas** basicamente **criando objetos** equivalentes ao mundo real, e fazendo com que esses **objetos se comuniquem?**

Objetos

- Definição
 - Um objeto é a **representação computacional de um elemento ou processo do mundo real**
 - Cada objeto possui suas **características** e seu **comportamento**

- Exemplos de Objetos

cadeira

mesa

caneta

lápis

carro

piloto

venda

mercadoria

cliente

aula

programa

computador

aluno

avião

Características de objetos

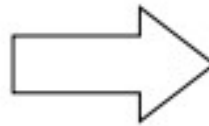
- Definição
 - Uma característica descreve uma propriedade de um objeto, ou seja, algum elemento que descreva o objeto.
 - Cada característica é chamada de **atributo** e funciona como uma **variável** pertencente ao objeto
- Exemplo de características do objeto **carro**
 - Cor
 - Marca
 - Número de portas
 - Ano de fabricação
 - Tipo de combustível

Comportamento de objetos

- Definição
 - Um comportamento representa uma ação ou resposta de um objeto a uma ação do mundo real
 - Cada comportamento é chamado de **método** e funciona como um **procedimento/função** pertencente ao objeto
- Exemplos de comportamento para o objeto **carro**
 - Acelerar
 - Frear
 - Virar para direita
 - Virar para esquerda

Mapeamento de Objetos

Objeto no Mundo Real



Objeto Computacional



Paradigma procedimental versus OO (exemplo: Agenda)

Paradigma Procedimental

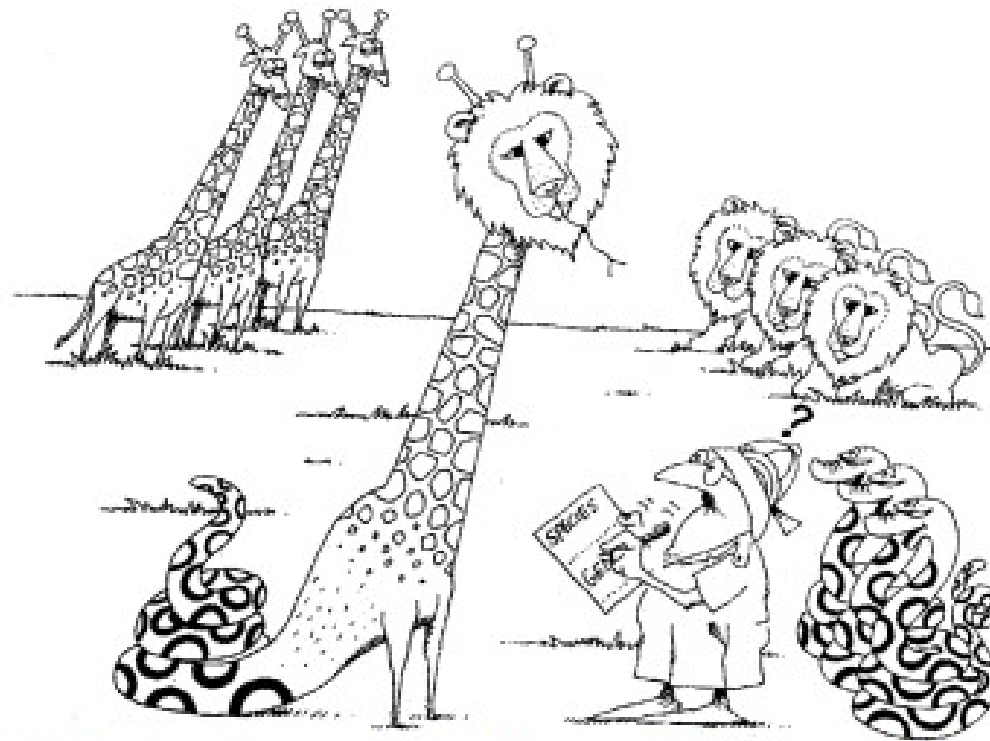
- Variáveis
 - Vetor de nomes
 - Vetor de endereços
 - Vetor de telefones
- Procedimentos
 - Listagem de todos os nomes
 - Listagem do endereço dado um nome
 - Listagem do telefone dado um nome
 - Adição de nome, endereço e telefone
 - Remoção de nome, endereço e telefone

Paradigma OO

- Objeto Agenda
 - Atributo
 - Vetor de Contatos
 - Métodos
 - Listagem de Contatos
 - Adição de um Contato
 - Remoção de um Contato
- Objeto Contato
 - Atributos
 - Nome
 - Endereço
 - Telefone
 - Métodos
 - Exibição de nome, endereço e telefone
 - Edição de nome, endereço e telefone

Classes versus objetos

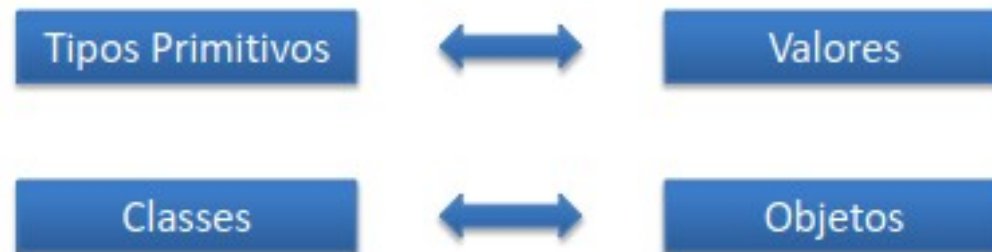
- A Classe é o tipo do Objeto



Fonte: livro "Object-Oriented Analysis and Design with Applications"

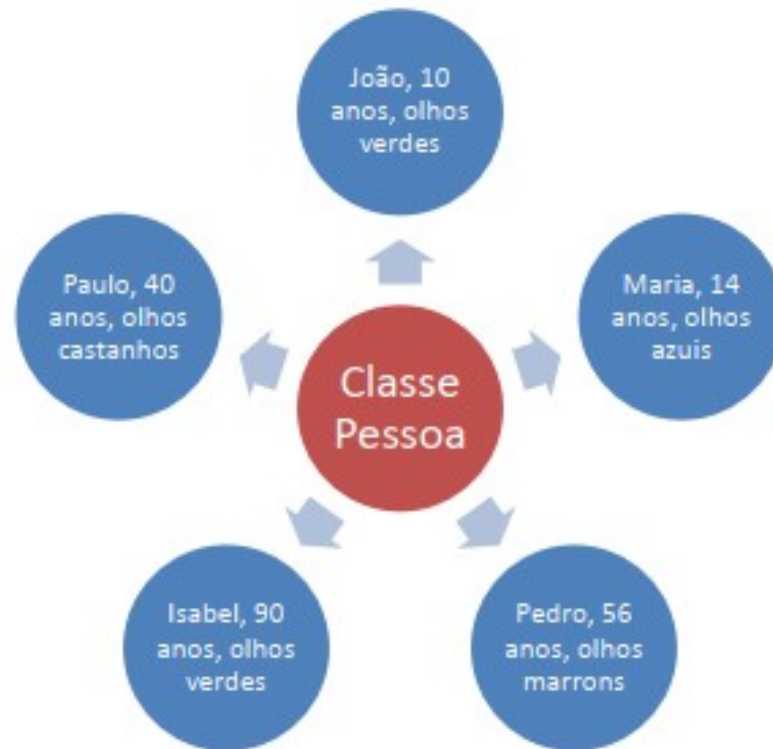
Classes versus objetos

- Valores têm tipos primitivos
 - 123 é um valor inteiro
 - True é um valor booleano
 - 12,3 é um valor real
- Objetos pertencem a classes
 - João, Pedro e Paulo são da classe Pessoa
 - Fusca e Ferrari são da classe Carro
 - Flamengo e Fluminense são da classe Time



Classes versus objetos

- Uma **classe** é uma **fôrma**, capaz de **produzir objetos**
- Os **programadores** **criam classes**, as **classes** **instanciam objetos**



Classes

- A classe descreve as características e comportamento de um conjunto de objetos
 - O objeto possuirá os atributos e métodos definidos na classe
 - O objeto é chamado de instância de sua classe
 - A classe é o bloco básico para a construção de programas OO

Exemplo de classe

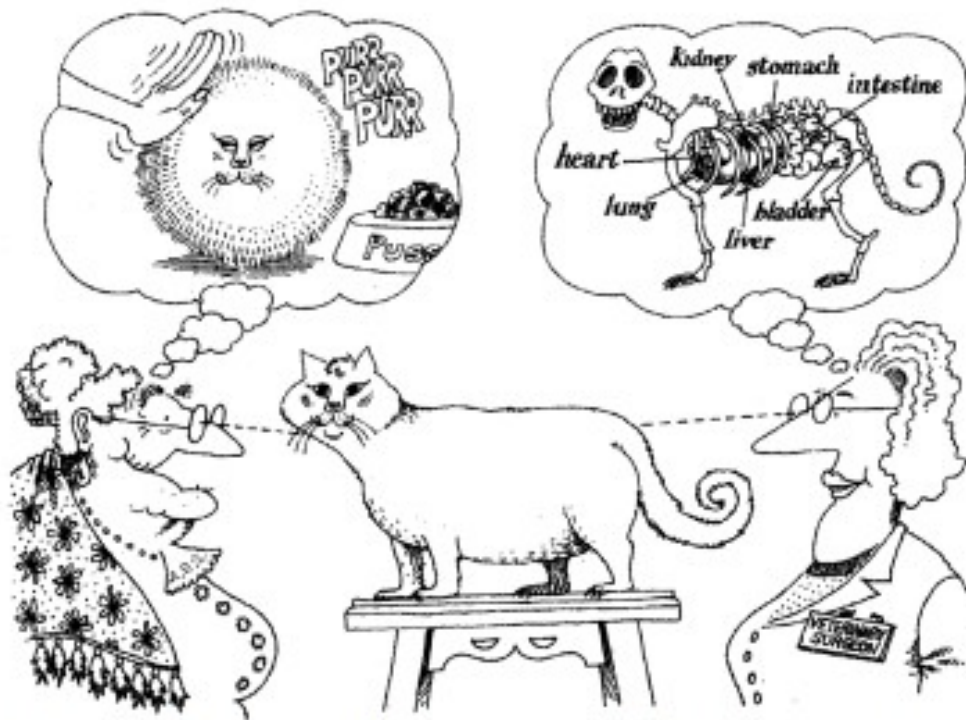
```
public class Carro {  
    private int velocidade;  
  
    public void acelera() {  
        velocidade++;  
    }  
  
    public void freia() {  
        velocidade--;  
    }  
}
```

*Atributos (características)
são variáveis globais
acessíveis por todos os
métodos da classe*

Métodos (comportamentos)

Abstração

- A representação computacional do objeto real deve se concentrar nas características que são relevantes para o problema



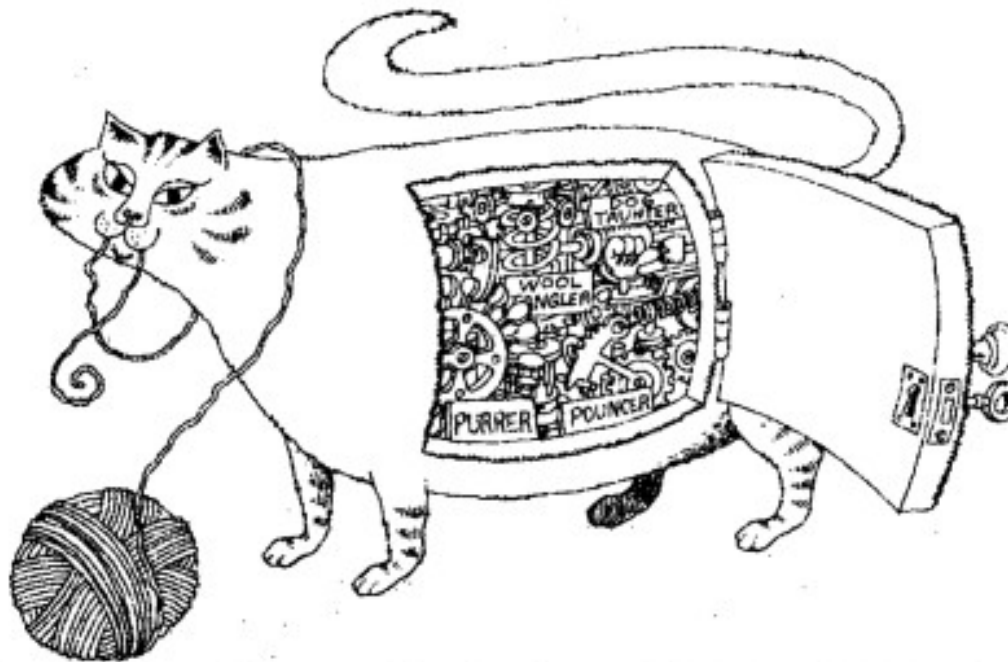
Fonte: livro "Object-Oriented Analysis and Design with Applications"

Abstração

- São criados somente os atributos e métodos necessários para o problema em mãos
- Quais seriam os atributos e métodos para o objeto Carro em cada uma das situações seguintes?
 - Sistema de uma locadora de carros
 - Sistema de uma revendedora de carros
 - Sistema de uma oficina mecânica
 - Sistema do DETRAN

Encapsulamento

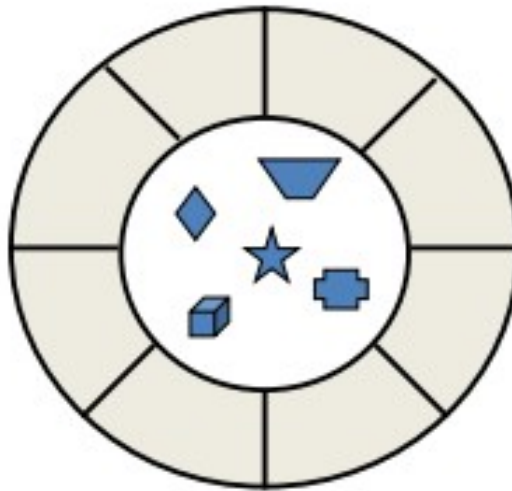
- O objeto deve esconder seus dados e os detalhes de sua implementação



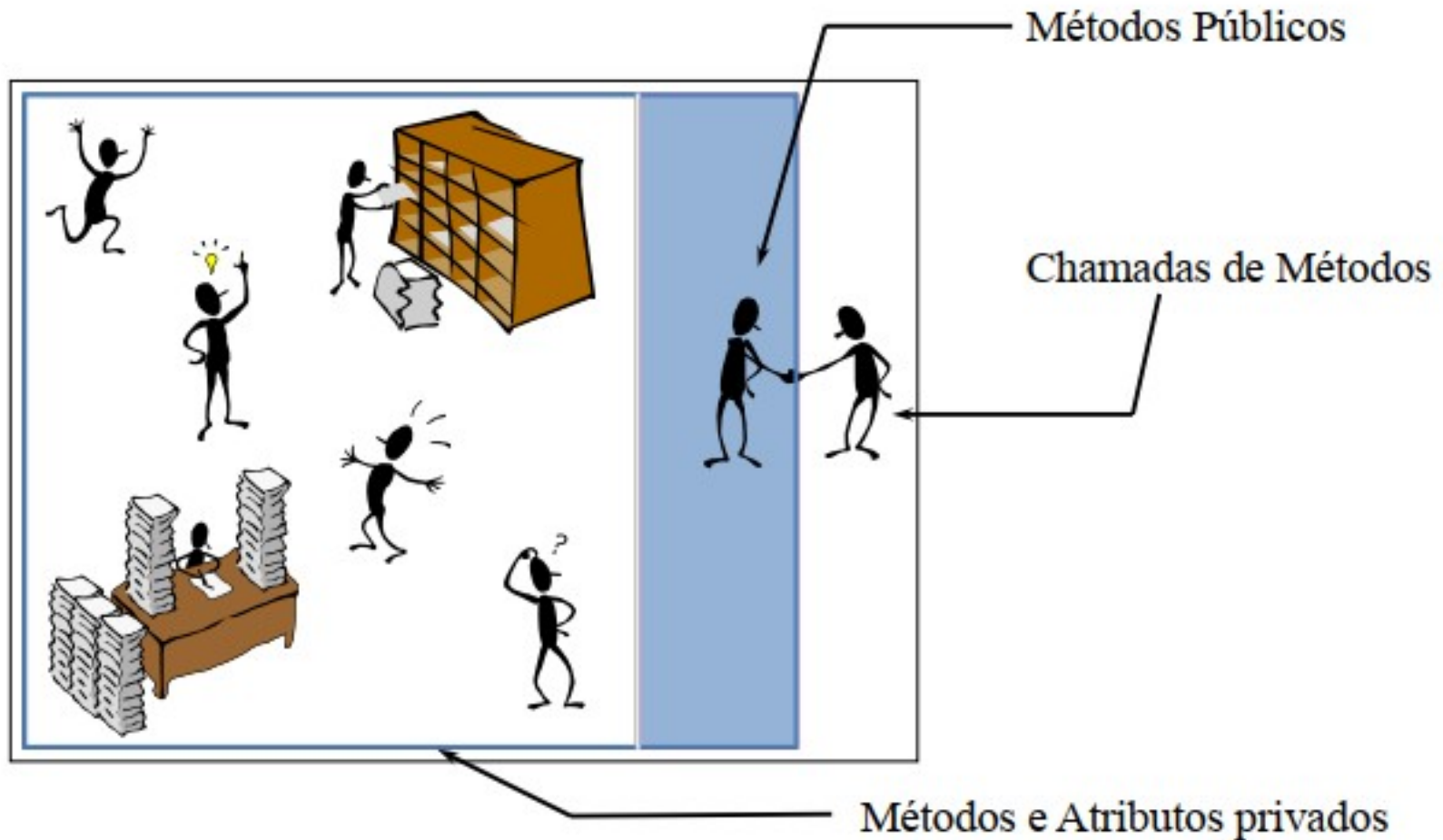
Fonte: livro "Object-Oriented Analysis and Design with Applications"

Encapsulamento

- Atributos e Métodos
 - Os métodos formam uma “cerca” em torno dos atributos
 - Os atributos não devem ser manipulados diretamente
 - Os atributos somente devem ser alterados ou consultados através dos métodos do objeto



Encapsulamento



Retomando – Classes/Objetos

- Classe: gabarito que define a forma de um objeto (tipo de dados)
- Classe: especifica tanto código quanto os dados
- Objetos são instâncias de uma classe
- Uma classe é uma abstração lógica até que um objeto tenha sido criado
- Quando se define uma classe, são declarados os dados que ela contém e o código que opera nesses dados (membros de uma classe)

Retomando – Classes/Objetos

- Notação Classes

Nome da Classe

Nome da Classe

lista de atributos

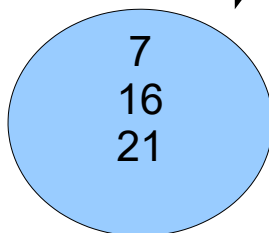
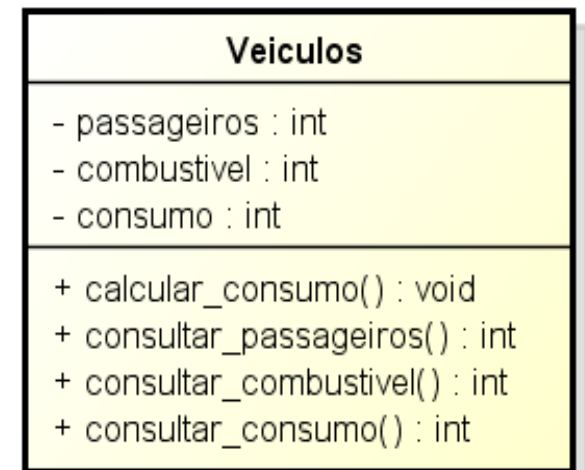
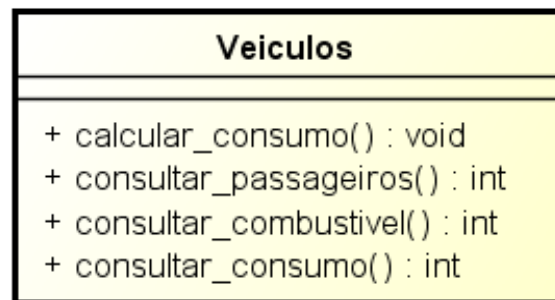
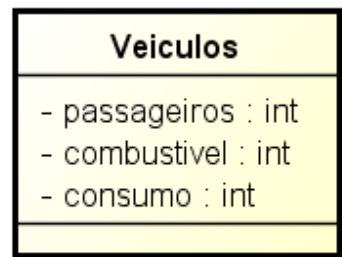
Nome da Classe

lista de operações

Nome da Classe

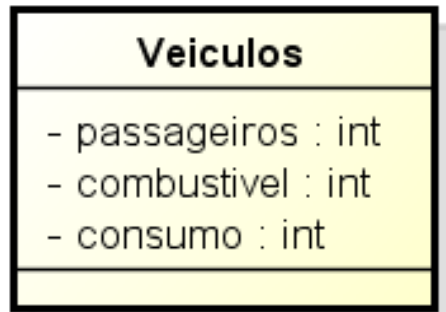
lista de atributos

lista de operações



Minivan (objeto de veículos)

Classes/Objetos – Um pouco de código JAVA



powered by Astah

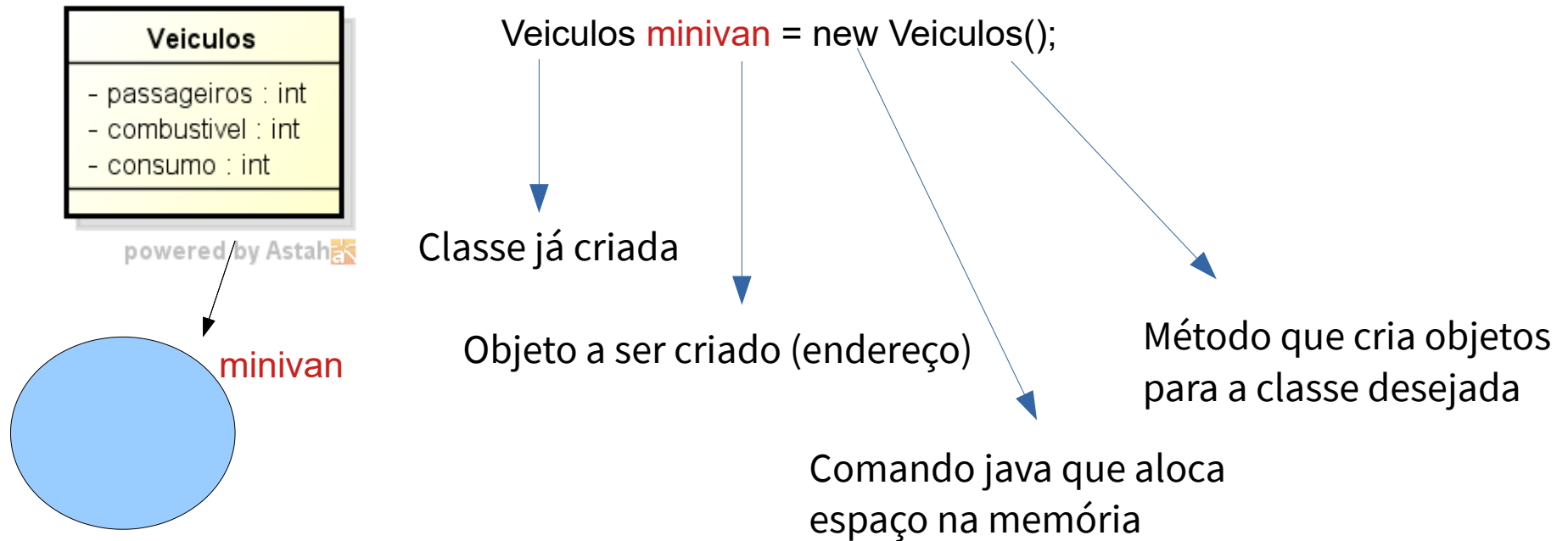
```
public class Veiculos {  
    private int passageiros;  
    private int combustivel;  
    private int consumo;  
}
```

Membro público (public): todos os módulos tem acesso

Membro protegido (protected): membros do mesmo pacote tem acesso

Membro privado (private) : apenas a própria classe tem acesso

Como criar um objeto de uma classe?



Como criar um objeto de uma classe?

Construtores

- Um construtor é o método que inicializa (cria) um objeto
- Tem o mesmo nome que sua classe
- É similar a uma função porém não tem tipo de retorno explícito

Como colocar valores nos atributos

Construtores

- No exemplo anterior o construtor criou o objeto vazio, os atributos passageiros, Combustível consumo ainda estão “zerados”
- Eu posso atribuir diretamente os valores?

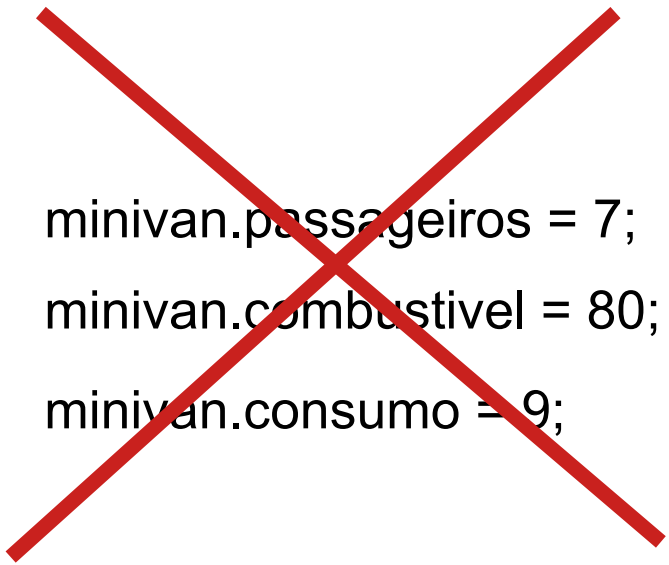
`minivan.passageiros = 7;`

`minivan.combustivel = 80;`

`minivan.consumo = 9;`

Como colocar valores nos atributos

- Eu posso atribuir diretamente os valores?



```
minivan.passageiros = 7;  
minivan.combustivel = 80;  
minivan.consumo = 9;
```

NÃO!!!!!!

Os construtores também serão usados para dar valores iniciais as variáveis de instância definidas pela classe!

Métodos

- Métodos são procedimentos ou funções, dependendo do seu escopo (se retornam valores ou não)
- Precisamos de métodos para buscar os valores dos atributos e para atualizar valores dos atributos se necessário (em Java os gets e sets)
- Métodos são operações que devem ser executadas sobre os dados
- Os métodos devem ser operados sobre a sua própria classe, dessa forma os dados podem ser privados (apenas seus métodos os acessam)

Métodos

- Esse conceito é importante para implementações 2 e 3 camadas, onde a interface está separada da lógica da aplicação (uma dos objetivos da orientação a objetos)
- As mensagens são as chamadas dos métodos, ou seja, chamar um procedimento (paradigma estruturado) é equivalente a passar uma mensagem para um método (paradigma OO)

Exemplo JAVA

Veículos com Construtor e Métodos

```
public class Veiculos {  
    private int passageiros;  
    private int combustivel;  
    private int consumo;  
  
    public Veiculos(int pass, int comb, int cons){  
        this.passageiros = pass;  
        this.combustivel = comb;  
        this.consumo = cons;  
    }  
  
    public int calcular_consumo(){  
        return (this.combustivel*this.consumo);  
    }  
  
    public int get_passageiros(){  
        return this.passageiros;  
    }  
}
```

Exemplo JAVA

Veículos com Construtor e Métodos

```
public static void main(String[] args) {  
    // TODO code application logic here  
    Veiculos minivan = new Veiculos(7,80,9);  
    Veiculos sportcar = new Veiculos(2,9,35);  
    System.out.print("Minivan pode transportar " + minivan.get_passageiros() + " passageiros ");  
    System.out.println("com uma autonomia de " + minivan.calcular_consumo() + " quilômetros");  
    System.out.print("Carro esporte pode transportar " + sportcar.get_passageiros() + "  
passageiros ");  
    System.out.println("com uma autonomia de " + sportcar.calcular_consumo() + " quilômetros");  
}  
}
```

Minivan pode transportar 7 passageiros com uma autonomia de 720 quilômetros
Carro esporte pode transportar 2 passageiros com uma autonomia de 315 quilômetros