

LINGUAGENS DE PROGRAMAÇÃO

Engenharia Informática

Nesta Aula...

■ Introdução OO

- *História*
- *P OO*
- *Objects*
- *Classes*

■ Conceitos básicos

- *Encapsulamento - Encapsulation*
- *Herança - Inheritance*
- *Abstracção - Abstraction*
- *Polimorfismo - Polymorphism*

Computer programming

- Um **algoritmo** é um processo ou um conjunto de regras que no seu conjunto resolvem um problema.
- Um **programa de computador** executa passo a passo um conjunto de instruções de um computador.
 - *O algoritmo é o raciocínio por detrás da execução*

Computer programming

– Um Pouco de História

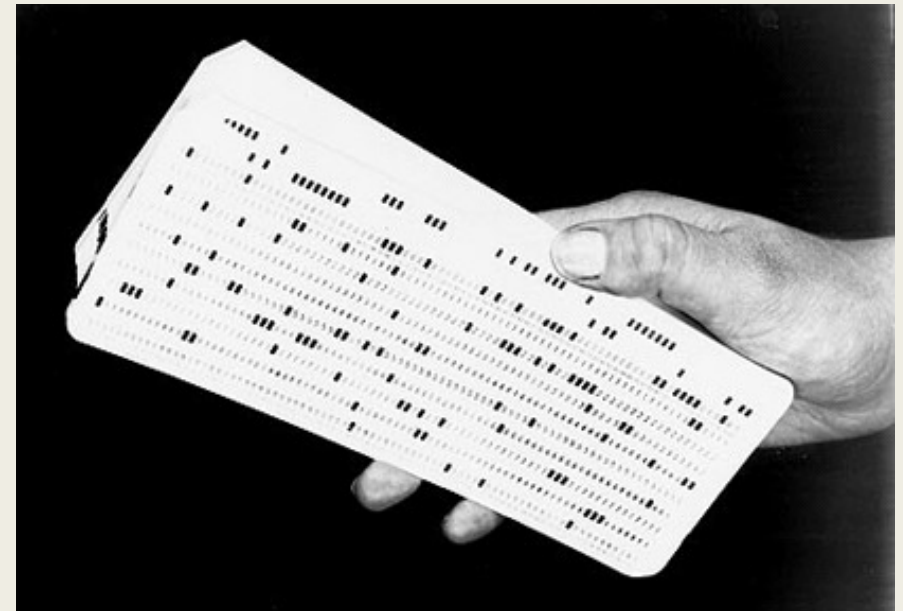
- A programação começou por estar intimamente ligada à arquitectura computacional da máquina (instruction set).
 - *Instruções*
 - *Relógio*
 - *Memória*
 - *Etc...*
- Com o tempo, a programação e as linguagens aproximaram-se mais da forma como nós (programadores) interagimos com o mundo à nossa volta e não tanto para que registo tenho de copiar.

Linguagens de Programação

- A Linguagem é a forma com que descrevemos a nossa intenção e criamos os algoritmos e objectivos do programa.
- Níveis de abstração:
 - *Linguagem máquina: 00011101110011...*
 - *Assembly: “mov ebx, count”*
 - *Linguagens alto nível: “for (int i=0; i<2; i++)...”*

Linguagem máquina

- Traduzida em 0's e 1's
- Compõe as instruções que alimentam sequencialmente a unidade de processamento
- Difícil de programar...



Assembly

- Um passo para melhorar a programação
- Introduz um conjunto de comandos que correspondem a códigos de instruções máquina.
- Tem de ser traduzido em linguagem máquina antes da execução.

Linguagens de alto nível

- Um salto grande em direcção à abstracção necessária.
 - *A sintaxe aproxima-se da linguagem humana (Inglês):
for, switch, while, increment += 1;*
- Podem ser divididas em 2 grandes grupos:
 - *Procedimental*
 - *Orientada a objectos*

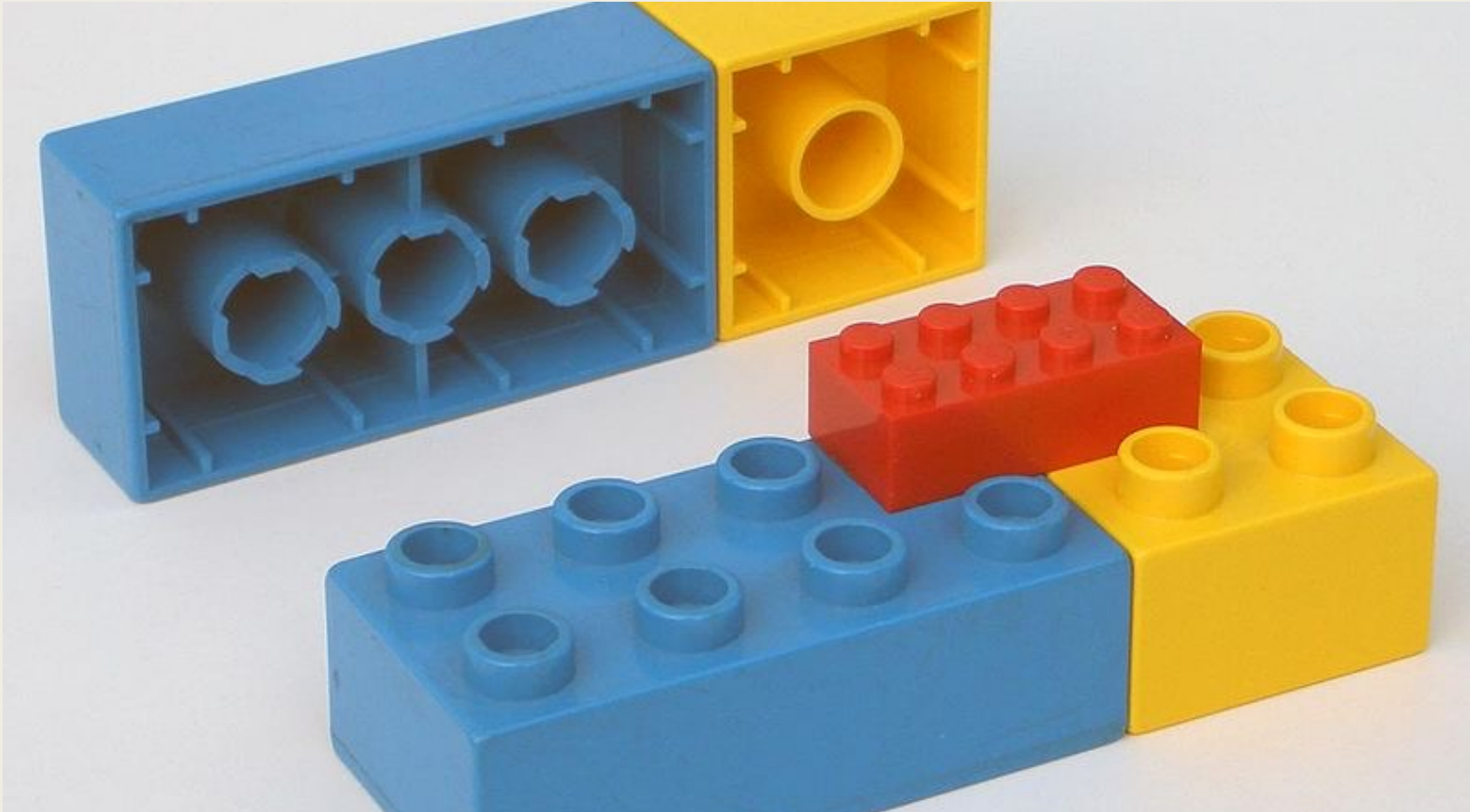
Linguagem Procedimental

- Conjuntos sequenciais de comandos lineares
 - *Agrupados em funções (call & return)*
- Programação Modular e estruturada.
- O foco deste tipo de linguagens é a estrutura do programa e dos dados.
- Exemplos:
 - *C, Pascal, Fortran...*

Linguagem Orientada a objectos

- Linguagens POO não têm o foco na estrutura, mas na forma como os dados são modelados e agrupados:
 - *Modelos... templates... Abstração...*
- Exemplos:
 - *C++, Java, Python...*

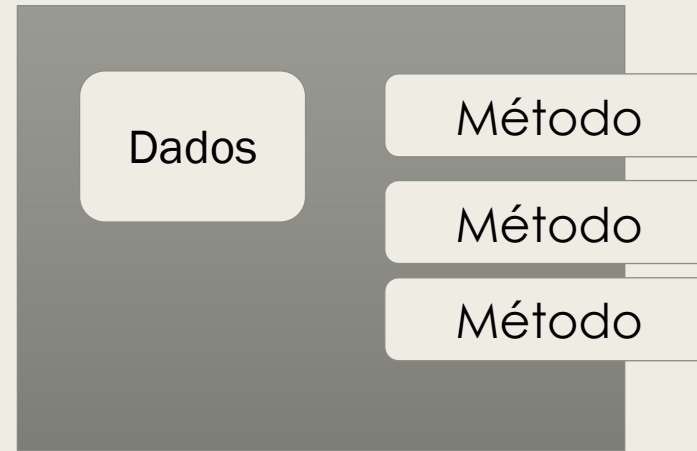
P00



P00

- Assenta numa filosofia de design específico
 - *Em Linguagens procedimentais:*
 - Criar estruturas de dados e Alocação em memória.
 - Criar procedimentos para manipular esses dados.
 - *Em POO:*
 - Instanciar objectos que podem conter instanciação de dados com procedimentos específicos para manipular esses dados.
 - Tudo é agrupados em objectos auto-suficientes.
 - Re-usabilidade a vários níveis

P00



- Em POO os programadores definem não só a estrutura de dados, mas também as operações / métodos / funções que se relacionam e manipulam esses dados.
- É possível criar relações entre objectos
- Objectos podem inclusivamente “herdar” características de outros objectos.

Linguagens P00

■ Puras

- *Todos os dados são objectos*
- *Eiffel, Actor, Ruby, Python, Smaltalk,...*

■ Híbridas

- *Java, C++, C#, VB.Net*

P00 - Terminologia

■ Objecto

- *Instanciação/Materialização do conceito - Classe.*
- *Normalmente em associação ao mundo real que se pretende modelar. (Nome)*

■ Método

- *Acção executada por um objecto. (Verbo)*

■ Propriedade ou atributo

- *Característica de um determinado objecto.*

■ Classe

- *Protótipo onde se insere um conjunto de objectos.*
- *Não armazena propriedade/atributos dos objectos.*

Car
Objects



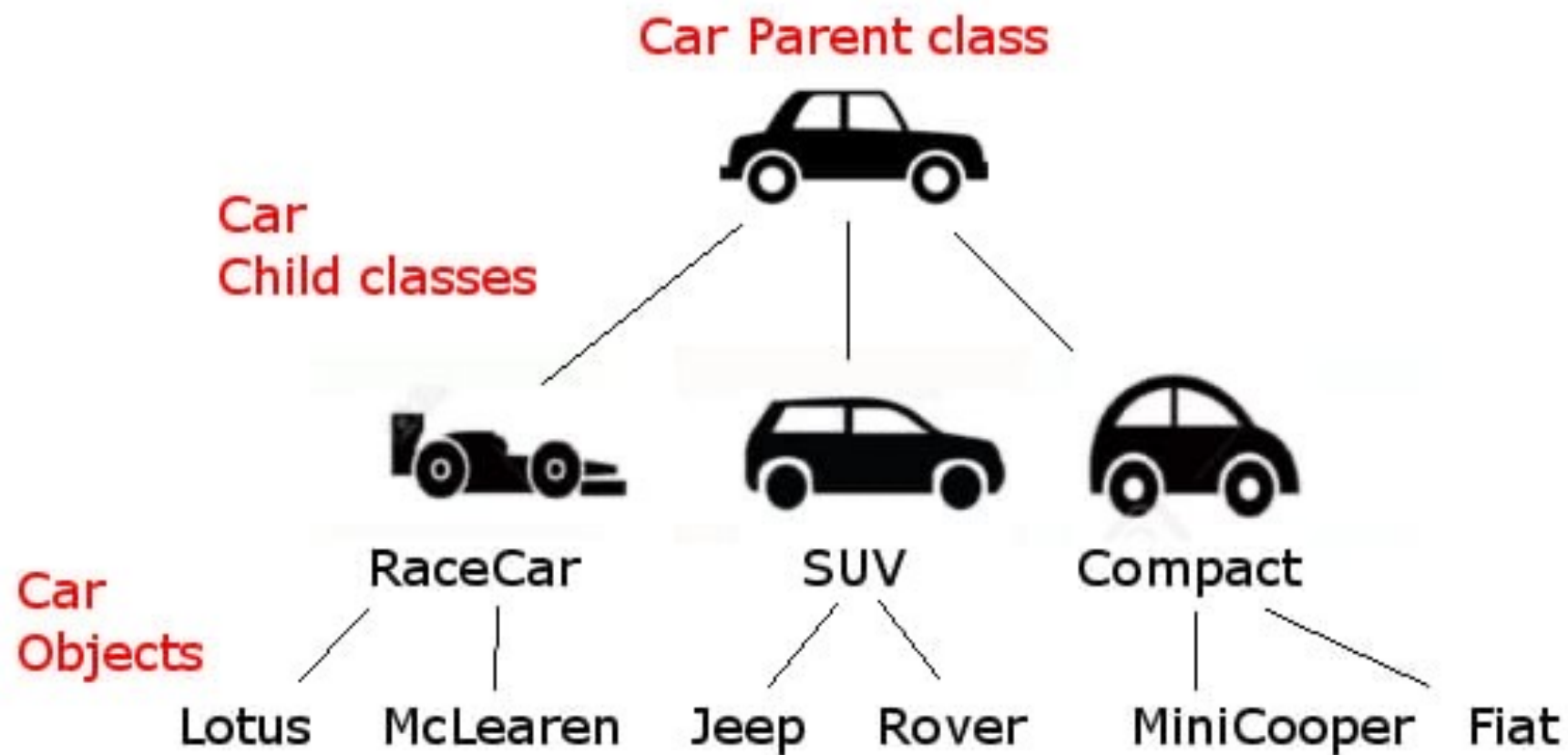
Green
Ford
Mustang



Red
Toyota
Prius



Blue
Volkswagon
Golf



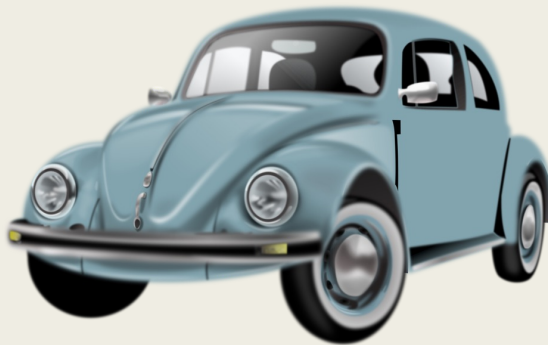
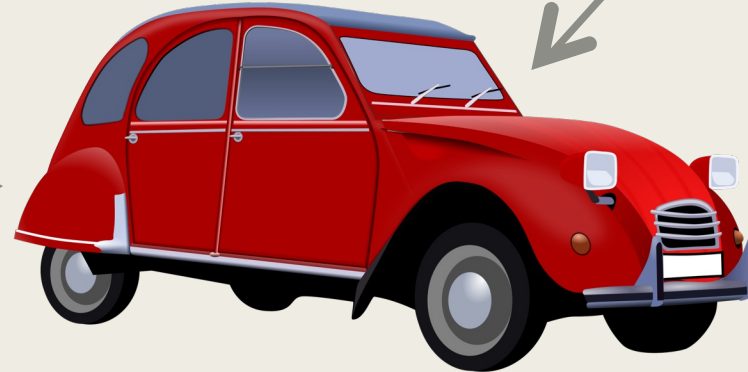
Classe - exemplo

Carro	Nome da Classe
Nome Marca Cor Motor	Atributos
Acelera() Trava()	Métodos

Classe: Carro

Nome: 2CV
Marca: Citroën
Cor: Vermelho

Acelera()
Trava()



Terminologia

- Mensagem

- Chamada a um objecto para invocar um dos seus métodos

- Herança

- Mecanismo pelo qual uma classe (sub-classe) pode incorporar (herdar de) uma outra classe (super-classe), aproveitando atributos e métodos.

- Polimorfismo

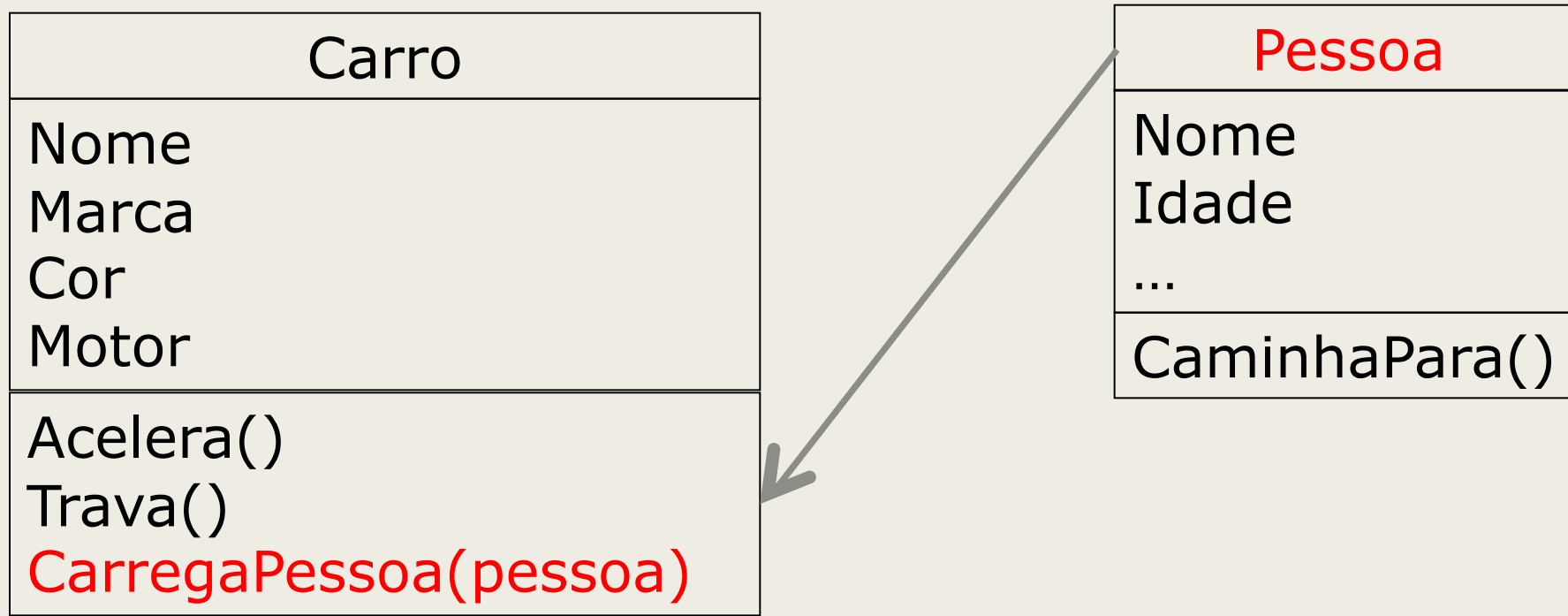
- Mecanismo pelo qual um método herdado pode-se comportar de forma diferente em classes diferentes.
 - Ferrari e 2CV Aceleram e Travam de forma diferente.

- Encapsulamento

- Mecanismo pelo qual se consegue conter a abrangência de um determinado atributo/método dentro do objecto. “Esconder” a complexidade.
 - O atributo Motor é uma propriedade interna que não tem de ser exposta para fora, mas é necessário para os métodos Acelera().

■ Associação

- *Mecanismo pelo qual um objecto utiliza recursos de outro.*



Fim

- Questões?