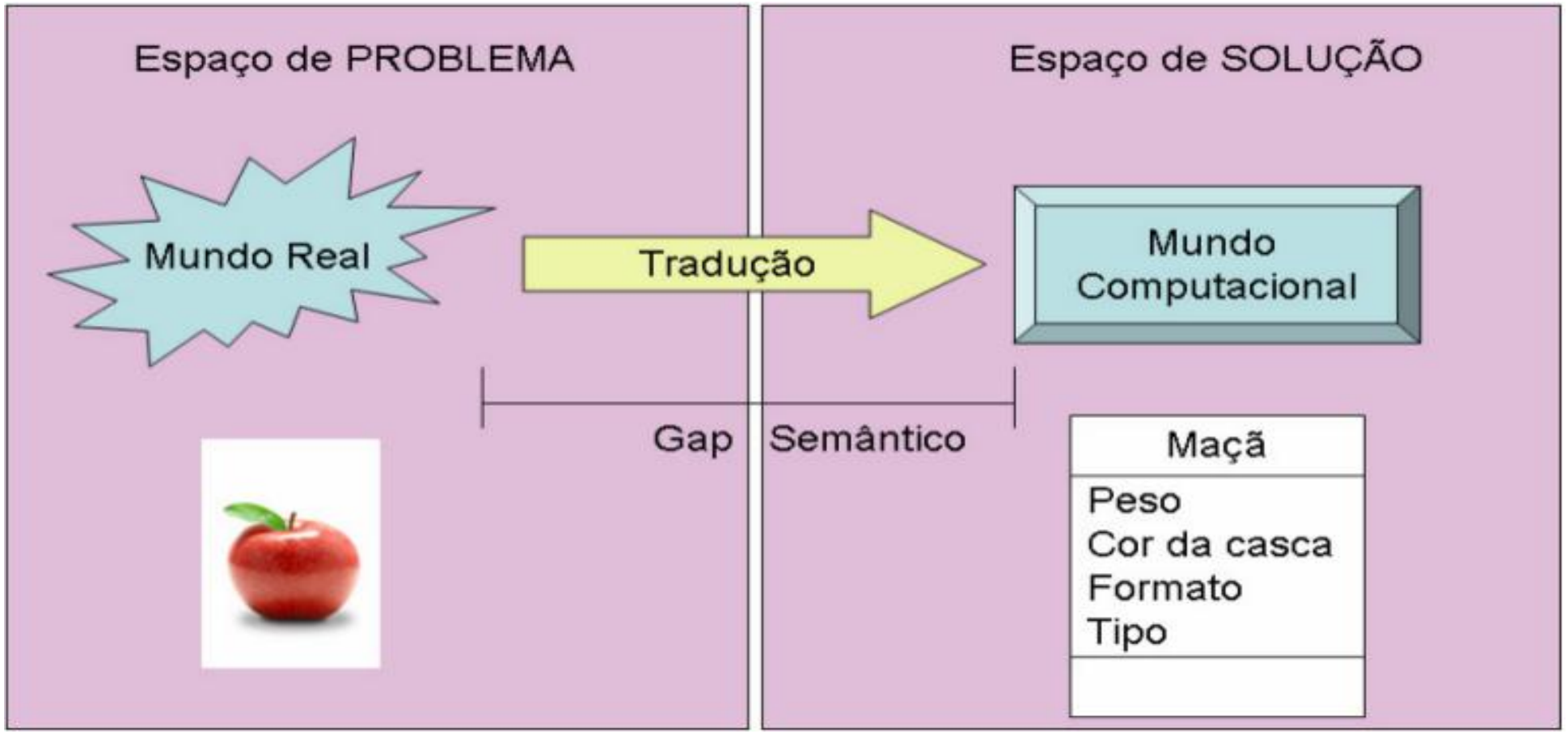
Programação Orientada à Objetos:

O propósito da Orientação à Objetos



Programação Orientada à Objetos:

Orientação à Objetos

É um paradigma para o desenvolvimento de software que basea-se na utilização de componentes individuais (objetos) que colaboram para construir sistemas mais complexos.

Programação Orientada à Objetos:

Objetos

É a metáfora para se compreender a tecnologia orientada a objetos;

Tudo neste mundo são objetos e estamos rodeados por eles: carro, pessoa, mesa, arvore, livro, ...

Os Objetos tem duas características em comum:

- **Estado** - representa as propriedades (nome, altura, peso,cor)
- **Comportamento**:representa as ações (andar, falar, abrir, cortar)

Programação Orientada à Objetos:

Nome: Fusca
cor: Preto
ano: 1965
veloc.max: 80
veloc.atual:0
estado: desligado

← Estados

ligar()
desligar()
acelerar()
parar()

← Comportamentos

Programação Orientada à Objetos:

Os quatro pilares da POO:

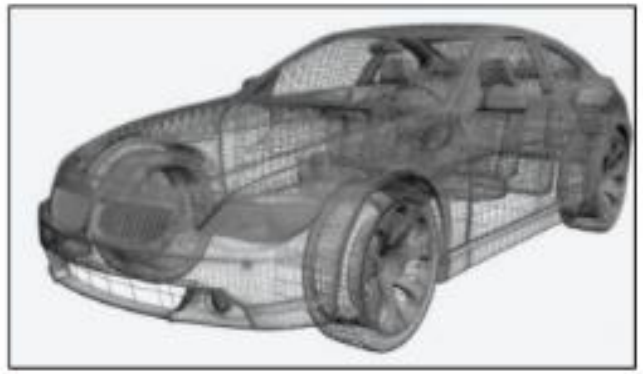
- Abstração
- Encapsulamento
- Herança
- Polimorfismo

Programação Orientada à Objetos:

Abstração

- Habilidade de se concentrar nos aspectos essenciais do sistema, ou um contexto qualquer, ignorando o que é supérfluo.
- Modelo, visão simplificada de algo, onde apenas elementos relevantes são considerados.
- Ex: Mapas

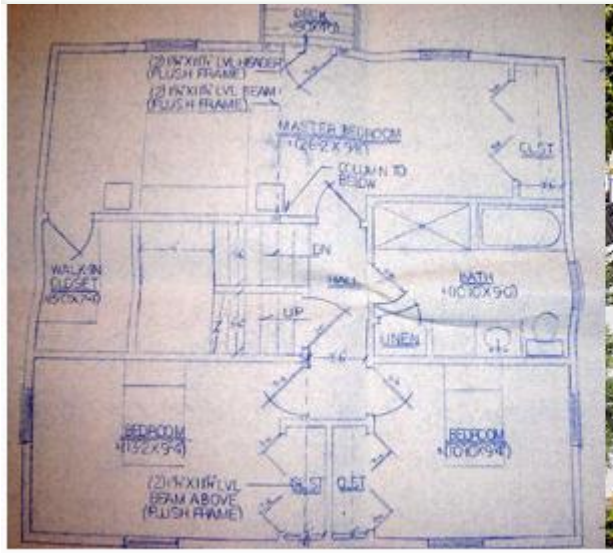
Programação Orientada à Objetos: Classes x Objetos



Classe

Objeto

Você dirige o carro ou o projeto do carro?



Classe

Objeto

Você mora na casa ou na planta da casa?

Qual a importância das Classes na Programação Orientada à Objetos?

Programação Orientada à Objetos: Classes x Objetos

- **Conclusão???**
- A estrutura fundamental para definir novos objetos é a classe.
- Uma **Classe** é uma especificação para um conjunto de objetos (características + comportamentos). Descrição genérica dos objetos individuais pertencentes a um dado conjunto

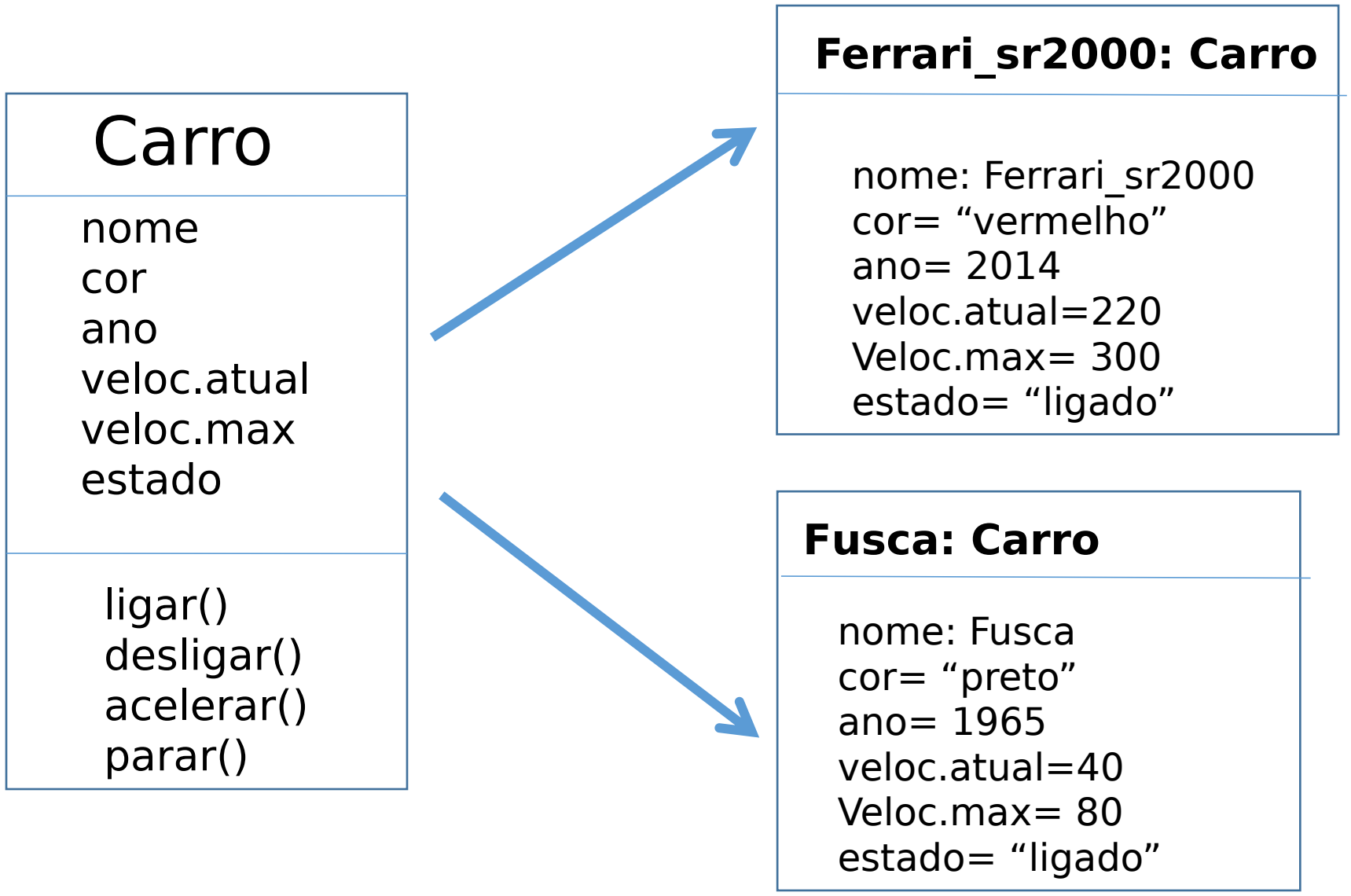
- Um **objeto** é a representação concreta (instância) de uma classe. É capaz de armazenar estados por meio de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.

Programação Orientada à Objetos: Classes x Objetos

Objetos

- **Estado (valores dos Atributos)**
  - Cor = vermelho; Ano=2014; Portas=2; velocidade_atual=0; velocidade_max=300;
- **Comportamento (métodos)**
  - Ao dirigir um carro, o ato de pressionar o acelerador envia uma mensagem para o carro realizar uma tarefa.
- **Identidade**: É a propriedade do objeto que o distingue de outros objetos.

Programação Orientada à Objetos: Classes x Objetos



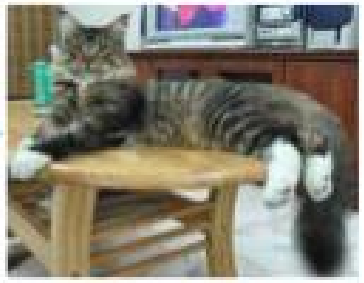
Programação Orientada à Objetos: Classes x Objetos

Gato

New

Gato
Raça: Savannah
Nome: Gatuno
Peso: 2,5 quilos
Idade: 2 anos

New



Gato
Raça: Maine Moon
Nome: Listrado
Peso: 3 quilos
Idade: 5 anos

New

Gato
Raça: Siamês
Nome: Bichano
Peso: 4 quilos
Idade: 3 anos

Como você definiria a classe Gato?

Gato
Raça Nome Peso Idade
engordar() envelhecer() mudar_nome()

Programação Orientada à Objetos: Classes x Objetos

Classe em Python

Estrutura

```
class nome_da_classe:
    atributos
    construtor
    métodos
```

Exemplo em Python:

```
1    class Gato:
2        raça  = None
3        nome  = None
4        peso  = None
5        idade = None
```

Programação Orientada à Objetos: Classes x Objetos

Instancia (objeto)

- Uma instância é um objeto criado com base em uma classe definida;
- Classe é apenas uma estrutura, que especifica objetos, mas que não pode ser utilizada diretamente;
- Instância representa o objeto concretizado a partir de uma classe;
- Uma instância possui um ciclo de vida (escopo):

  Criação ➡ Manipulação ➡ Destruição.

Programação Orientada à Objetos: Classes x Objetos

Estrutura Python
variável = Classe()

Exemplo:

```
7   mimi=Gato()
8   mimi.nome="mimi"
9   mimi.peso=2
10  mimi.idade=1
11  mimi.raça="sem raça"
```

Programação Orientada à Objetos: Classes x Objetos

Métodos

- Representam os comportamentos de uma classe;
- Premitem que acessemos os atributos, tanto para recuperar os valores, como para alterá-los caso necessário;
- Podem retornam ou não algum valor e podem possuir ou não parâmetros.

Programação Orientada à Objetos: Classes x Objetos

Sintaxe em Python

Estrutura

```
def nome_do_método(self,parâmetros):
```

Importante

O parâmetro **self** é obrigatório.

Introdução à linguagem Python: Variáveis e Entrada/Saída de Dados

- Exemplo:

```
1   class Gato:
2       raça  = None
3       nome  = None
4       peso  = None
5       idade = None
6
7       def mudar_nome(self,nome):
8           self.nome=nome
9       def engordar(self,peso):
10          self.peso+=peso
11      def envelhecer(self):
12          self.idade +=1
```

Programação Orientada à Objetos: Classes x Objetos

```
14  meu_gato=Gato()
15  meu_gato.nome="mimi"
16  meu_gato.peso=2
17  meu_gato.idade=1
18  meu_gato.raça="sem raça"
19  meu_gato.mudar_nome("tom")
20  meu_gato.engordar(3)
21  meu_gato.envelhecer()
22  print("o nome do meu gato agora é:",meu_gato.nome)
23  print("ele pesa atualmente:{} quilos".format(meu_gato.peso))
24  print("sua idade atual é {} anos".format(meu_gato.idade))
```

Programação Orientada à Objetos: Classes x Objetos

Exercícios