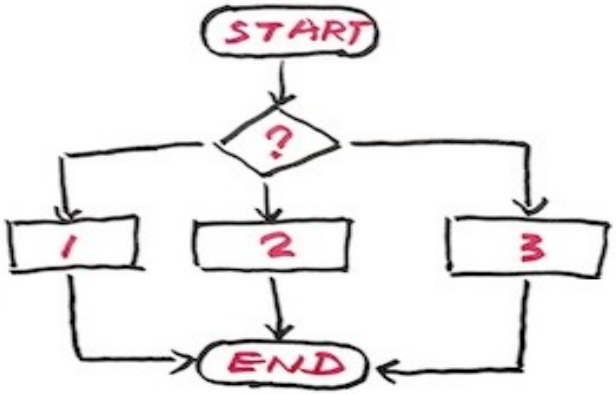
Introdução à Programação



Aula 16

Classe

Prof. Dr. Ivan José dos Reis Filho

Aula anterior

Funções

Aula de hoje

Classes

- Classes
- Objetos
- Atributos

O que é uma classe?

Classe

Uma classe define um conjunto de objetos com comportamento, através de métodos, e quais estados ele é capaz de manter, através de atributos.

Por exemplo:

Qual classe esses objetos pertencem?

Qual classe esse objeto pertence?

Qual classe esse objeto pertence?

Objetos

Um objeto é uma instância de uma classe.

Para diferenciar um objeto de outro, é preciso criar os ATRIBUTOS.

Atributos/Propriedade

Quais são os atributos de um carro?

Atributos

Quais são os atributos de um carro?

Carro
Cor Marca Modelo Ano

Atributos

São dados ou informações do objeto, basicamente a estrutura de dados que representa a classe.

Exemplos de classes: atributos

Funcionário: nome, end, fone, CPF;

Carro: nome, marca, ano, cor;

Livro: autor, editora, ano;

Métodos

Definem as habilidades dos objetos.

Exemplo:

Quais são as ações de um automóvel?

Métodos

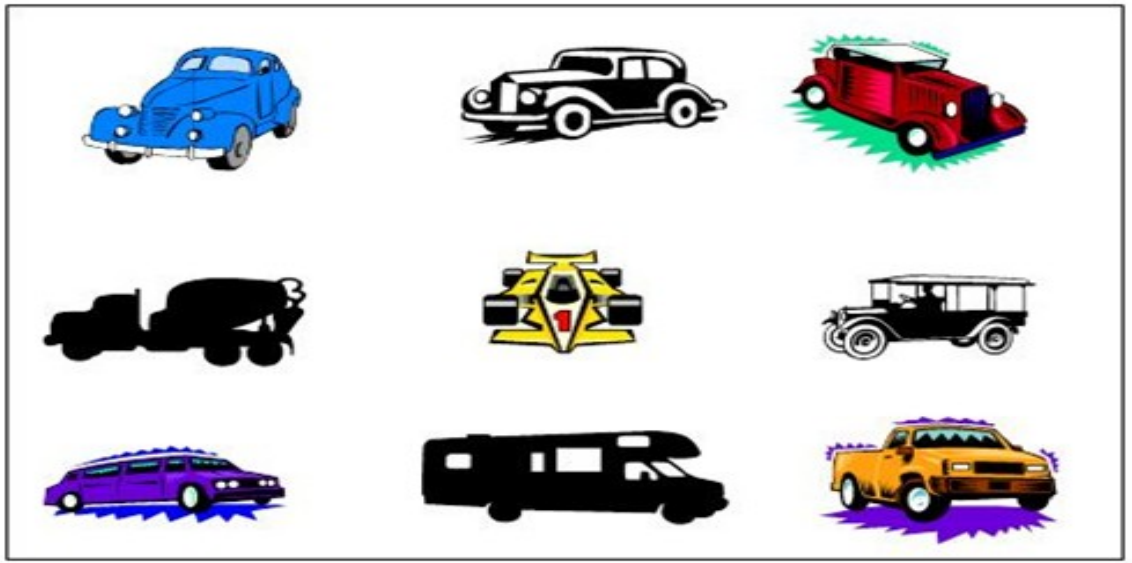
Definem as habilidades dos objetos.

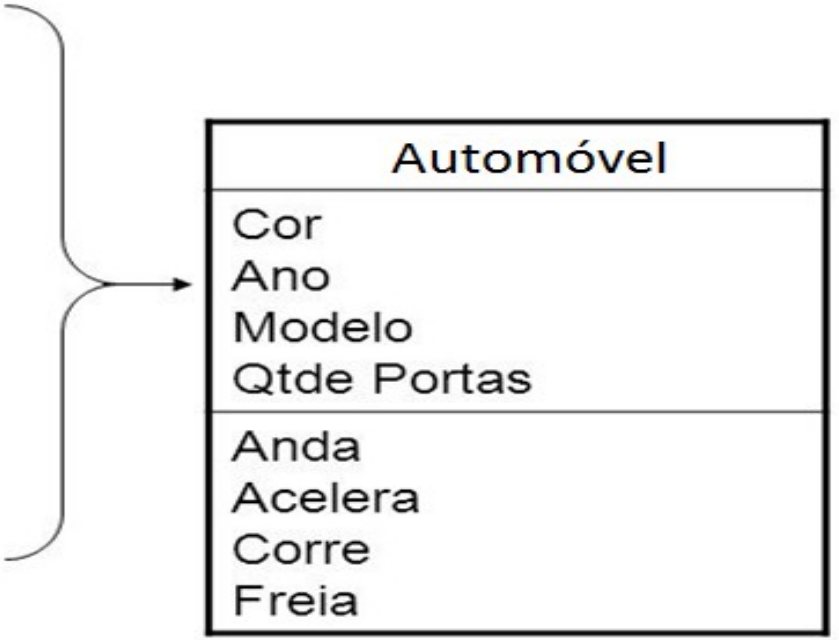
Exemplo:

Acelera, Freia, Anda, Corre e etc...

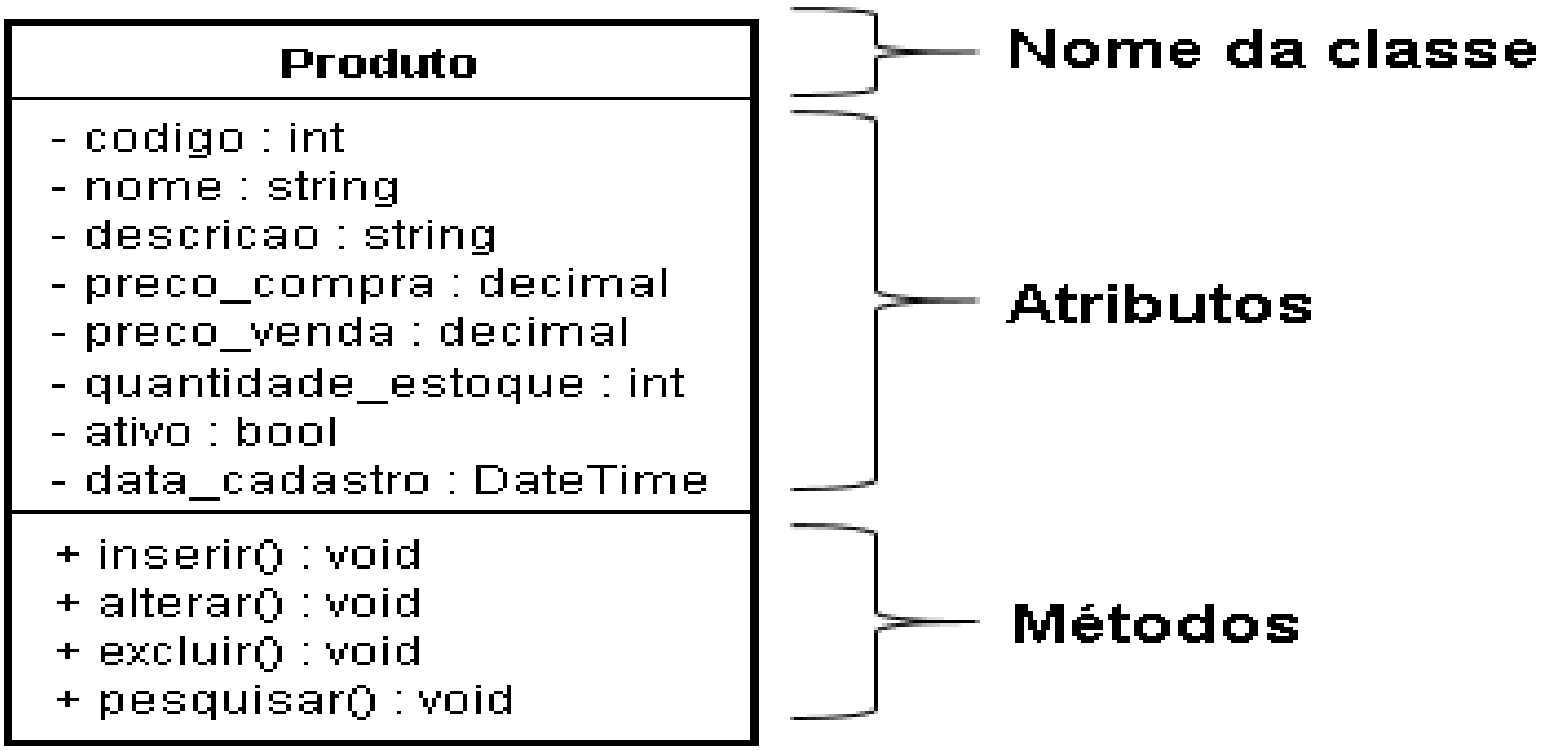
Quais são as ações de um automóvel?

Representação de uma Classe





Representação de uma Classe



Classes e atributos

Classe

Criação

Classe chamada MyClass, com uma propriedade/atributo **x**.

Example

Create a class named MyClass, with a property named x:

```
class MyClass:
  x = 5
```

Classe

Criação

Create a class named MyClass, with a property named x:

```
class MyClass:
  x = 5
```

Criando um **objeto** chamado p1 e exibindo o valor de x

Example

Create an object named p1, and print the value of x:

```
p1 = MyClass()
print(p1.x)
```

Classe

The __init__( ) Function

Os exemplos anteriores são classes e objetos na sua forma mais simples, não são úteis para aplicações reais

- Entender o significado de classes é preciso entender a função __init__();
- Toda classe possui um função chamada __init__(), que é sempre executada quando a classe é iniciada;
- Use a função para atribuir valores nas propriedade/atributos do objeto;
- Ou, operações que são necessariamente necessárias quando o objeto é criado.

Classe

The __init__( ) Function

Exemplo: Criar uma classe chamada Person, use a função __init__() para atribuir valor em name e age (idade)

Create a class named Person, use the __init__() function to assign values for name and age:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Note: The `__init__()` function is called automatically every time the class is being used to create a new object.

Exercícios

Aula15Exs.py

1. Crie uma classe pessoa com atributos *nome*, *sexo*, *dt_nascimento* e *rg*. Crie dois objetos (pessoas) com dados diferentes e exibe no programa principal.

2. Crie uma classe moto com atributos *modelo*, *ano*, *cor* e *cilindrada*. Crie dois objetos (moto) com dados diferentes e exibe no programa principal os dados informados.

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Método

Classe

Método

Objetos podem também conter métodos. Métodos são funções que pertencem ao objeto.

- Uma função que exibe uma saudação e executa no objeto p1

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)

p1 = Person("John", 36)
p1.myfunc()
```

Note: The `self` parameter is a reference to the current instance of the class, and is used to access variables that belong to the class.

Exercícios

Continuação

Exercícios

Continuação

1. Crie uma classe pessoa com atributos *nome*, *sexo*, *dt_nascimento* e *rg*. Crie dois objetos (pessoas) com dados diferentes e exibe no programa principal.

a) Crie um método de saudação e use os objetos para chama-la.

b) Crie um método de despedida e use os objetos para chama-la.

2. Crie uma classe moto com atributos *modelo*, *ano*, *cor* e *cilindrada*. Crie dois objetos (moto) com dados diferentes e exibe no programa principal.

a) Crie um método de partida (ligar a moto) e use os objetos para chama-la.

b) Crie um método de desliga (desligar a moto) e use os objetos para chama-la.

Classe

Parâmetro self

O Parâmetro self é uma referência para a instância atual da classe. Usada para acessar variáveis que pertence a classe.

- Não precisa ser nomeado como self;
- Deve ser o primeiro parâmetro da função da classe.

Use the words *mysillyobject* and *abc* instead of *self*:

```
class Person:
  def __init__(mysillyobject, name, age):
    mysillyobject.name = name
    mysillyobject.age = age

  def myfunc(abc):
    print("Hello my name is " + abc.name)

p1 = Person("John", 36)
p1.myfunc()
```

Exercícios

Continuação

Exercício

Continuação

1. Crie uma classe pessoa com atributos *nome*, *sexo*, *dt_nascimento* e *rg*. Crie dois objetos (pessoas) com dados diferentes e exibe no programa principal.

a) Use o parâmetro self

2. Crie uma classe moto com atributos *modelo*, *ano*, *cor* e *cilindrada*. Crie dois objetos (moto) com dados diferentes e exibe no programa principal.

a) Use o parâmetro self

Modificando propriedades

Classes

Modificando propriedades dos objetos

Você pode modificar propriedade do objeto

Set the age of p1 to 40:

```
p1.age = 40
```

Você pode deletar propriedade do objetos usando a palavra reservada del

Delete the age property from the p1 object:

```
del p1.age
```

Você pode deletar os objetos usando o del

Delete the p1 object:

```
del p1
```

Exercícios

Continuação

Exercícios

Continuação

1. Crie uma classe pessoa com atributos *nome*, *sexo*, *dt_nascimento* e *rg*. Crie dois objetos (pessoas) com dados diferentes e exibe no programa principal.

a) Solicite para o usuário informar os dados. O algoritmo deve atribuir os dados nos atributos da classe.

b) Depois de informar os dados, exibe a saudação e despedida.

2. Crie uma classe moto com atributos *modelo*, *ano*, *cor* e *cilindrada*. Crie dois objetos (moto) com dados diferentes e exibe no programa principal.

a) Solicite para o usuário informar os dados. O algoritmo deve atribuir os dados nos atributos da classe;

b) Depois de informar os dados, mostre que a moto foi ligada e em seguida desligada.

Exercícios

Entregar no Teams

Aula16_SeuNome.py

Continuação

Ex01: Crie uma classe chamada "Cachorro" com os atributos "nome" e "idade". Em seguida, crie um método chamado "latir" que imprima uma mensagem dizendo "O cachorro está latindo!".

Ex02: Crie uma classe chamada "Retângulo" com os atributos "largura" e "altura". Em seguida, crie um método chamado "calcular_area" que retorna a área do retângulo (largura * altura).

Ex03: Crie uma classe chamada "ContaBancaria" com os atributos "titular" e "saldo". Em seguida, crie métodos chamados "depositar" e "sacar" que permitam adicionar ou retirar uma determinada quantia do saldo da conta.

Ex04: Crie uma classe chamada "Carro" com os atributos "marca", "modelo" e "ano". Em seguida, crie um método chamado "descrição" que retorna uma string contendo as informações do carro.

Ex05: Crie uma classe chamada "Estudante" com os atributos "nome" e "idade". Em seguida, crie um método chamado "saudacao" que imprima uma mensagem personalizada de saudação para o estudante