

ESCOLA E FACULDADE DE TECNOLOGIA SENAI “ROBERTO MANGE”

Rebeca Bugati Preza

PBE: Pesquisa POO

Campinas

2025

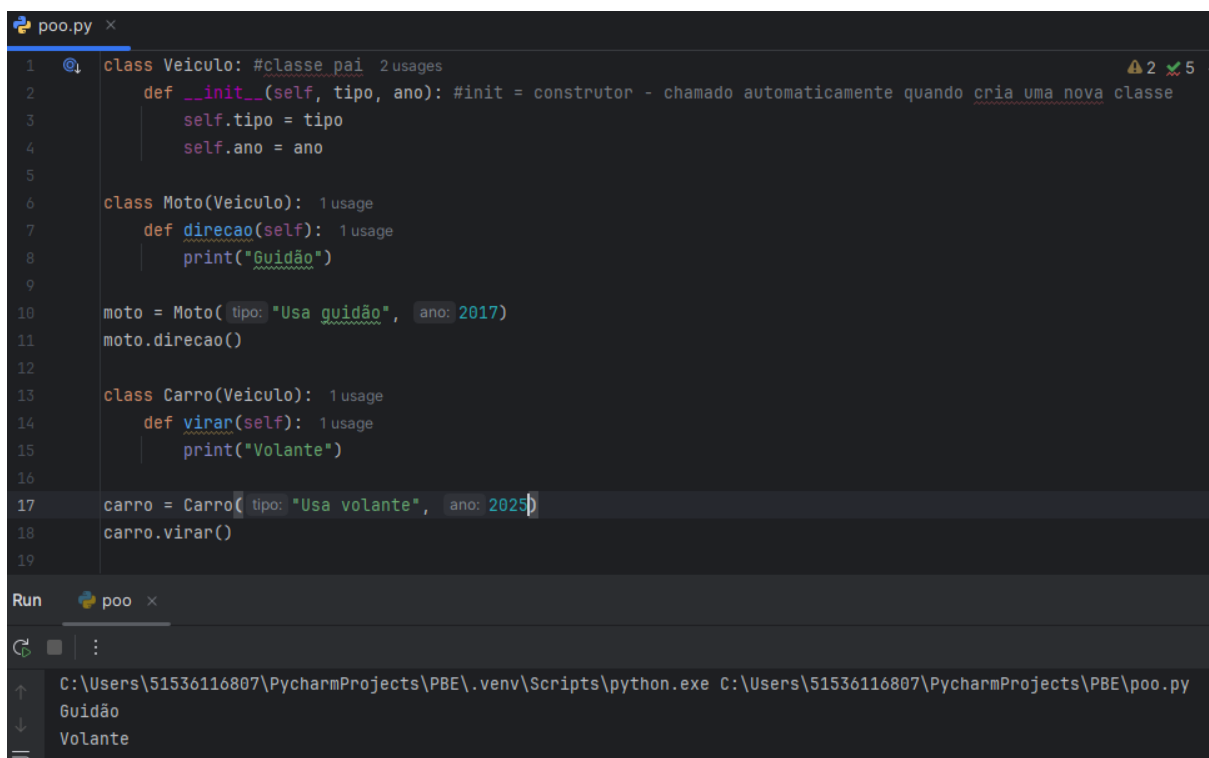
Herança (POO) - também conhecida como classe pai ou superclasse – permite que uma classe herde de outra classe, todos os comportamentos e atributos. Através dela a classe possui imediatamente todas as funcionalidades de uma classe já existente.

Na classe já existente, chamada de classe pai, onde declara todos os métodos (Modelo geral) a todos os subníveis (generalização - modelo específico) na hierarquia de herança - subclasses. Dentro dessas subclasses, são definidas suas particularidades (especialização (o que fazem de não comum)), que herdam as características da classe pai.

Exemplo:

Termo	O que é	Exemplo
Classe pai	Modelo geral	Veículo
Subclasse	Modelo específico	Carro, moto
Herança	Reaproveita código	Acelerar
Generalização	Coisas em comum	Roda, motor
Especialização	Coisas diferentes	Tem volante, usa guidão

Exemplo na prática:



```
1 class Veiculo: #classe pai 2 usages
2     def __init__(self, tipo, ano): #init = construtor - chamado automaticamente quando cria uma nova classe
3         self.tipo = tipo
4         self.ano = ano
5
6     class Moto(Veiculo): 1 usage
7         def direcao(self): 1 usage
8             print("Guidão")
9
10    moto = Moto(tipo: "Usa guidão", ano: 2017)
11    moto.direcao()
12
13    class Carro(Veiculo): 1 usage
14        def virar(self): 1 usage
15            print("Volante")
16
17    carro = Carro(tipo: "Usa volante", ano: 2025)
18    carro.virar()
19
```

Run

C:\Users\51536116807\PycharmProjects\PBE\.venv\Scripts\python.exe C:\Users\51536116807\PycharmProjects\PBE\poo.py

Guidão

Volante

Polimorfismo (POO) - métodos com o mesmo nome se comportam de maneiras diferentes, dependendo de quem esteja usando.

Exemplo:

- ✓ Método - falar
- Ser humano fala em palavras;
- Cachorro fala latindo;
- Gato fala miando;

Exemplo na prática:

```
21 #Polimorfismo
22 class Animal: 3 usages
23     def falar(self): 1 usage (1 dynamic)
24         print("0 animal faz um som.")
25
26 class Cachorro(Animal): 1 usage
27     def falar(self): 1 usage (1 dynamic)
28         print("0 cachorro diz: Au au!")
29
30 class Gato(Animal): 1 usage
31     def falar(self): 1 usage (1 dynamic)
32         print("0 gato diz: Miau!")
33
34 class Ser_humano(Animal): 1 usage
35     def falar(self): 1 usage (1 dynamic)
36         print("0 humano diz: Oi!")
37
38
39 def fazer_falar(animal): 1 usage
40     animal.falar()
41
42 animais = [Cachorro(), Gato(), Ser_humano()]
43
44 for fala in animais:
45     fazer_falar(fala)
```

Run poo x

```
C:\Users\51536116807\PycharmProjects\PBE\.venv\Scripts\python.exe C:\Users\51536116807\PycharmProjects\PBE\poo.py
0 cachorro diz: Au au!
0 gato diz: Miau!
0 humano diz: Oi!
Process finished with exit code 0
```

Diferença entre Herança e polimorfismo – A herança permite criar novas classes (subclasses) e classes existentes (superclasses), reutilizando código. Já o polimorfismo permite que diferentes objetos respondam de forma diferente a uma mesma mensagem (método).

Vantagens - Herança: Através da herança, você pode criar novas classes que herdam atributos e métodos de uma classe base. Isso elimina a necessidade de reescrever código semelhante, promovendo a eficiência e a consistência.

Polimorfismo: O polimorfismo permite que objetos de diferentes classes sejam tratados de maneira uniforme, facilitando a criação de sistemas flexíveis e extensíveis.

Referências: <https://www.dio.me/articles/vantagens-da-programacao-orientada-a-objetos-poo>

https://sae.unb.br/cae/conteudo/unbfga/oo/new_heranca.html

https://cursos.alura.com.br/forum/topico-duvida-sobre-o-__init__-e-self-320937