

Poliformismo de sobreposição

quinta-feira, 27 de outubro de 2022 14:20

Polimorfismo é o princípio que permite que classes derivadas de uma mesma superclasse tenham métodos iguais (de mesma assinatura) mas comportamentos diferentes.

mesma assinatura = mesma quantidade e tipo de parâmetros.

no polimorfismo o que acontece é que criamos uma superclasse com um determinado método e outras classes estão pegando algo dessa classe e usam de forma diferente, porém com mesma assinatura (parâmetros).

Ex:

```
from abc import ABC, abstractmethod

class A(ABC):
    def fala(self, msg):
        pass

class B(A):
    def fala(self, msg):
        print(f"Estou falando {msg}")

class C(A):
    def fala(self, msg):
        print(f"Estou falando outra coisa {msg}")

b = B()
c = C()

b.fala('estou aqui')
c.fala('estou aqui agora')
```

```
b.fala( 'estou aqui' )  
c.fala('estou aqui agora')
```

Podemos também usar com abstractmethod.