

Classes Abstratas

terça-feira, 25 de outubro de 2022 18:59

Uma classe abstrata é como você liga uma classe a outra por outra classe, isso é uma classe abstrata, que também tem métodos abstratos.

Método abstrato não tem corpo, não escreve o código dele.

precisamos importar 2 módulos para poder criar

from abc (abstract base classic) import ABC, abstractmethod (decorador) - é parecido com getters e setters

quando criamos uma função e usamos como herança o ABC enquanto não usamos o método abstractmethod conseguimos instanciar a classe, porém quando usamos o abstractmethod conseguimos usar ela.

Ex:

```
from abc import ABC, abstractmethod

class A(ABC):
    @abstractmethod
    def falar(self):
        pass

class B(A):
    def falar(self):
        print("Estou falando B")

a = B()
a.falar()
```

Se instanciar a classe A a alguma instancia encontramos um erro.

Ex:

```
from abc import ABC, abstractmethod

class A(ABC):
    @abstractmethod
    def falar(self):
        pass

class B(A):
    def falar(self):
        print("Estou falando B")

a = A()
a.falar()
```

TypeError: Can't instantiate abstract class A with abstract method falar

quando colocamos um dos metodos de abstração automaticamente a classe fica abstrata.