Fatec
Rio Claro

Linguagem de Programação II

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Decorator

Decorators

Fatec
Rio Claro

Decoradores de função nos permitem "marcar" funções no código-fonte, para aprimorar de alguma forma seu comportamento. É um mecanismo muito poderoso.

Por exemplo, o decorador **@functools.cache** armazena um mapeamento de argumentos para resultados, e depois usa esse mapeamento para evitar computar novamente o resultado quando a função é chamada com argumentos já vistos. Isso pode acelerar muito uma aplicação.

Decorators

Fatec
Rio Claro

Um decorador é um invocável que recebe outra função como um argumento (a função decorada).

Um decorador pode executar algum processamento com a função decorada, e pode devolver a mesma função ou substituí-la por outra função ou objeto invocável.

Decorators

Fatec
Rio Claro

Em outras palavras, supondo a existência de uma função decoradora chamada decorate, este código:

```
@decorate
def target():
    print('running target()')
```

Tem o mesmo efeito de:

```
def target():
    print('running target()')

target = decorate(target)
```

Decorators

Fatec
Rio Claro

O resultado final é o mesmo: após a execução de qualquer um destes exemplos, o nome target está vinculado a qualquer que seja a função devolvida por decorate(target)—que tanto pode ser a função inicialmente chamada target quanto uma outra função diferente.

Prática

Fatec
Rio Claro

```
>>> def deco(func):
...     def inner():
...         print('running inner()')
...     return inner  (1)
...
>>> @deco
... def target():  (2)
...     print('running target()')
...
>>> target()  (3)
running inner()
>>> target  (4)
<function deco.<locals>.inner at 0x10063b598>
```

Decorators

Fatec
Rio Claro

Estritamente falando, decoradores são apenas açúcar sintático. Como vimos, é sempre possível chamar um decorador como um invocável normal, passando outra função como parâmetro. Algumas vezes isso inclusive é conveniente, especialmente quando estamos fazendo metaprogramação—mudando o comportamento de um programa durante a execução.

Três fatos essenciais sobre decoradores:

Um decorador é uma função ou outro invocável.

Um decorador pode, opcionalmente, substituir a função decorada por outra.

Decoradores são executados assim que um módulo é carregado.

Decoradores na biblioteca padrão

Decorators

Fatec
Rio Claro

Python tem três funções embutidas projetadas para decorar métodos:

property

classmethod

staticmethod

Property

Fatec
Rio Claro

As propriedades também são usadas para impor regras de negócio, transformando um atributo público em um atributo protegido por um getter e um setter, sem afetar o código cliente.

Property

Fatec
Rio Claro

As propriedades também são usadas para impor regras de negócio, transformando um atributo público em um atributo protegido por um getter e um setter, sem afetar o código cliente.

Prática

Fatec
Rio Claro

Observe os códigos:

bulkfood_v2.py

bulkfood_v2.py

Property

Fatec
Rio Claro

1) Aqui o setter da propriedade já está em uso, assegurando que nenhuma instância com peso negativo possa ser criada.
2) @property decora o método getter.
3) Todos os métodos que implementam a propriedade compartilham o mesmo nome, do atributo público: weight.
4) O valor efetivo é armazenado em um atributo privado __weight.
5) O getter decorado tem um atributo .setter, que também é um decorador; isso conecta o getter e o setter.
6) Se o valor for maior que zero, definimos o __weight privado.
7) Caso contrário, uma ValueError é gerada.

staticmethod

Fatec
Rio Claro

O @staticmethod em Python é um método que pertence a uma classe, mas não depende nem da instância (self) nem da própria classe (cls) para funcionar.

Ele é usado quando você quer agrupar uma função logicamente relacionada à classe, mas que não precisa acessar ou modificar nenhum atributo dela.

Em outras palavras, ele se comporta como uma função comum, porém é definido dentro da classe por questões de organização ou semântica.

Por exemplo, em uma classe Pessoa, um método eh_maior_de_idade(idade) pode ser estático, pois apenas realiza um cálculo lógico independente dos atributos da classe ou do objeto.

classmethod

Fatec
Rio Claro

Já o @classmethod é um método que, em vez de receber automaticamente a instância (self), recebe a própria classe (cls) como primeiro parâmetro. Isso permite acessar e modificar atributos de classe ou criar instâncias de forma padronizada.

Ele é útil quando o comportamento deve afetar todas as instâncias da classe, e não apenas uma delas.

Por exemplo, um método mudar_especie(cls, nova_especie) poderia alterar um atributo de classe chamado especie, impactando todas as instâncias criadas a partir dessa classe.

Assim, enquanto o @staticmethod é usado para lógica independente, o @classmethod é usado quando há necessidade de manipular a própria estrutura da classe.

Prática

Fatec
Rio Claro

Observe o código:

pessoa.py

Fatec
Rio Claro

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

Bibliografia

RAMALHO, Luciano. Fluent Python: clear, concise, and effective programming. 2. ed. Sebastopol, CA: O'Reilly Media, Inc., 2022. 1014 p.