



Сдать решение задачи I

Полный балл:	5
Бонусные баллы:	
Ограничение времени:	1 с
Ограничение реального времени:	5 с
Ограничение памяти:	64М

Хрупкий словарь

В банках всего мира используются SQL-базы данных. Основным преимуществом этих баз данных является высокая скорость доступа к данным, а также выполнение принципов ACID (атомарность, согласованность, изолированность, долговечность). Транзакции обязаны удовлетворять этим принципам, иначе любая ошибка может привести к огромным потерям со стороны банка.

В этом задании предлагается реализовать прототип простейшего "безопасного" хранилища. Наше хранилище будет называться «хрупким словарем». Класс `FragileDict` должен реализовывать следующий интерфейс:

- Инициализатор опционально может принимать на вход словарь, содержимое которого будет храниться в «хрупком словаре». В конструкторе заполняются два атрибута: `_data` — хранилище с данными, `_lock` — булев флаг, показывающий разрешено ли редактировать хранилище. Других полей быть НЕ должно.
- Из любого экземпляра класса `FragileDict` можно читать данные аналогично тому, как это делается для словарей, например, `d['key']`. Если ключ (`key`) отсутствует, то бросается исключение `KeyError(key)`, как если бы это был просто словарь.
- Класс `FragileDict` должен поддерживать механизм менеджера контекста.
- В экземпляры класса можно записывать данные аналогично тому, как это делается для словарей, например, `d['key'] = value`. Однако разрешается это делать только внутри контекста. Если этот контракт нарушается, то бросается исключение `RuntimeError("Protected state")`. При входе в контекст разрешается создавать любые атрибуты класса, но на выходе из контекста никаких сторонних атрибутов быть не должно.
- Если внутри контекста возникло исключение, то данные не записываются. На выходе из контекста «словарь» должен иметь точно такое же состояние, как и на входе. Само исключение подавляется, и пишется сообщение об ошибке `"Exception has been suppressed."`.
- Класс должен поддерживать проверку наличия ключа в формате `key in d`, где `key` — некоторый ключ, а `d` — экземпляр класса «хрупкого словаря».

Примеры

Входные данные

```
d = FrangibleDict({'key': 5})

with d:
    d['key'] = 6
    d['ord'] = 7

print(d['key'])
print(d['ord'])
```

Результат работы

```
6
7
```

Входные данные

```
d = FrangibleDict({'key': 5})

try:
    d['key'] = 6
except RuntimeError as e:
    print(e)

try:
    d['ord'] = 7
except RuntimeError as e:
    print(e)

print(d['key'] == 5)
print('ord' not in d)
```

Результат работы

```
Protected state
Protected state
True
True
```

Входные данные

```
d = FrangibleDict({'key': 5})
```

```
d = FragileDict({'key': 6})

with d:
    d['key'] = 6
    print(d['key'])
    d['ord'] = 7
    print('ord' in d and d['ord'] == 7)
    raise Exception()

print(d['key'])
print('ord' not in d)
```

Результат работы

```
6
True
Exception has been suppressed.
5
True
```

Входные данные

```
d = FragileDict({'key': []})

with d:
    a = d['key']
    d['key'].append(10)
    a.append(10)

a.append(10)
print(a == [10, 10, 10] and d['key'] == [10, 10])
```

Результат работы

True

Примечания

Как вы вероятно помните, в Python все объекты передаются по ссылке. Для обеспечения большей безопасности вашего хранилища вам должен пригодиться модуль copy. Вспомните про разницу между поверхностными копированием (shallow copy) и глубоким копированием (deep copy).

Сдать решение

Язык: python3 - Python3 3.6.5

Файл

Choose File

 No file chosen

Отправить!

Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть исходный текст	Посмотреть протокол
2106	510:16:08	2213	I	python3	OK	8	5	Просмотр	Просмотр

- A
- B
- C
- D
- E
- F
- G
- H
- I