

Zadanie I

Dekorator Private

Napisz dekorator `private`, który ma działać w następujący sposób. W instancji `a` klasy finalnej `A`, udekorowanej dekoratorem `@private("attr", "method")`, niedozwolone są odczyty oraz przypisania do atrybutów wymienionych na liście dekoratora.

Wejście

Testy przypisania/odczytania atrybutów wykonywane są tylko na instancjach udekorowanych klasach finalnych. Próby odczytu/przypisania do atrybutów prywatnych mają rzucać wyjątek `TypeError` ze stosownym błędem (patrz test przykładowy).

Dostępna pamięć: 32MB

Przykład

Dla testu:

```
from private import private

@private('data', 'size')
class Doubler:
    def __init__(self, label, start):
        self.label = label
        self.data = start

    def size(self):
        return len(self.data)

    def double(self):
        for i in range(self.size()):
            self.data[i] = self.data[i]*2

    def display(self):
        print("{} => {}".format(self.label, self.data))

X = Doubler('X to ', [1,2,3])
Y = Doubler('Y to ', [-10, -20, -30])

X.double()
X.display()
Y.label = "mielonka"
Y.double()
Y.display()

try:
    print(X.size())
except TypeError as excp:
    print(excp)

try:
    X.size = 12
except TypeError as excp:
    print(excp)

X.display()
```

Poprawną odpowiedzią jest:

```
X to => [2, 4, 6]
mielonka => [-20, -40, -60]
Pobranie atrybutu prywatnego: size
Modyfikacja atrybutu prywatnego: size
X to => [2, 4, 6]
```