Zadanie F
Proxy

Jasio myli litery. Najczęściej myli małe z dużymi, co bardzo utrudnia mu programowanie w Pythonie. Jasio boryka się z tym problemem już od wielu lat, dzięki czemu nabrał dobrego nawyku – żadna instancja klasy napisanej przez Jasia nie zawiera dwóch atrybutów/metod o nazwach, które byłby takie same, gdyby zaniedbać wielkość liter. Jasio prosi Cię o pomoc - Twoim zadaniem jest napisanie klasy `Proxy`. Konstruktor klasy `Proxy` ma przyjmować jeden argument - obiekt, będący instancją klasy, napisanej przez Jasia. Każda instancja klasy `Proxy` powininna posiadać atrybut `obj`, przechowujący obiekt przekazany w konstruktorze klasy `Proxy` (Jasio nie opisuje precyzyjnie, jak ma się zachywywać odwołanie do atrybutu `obj` klasy `Proxy`; liczy, że domyślisz się, analizując test przykładowy). Każde odczytanie atrybutu/metody A obiektu klasy `Proxy`:

- jeżeli obiekt klasy Jasia związany z danym obiektem klasy `Proxy` zawiera atrybut A o nazwie, która byłaby taka sama jak nazwa atrybutu/metody A, gdyby zaniedbać wielkość liter, powinno zachować się jak odczytanie atrybutu A obiektu klasy Jasia
- zwrócić 42 w przeciwnym wypadku

Każde przypisanie do atrybutu/metody A obiektu klasy `Proxy`:

- jeżeli obiekt klasy Jasia związany z danym obiektem klasy `Proxy` zawiera atrybut A o nazwie, która byłaby taka sama jak nazwa atrybutu/metody A, gdyby zaniedbać wielkość liter, powinno zachować się jak przypisanie do atrybutu A obiektu klasy Jasia
- w przeciwnym przypadku, powinno się zachować jak przypisanie do atrybutu/metody obiektu klasy Jasia, którego/której nazwa powstała przez zamienienie wszystkich liter nazwy atrybutu/metody A na małe. Wyjątek stanowi wspomniany już atrybut `obj`.

W swoim rozwiązaniu nie możesz wykorzystywać funkcji wbudowanej `dir()`

Dostępna pamięć: 32MB

Przykład

Dla danych wejściowych:

```
from proxy import Proxy

class A:
  Test0 = 'test0'
  def __init__(self):
    self.test1 = 'test1'
  def test2(self):
    return 'test2'

class B(A):
  pass

b = B()
p = Proxy(b)
print(p.TeSt0)
print(p.TEST1)
print(p.tESt2())

p.tEsT3 = 'test3'
print(b.test3)
try:
  print(b.TEST3)
except:
  print('b.TEST3 not defined')

p.obj = 'obj'
print(b.obj)
p.Test0 = 'TEST0'
print(p.TEst0)
print(b.Test0)

try:
  print(b.test0)
except:
  print('b.test0 not defined')

print(id(b) == id(p.obj))
print(p.obj.test2())
#taki odczyt nie bedzie testowany:
#print(p.obj)
```

Poprawną odpowiedzią jest:

```
test0
test1
test2
test3
b.TEST3 not defined
obj
TEST0
TEST0
b.test0 not defined
True
test2
```