

## پروکسی

می‌خواهیم کلاسی در زبان پایتون طراحی کنیم که عملکرد یک پروکسی را برای اشیائی که نمایندهٔ آنان است داشته باشد. دقت کنید که این کلاس باید قادر باشد تمامی توابعی را که روی شی اصلی فراخوانی می‌شود، بدون تغییر کارکرد روی آنان اجرا کند. توجه کنید که این شی می‌تواند از هر نوع کلاسی باشد.

### نمونه

برای مثال اگر کد زیر را برای کلاس `Radio` داشته باشیم:

 proxy.py

```
1 class Radio():
2     def __init__(self):
3         self._channel = None
4         self.is_on = False
5         self.volume = 0
6
7     @property
8     def channel(self):
9         return self._channel
10
11    @channel.setter
12    def channel(self, value):
13        self._channel = value
14
15    def power(self):
16        self.is_on = not self.is_on
```

سپس اگر یک پروکسی از شی از کلاس `Radio` بسازیم

 proxy.py

```

1 | radio = Radio()
2 | radio_proxy = Proxy(radio)

```


تمامی توابع بالا باید بر روی radio\_proxy قابل فراخوانی باشد:

 proxy.py

```

1 | radio_proxy.channel = 95
2 | radio_proxy.power()

```

 terminal

```

1 | >>> radio.channel
2 | 95

```

اگر تابع فراخوانی شده، جزو توابع تعریف شده برای شی هدف نباشد، باید استثنایی با پیام No Such Method پرتاب شود.

همچنین کلاس Proxy که باید توابع آن را پیاده سازی کنید، باید قادر باشد اطلاعاتی را نیز ذخیره سازی کند. توابع و ویژگی هایی که این کلاس باید دارا باشد به شرح زیر است.

 proxy.py

```

class Proxy:
    def __init__(self, obj):
        self._obj = obj

    def last_invoked_method(self):
        pass

    def count_of_calls(self, method_name):
        pass

```

```

10
11
12 | def was_called(self, method_name):
    pass

```

#### ▼ تابع last\_invoked\_method

نام آخرین متد معتبر فراخوانی شده بر روی شی هدف را باز می‌گرداند و نوع خروجی آن باید رشته باشد. در صورتی که تا لحظه فراخوانی این تابع، متدی روی شی فراخوانی نشده بود، استثنایی با پیام No Method Is Invoked پرتاب می‌شود.

#### ▼ تابع count\_of\_calls

می‌بایست به ازای نام متد ورودی، تعداد دفعاتی که این متد بر روی شی فراخوانی شده را بازگرداند. اگر متد ورودی معتبر نباشد، مقدار 0 را برمی‌گرداند.

#### ▼ تابع was\_called

به ازای نام متد معتبر ورودی، یک Boolean را به عنوان خروجی برمی‌گرداند که اگر آن متد بر روی شی فراخوانی شده باشد مقدار True و در غیر این صورت مقدار False را برمی‌گرداند. اگر متد ورودی معتبر نباشد مقدار False را برمی‌گرداند.

در نهایت توجه کنید که در صورتی که ویژگی‌ها و متدهایی به شی هدف، پس از ساختن پروکسی اضافه شوند، شی پروکسی باید قادر به مدیریت توابع و ویژگی‌های جدید نیز باشد.

## نکات

- می‌توانید فایل اولیه این سوال را از [این لینک](#) دریافت کنید.
- توابع شما نباید مقداری را چاپ کنند، بلکه باید مقادیر مورد نظر را بازگردانند.

## نحوه ارسال

شما باید کلاس Proxy موجود در فایل proxy.py را تکمیل و سپس این فایل را ارسال کنید.