

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 می‌شوند، بدون تغییر کارکرد روی آن اجرا کند. توجه کنید که این شی می‌تواند از هر نوع کلاسی باشد.

فایل اولیه پروژه را از [اینجا](#) دریافت کنید.

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

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

سپس اگر یک پروکسی از شی از کلاس Radio بسازیم،

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

تمامی توابع بالا باید بر روی radio_proxy قابل فراخوانی باشد:

```
1 >>> radio_proxy.set_channel(95)
2 >>> radio_proxy.power()
```

```

3 | >>> radio.get_channel()
4 | 95

```

اگر تابع فراخوانی‌شده، جزو توابع تعریف‌شده برای شی هدف نباشد، باید استثنایی با پیام No Such Method پرتاب شود.

همچنین کلاس Proxy که باید توابع آن را پیاده‌سازی کنید، باید قادر باشد اطلاعاتی را نیز ذخیره‌سازی کند. توابع و ویژگی‌هایی که این کلاس باید دارا باشد به شرح زیر است:

```

1 | class Proxy:
2 |     def __init__(self, obj):
3 |         self._obj = obj
4 |
5 |     def last_invoked_method(self):
6 |         pass
7 |
8 |     def count_of_calls(self, method_name):
9 |         pass
10 |
11 |     def was_called(self, method_name):
12 |         pass

```

که در آن تابع last_invoked_method نام آخرین متد معتبر فراخوانی شده بر روی شی هدف را بازمی‌گرداند و تایپ 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.