

```
Time to create 1 million NormalClass instances: 1.161 seconds
Time to create 1 million SlotsClass instances: 1.109 seconds
Time to create 1 million WeakrefClass instances: 2.957 seconds
```

Мы можем видеть, что классы с атрибутами slots и normal создаются быстрее, чем класс с атрибутами weakref.

```
Time to read and change attributes of NormalClass instances: 0.175 seconds
Time to read and change attributes of SlotsClass instances: 0.175 seconds
Time to read and change attributes of WeakrefClass instances: 0.179 seconds
```

Мы можем видеть, что классы с normal атрибутами и slots быстрее читают и изменяют атрибуты, чем класс с атрибутами weakref.

Итак, если вас беспокоит производительность, лучше использовать классы со slots или normal атрибутами вместо классов с атрибутами weakref.

```
3000007 function calls in 14.440 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1    1.435    1.435   14.440   14.440 {built-in method builtins.exec}
      1    1.099    1.099   13.005   13.005 <string>:1(<module>)
      1    0.971    0.971   11.905   11.905 my_profile.py:67(test)
      1    0.982    0.982    8.237    8.237 my_profile.py:80(<listcomp>)
1000000    7.255    0.000    7.255    0.000 my_profile.py:22(__init__)
      1    0.618    0.618    2.021    2.021 my_profile.py:74(<listcomp>)
1000000    1.402    0.000    1.402    0.000 my_profile.py:16(__init__)
      1    0.328    0.328    0.676    0.676 my_profile.py:68(<listcomp>)
1000000    0.348    0.000    0.348    0.000 my_profile.py:8(__init__)
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Мы видим, что большая часть времени тратится на __init__ методы классов. Остальное время тратится в основном на list comprehension.