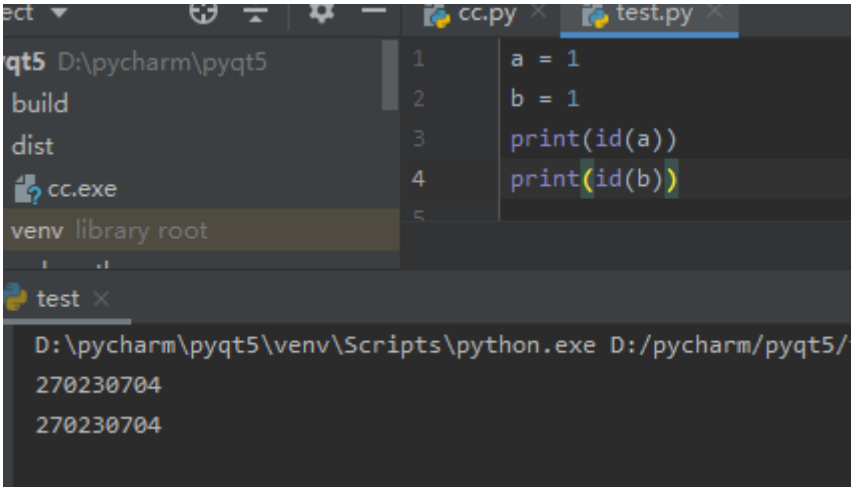
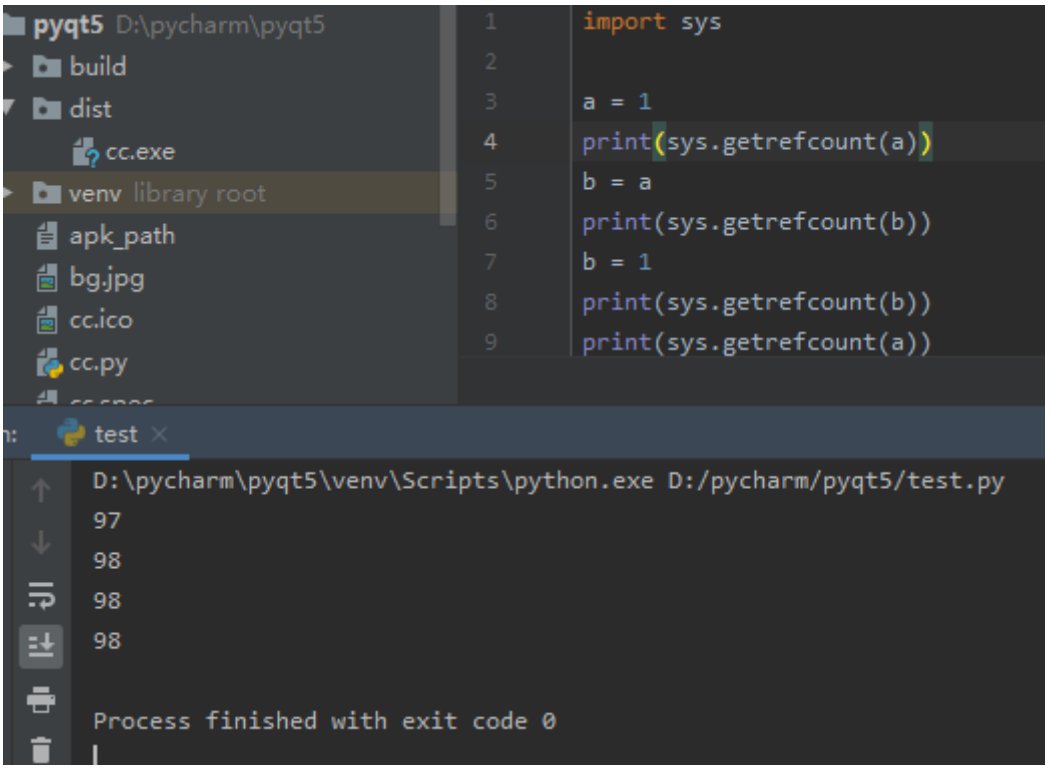
1.python的内存管理



计数机制：

解释：常量1占有固定的内存地址，被不同的对象调用，被调用的个数就是它的计数个数



解释：sys.getrefcount去计算对象的计数个数，不同对象调用时不会增加计数，赋值的对象被调用时会增加计数

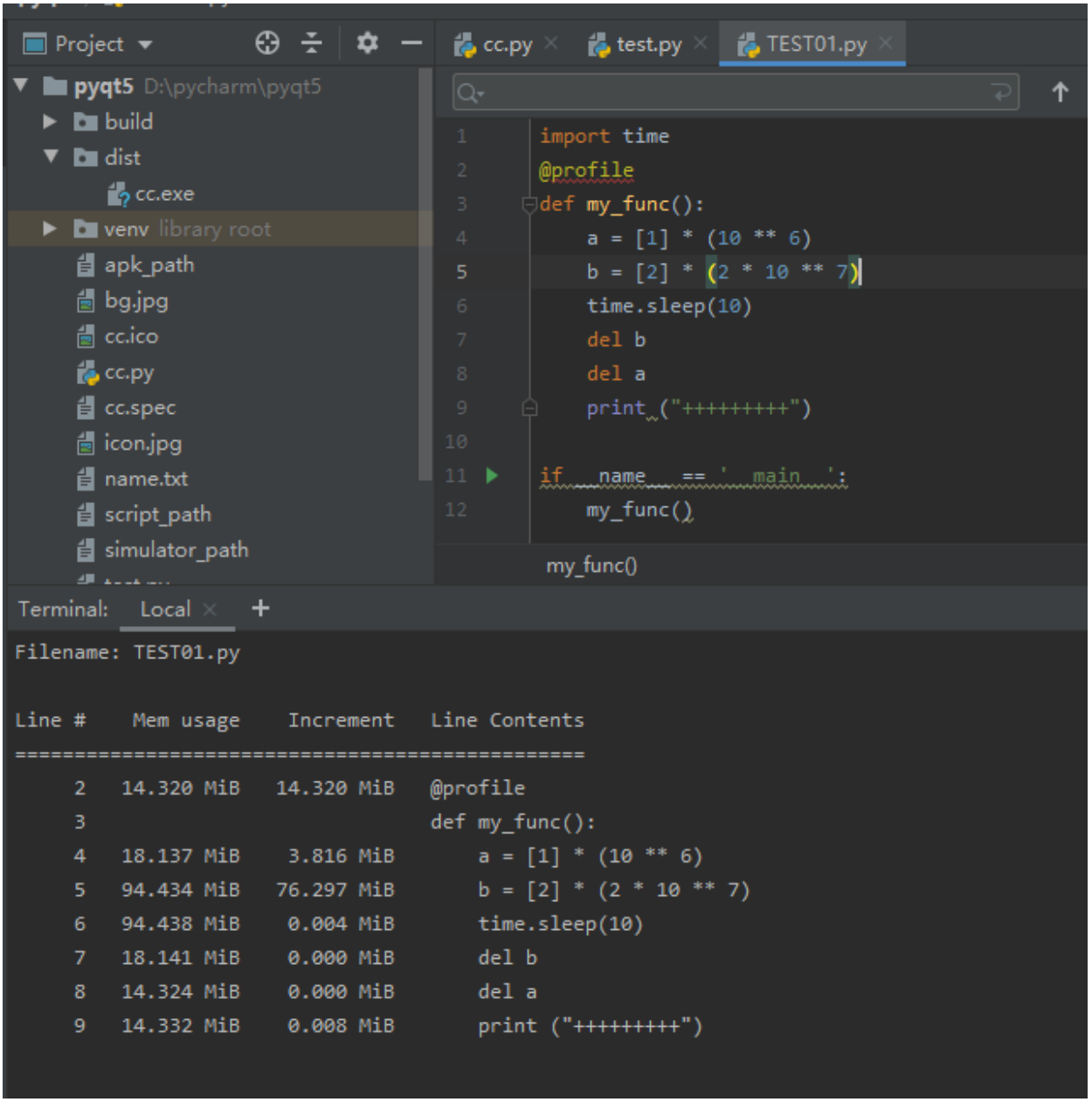
内存泄漏问题：

```python
import sys

list = [1, 2]
a=sys.getsizeof(list)
print(a)
list.append(3)
print(list)
b=sys.getsizeof(list)
print(b)
```

```python
def test():
    list = [1, 2]
    a = sys.getrefcount(list)
    print(a)
    list.append(3)
    b = sys.getrefcount(list)
    print(b)


if __name__ == "__main__":
    test()
```

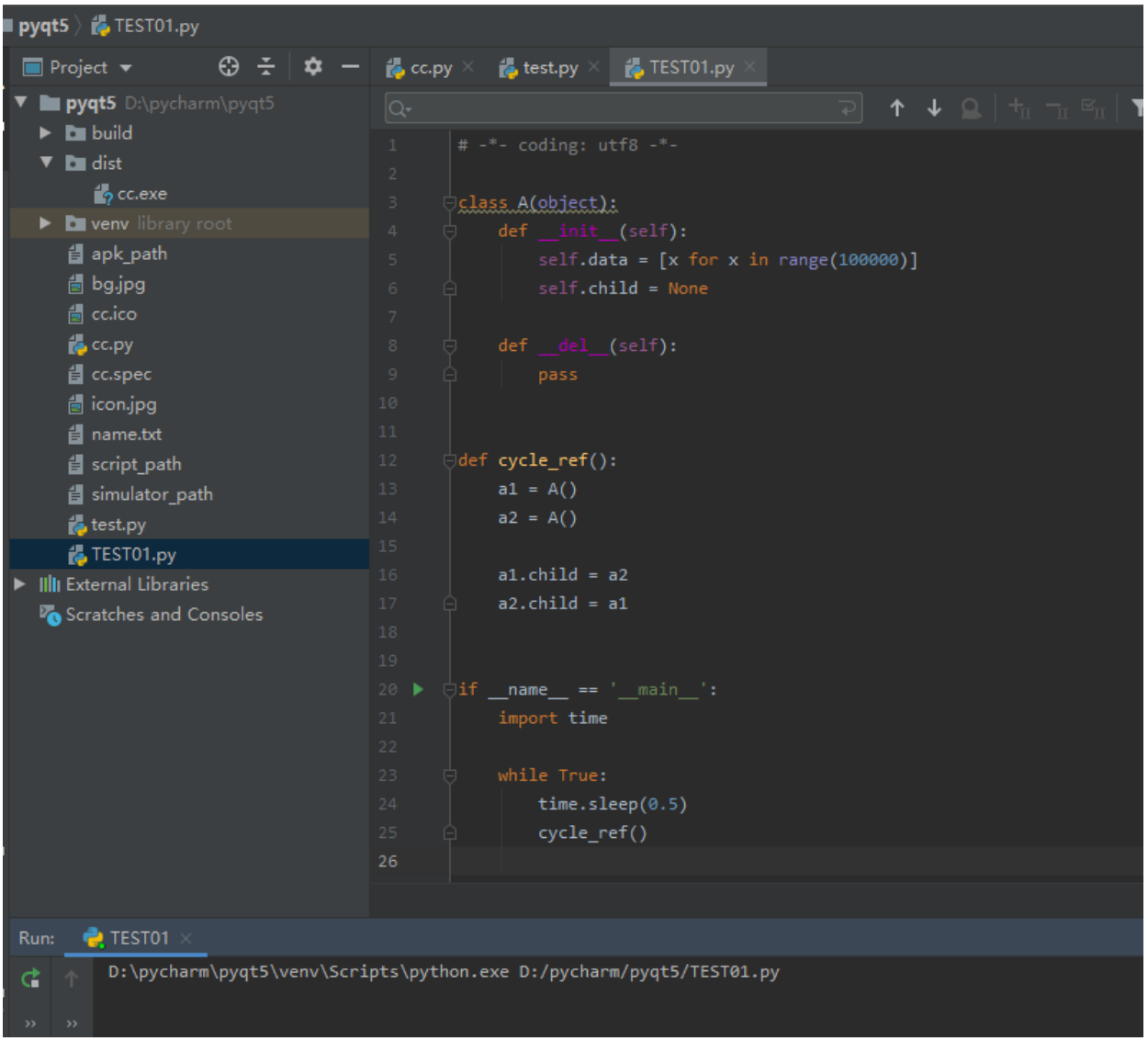
```
Filename: TEST01.py

Line #    Mem usage    Increment   Line Contents
================================================
     3   14.281 MiB   14.281 MiB   @profile
     4                             def test():
     5   14.281 MiB    0.000 MiB       list = [1, 2]
     6   14.281 MiB    0.000 MiB       a = sys.getrefcount(list)
     7   14.289 MiB    0.008 MiB       print(a)
     8   14.289 MiB    0.000 MiB       list.append(3)
     9   14.289 MiB    0.000 MiB       b = sys.getrefcount(list)
    10   14.289 MiB    0.000 MiB       print(b)
```

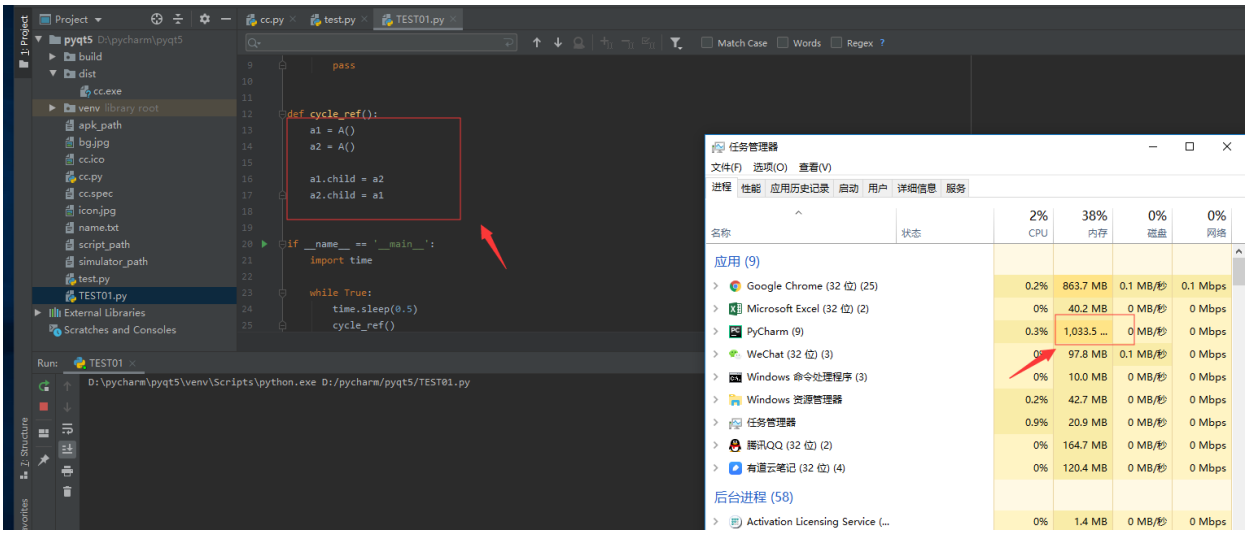
解释：sys.getsizeof检测一般对象的内容占有，list容器修改不会增加内存



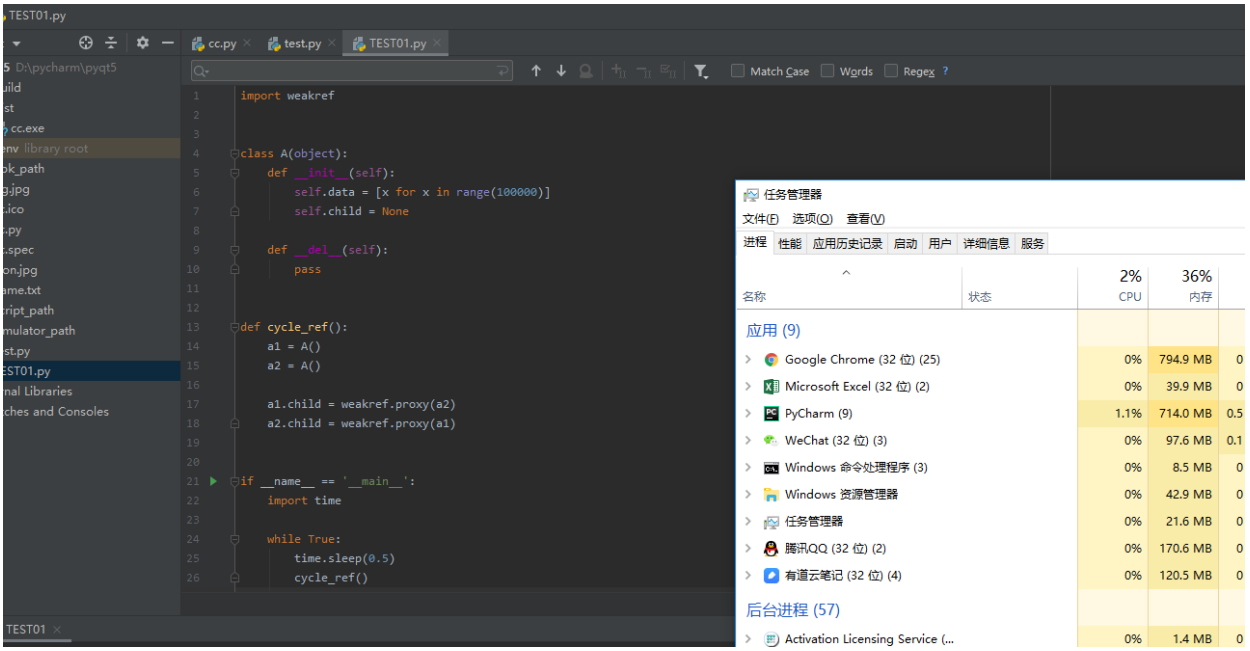
命令:python -m memory_profiler TEST01.py

解释：使用memory_profiler 去查看占用的内存变化

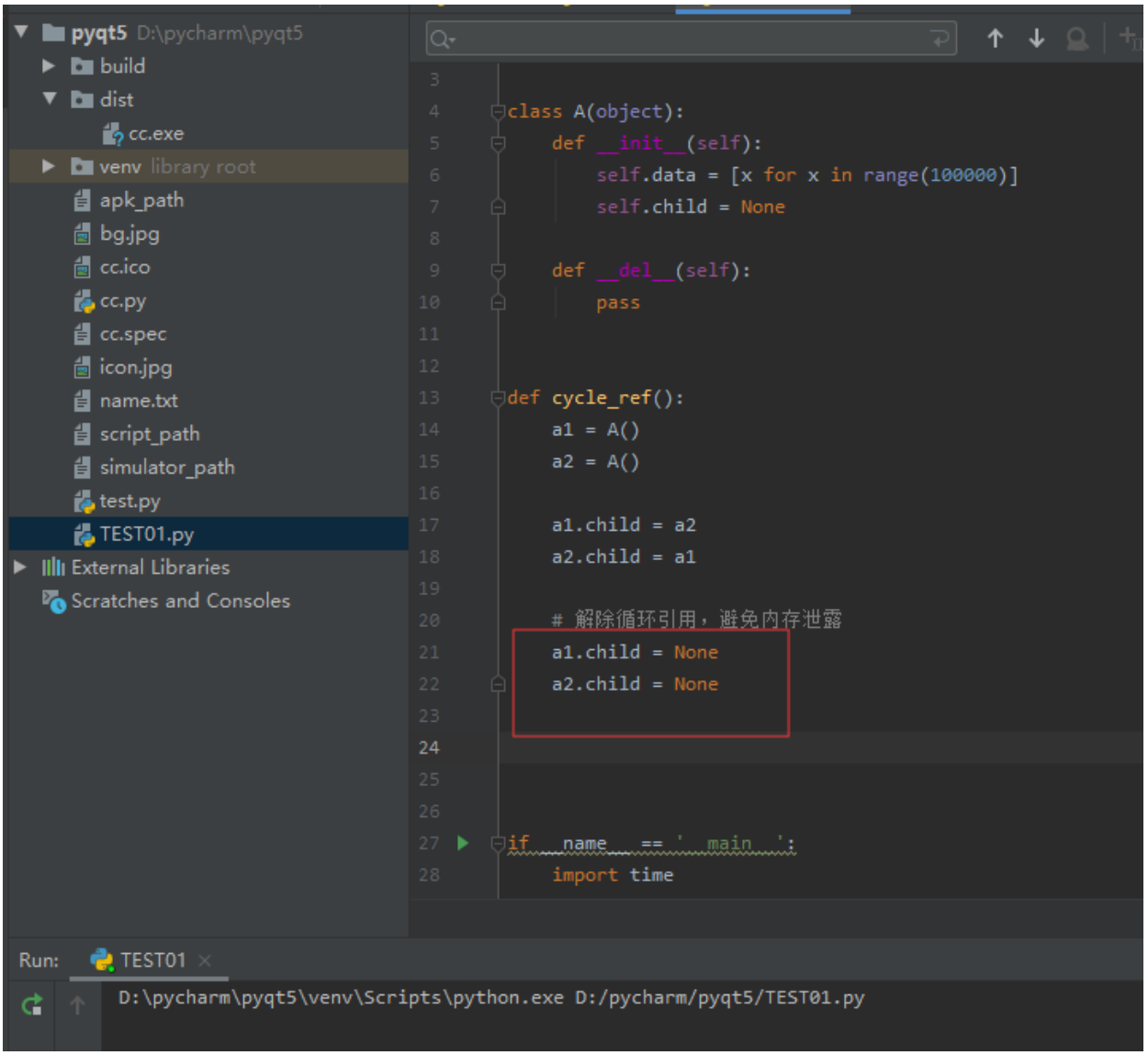




解释：python容器对象的，循环引用，会导致计数器无法垃圾回收，内存持续飙高



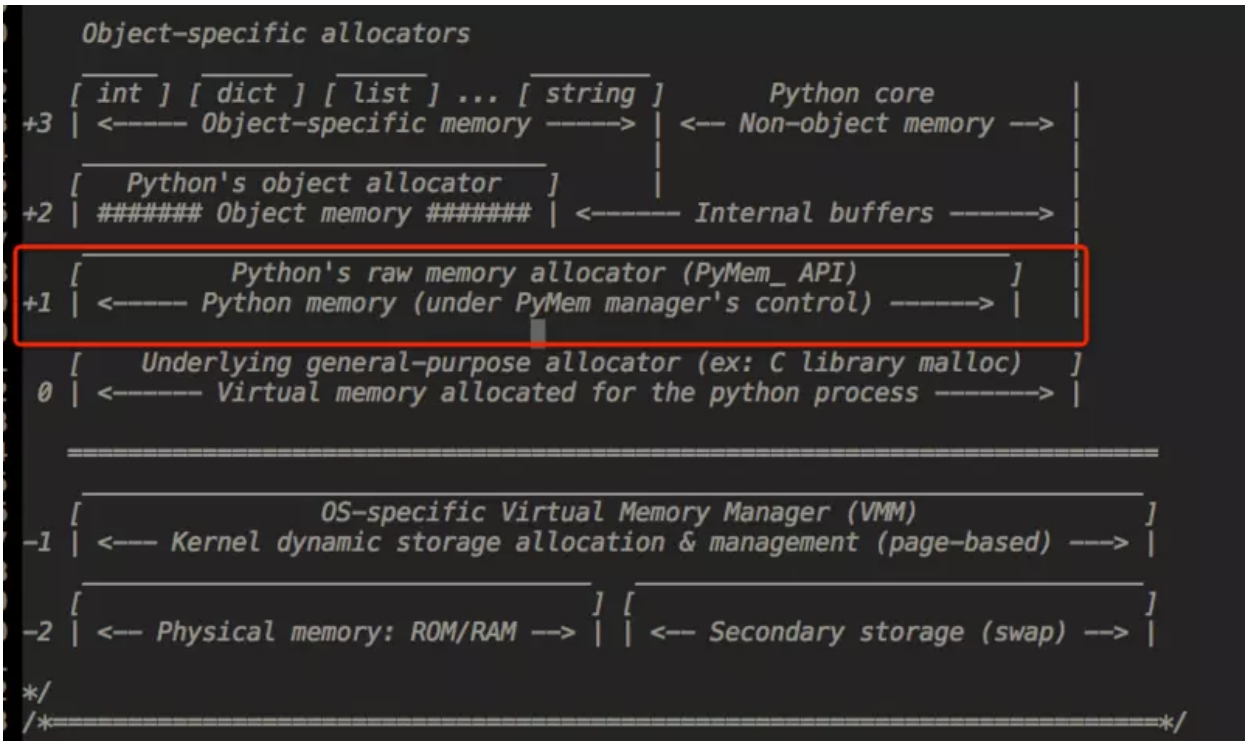
解释：python提供的weakref来处理循环引用的问题，弱引用对象，不计入计数器，但是list、dict不支持



解释：写足够安全的代码，是解决内存泄漏的本质方法

解释：调用gc垃圾回收库，但是会影响性能，降低内存消耗

内存池机制



解释：

对int、dict、list、string对象的直接操作【不分配内存】

最小的int、dict、list、string对象分配内存【缓存对象】pymalloc函数分配<256k

最小的object对象分配内存【缓存对象】pymalloc函数分配<256k

>256kobject对象分配内存【cmalloc函数分配】