In [ ]:

```
%load_ext memory_profiler
```

In [ ]:

```
import gc
from weakref import WeakValueDictionary

from demo import WithoutGc, WithTraverse, WithTraverseAndClear


class Foo:
    def __init__(self, wrapper):
        self.wrapped = wrapper(self)


def run(wrapper) -> int:
    cache = WeakValueDictionary()

    for _ in range(100_000):
        f = Foo(wrapper)
        cache[id(f)] = f
        del f

    gc.collect(0)
    gc.collect(1)
    gc.collect(2)

    return len(cache)
```

In [ ]:

```
print(f"WithoutGc: {run(WithoutGc)}")
print(f"WithTraverse: {run(WithTraverse)}")
print(f"WithTraverseAndClear: {run(WithTraverseAndClear)}")
```

```
WithoutGc: 100000
WithTraverse: 0
WithTraverseAndClear: 0
```

In [ ]:

```
%memit run(WithoutGc)
%memit run(WithTraverse)
%memit run(WithTraverseAndClear)
```

```
peak memory: 134.23 MiB, increment: 26.30 MiB
peak memory: 134.25 MiB, increment: 0.00 MiB
peak memory: 134.25 MiB, increment: 0.00 MiB
```