

Clock.py

```
1 from Counter import Counter
2 class Clock:
3     def __init__(self):
4         self._hour=Counter("Hour")
5         self._min=Counter("Minute")
6         self._sec=Counter("Second")
7
8     def tick(self):
9         if self._sec.ticks<59:
10             self._sec.increment()
11         else:
12             self._sec.reset()
13             if self._min.ticks<59:
14                 self._min.increment()
15             else:
16                 self._min.reset()
17                 if self._hour.ticks<11:
18                     self._hour.increment()
19                 else:
20                     self._hour.reset()
21
22     def reset(self):
23         self._hour.reset()
24         self._min.reset()
25         self._sec.reset()
26
27     @property
28     def clock_time(self):
29         return f"{self._hour.ticks:02}:{self._min.ticks:02}:{self._sec.ticks:02}"
30
```

Counter.py

```
1 class Counter:
2     def __init__(self, name):
3         self._name=name
4         self._count=0
5
6     def increment(self):
7         self._count+=1
8
9     def reset(self):
10        self._count=0
11
12    @property
13    def name(self):
14        return self._name
15
16    @name.setter
17    def name(self, value):
18        self._name=value
19
20    @property
21    def ticks(self):
22        return self._count
23
24    @ticks.setter
25    def ticks(self, value):
26        self._count=value
27
```

Program.py

```
1 import os
2 import tracemalloc
3 from Counter import Counter
4 from Clock import Clock
5
6 def main():
7     tracemalloc.start()
8     myclock = Clock()
9     for i in range(473): # 43200
10         print(myclock.clock_time)
11         myclock.tick()
12
13     current_memory, peak_memory = tracemalloc.get_traced_memory()
14     tracemalloc.stop()
15
16     print(f"Current process:{os.getpid()}")
17     print(f"Current memory usage:{current_memory} bytes")
18     print(f"Peak memory usage:{peak_memory} bytes")
19
20 if __name__ == "__main__":
21     main()
22
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

00:07:44

00:07:45

00:07:46

00:07:47

00:07:48

00:07:49

00:07:50

00:07:51

00:07:52

Current process:17292

Current memory usage:1288 bytes

Peak memory usage:1623 bytes

PS E:\COS20007\week11>

C# memory usage and execution time

```
00:07:52
00:07:53
Current process: System.Diagnostics.Process (Task3_1P)
Physical memory usage: 29278208 bytes
Peak physical memory usage 29278208 bytes

E:\COS20007\week3\Task3_1P\Task3_1P\bin\Debug\net8.0\Task3_1P
.exe (process 32072) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable
Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .|
```

Python memory usage and execution time

```
00:07:44
00:07:45
00:07:46
00:07:47
00:07:48
00:07:49
00:07:50
00:07:51
00:07:52
Current process:17292
Current memory usage:1288 bytes
Peak memory usage:1623 bytes
PS E:\COS20007\week11> █
```