Clock.py

```
from Counter import Counter

class Clock:

    def __init__(self):
        self.second = Counter(name="second")
        self.minute = Counter(name="minute")
        self.hour = Counter(name="hour")

    def Tick(self):
        self.second.Increment()
        if self.second.Ticks > 59:
            self.second.Reset()
            self.minute.Increment()

            if self.minute.Ticks > 59:
                self.minute.Reset()
                self.hour.Increment()

                if self.hour.Ticks > 23:
                    self.Reset()

    def Reset(self):
        self.second.Reset()
        self.minute.Reset()
        self.hour.Reset()

    @property
    def Time(self):
        return
f"{self.hour.Ticks:02}:{self.minute.Ticks:02}:{self.second.Ticks:02}"
```

Counter.py

```
class Counter:
    def __init__(self, name):
        self.name = name
        self.count = 0

    def Increment(self):
        self.count += 1

    def Reset(self):
        self.count = 0

    @property
    def Name(self):
```

```
        return self.name

    @Name.setter
    def Name(self, value):
        self.name = value

    @property
    def Ticks(self):
        return self.count
```

Main.py

```
from Clock import Clock

def main():

    clock = Clock()
    i = 0

    for i in range(0, 86400):
        clock.Tick()
        print(clock.Time)
        i += 1

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