

《Python 程序设计高阶》

(2023-2024 学年第 1 学期)

作业 4

学号:2021329600006 姓名: 陈昊天 班级: 计算机科学与技术 21(4) 班
学号:2021329621213 姓名: 陈佳伟 班级: 计算机科学与技术 21(3) 班
学号:2021329621257 姓名: 冯佳钧 班级: 计算机科学与技术 21(4) 班

§1 题目 1

请写一个函数，输入一个整数 n ，然后控制计算机在 n 秒内关机

§1.1 解答

```
1 import subprocess
2 import sys
3 import time
4
5 def schedule_shutdown_precise(seconds):
6     try:
7         seconds = int(seconds)
8         if seconds <= 0:
9             raise ValueError("请输入一个正整数。")
10
11         print(f"计算机将在{seconds}秒后关机。")
12         time.sleep(seconds)
13
14         subprocess.run(["sudo", "shutdown", "-h", "now"], check=True)
15     except ValueError as e:
16         print("错误:", e)
17     except subprocess.CalledProcessError as e:
18         print("关机命令执行失败:", e)
19     except KeyboardInterrupt:
20         sys.exit("用户中断了操作。")
21
22 if __name__ == "__main__":
23     try:
24         n = input("请输入计划关机的秒数: ")
25         schedule_shutdown_precise(n)
26     except KeyboardInterrupt:
27         sys.exit("用户中断了操作。")
```

```
root@ex8:~# vim shutdown.py
root@ex8:~# python3 shutdown.py
请输入计划关机的秒数：5
计算机将在5秒后关机。
```

§2 题目 2

用上课的课件中带参数的装饰器的例子，来解决一个实际的问题，说明在这个例子中，装饰器的作用。

§2.1 解答

定义一个装饰器工厂函数 [1]，接收一个参数来决定是否记录被装饰函数的执行时间。如果启用记录，装饰器会在函数执行前后分别记录时间，并计算出函数执行所需的总时间。

```
1 import time
2
3
4 # 装饰器工厂函数，决定是否记录函数执行时间
5 def record_time(enabled=True):
6     def decorator(func):
7         def wrapper(*args, **kwargs):
8             if enabled:
9                 start_time = time.time()
10                result = func(*args, **kwargs)
11                end_time = time.time()
12                print(
13                    f"Function {func.__name__} took {end_time - start_time}
14                      seconds to run."
15                )
16            else:
17                result = func(*args, **kwargs)
18            return result
19        return wrapper
20
21    return decorator
22
23
24 # 使用装饰器，启用时间记录
25 @record_time(enabled=True)
26 def do_some_calculation(n):
27     sum = 0
28     for i in range(n):
29         sum += i
30     return sum
31
```

```
32 |
33 | # 调用函数
34 | result = do_some_calculation(1000000)
```

```
root@ex8:~# python3 test.py
Function do_some_calculation took 0.1831073760986328 seconds to run.
root@ex8:~# █
```

§3 总结

§3.1 分工情况

以下说明小组的分工情况，每个人完成了什么功能，完成的情况如何

陈昊天

负责报告的整合撰写；

负责题目 1 的研究和解答；

陈佳伟

负责题目 2 装饰器工厂函数的编写；

冯佳钧

负责题目 2 装饰器的使用和功能测试；

§3.2 解决方案评价

以下说明小组对解决方案的评价，是否完成了题目的要求，怎么证明完成了题目的要求，好的地方在哪里，哪里有不足。

3.2.1 题目 1

完成了题目要求，控制计算机在 n 秒内关机；

优点：能够处理非法输入和运行时错误；

缺点：在非 Linux 系统上需要修改关机命令；

3.2.2 题目 2

完成了题目要求，用带参数的装饰器来解决记录函数的执行时间的问题；

优点：装饰器使得记录执行时间的功能可以应用于任何函数，增加了代码的复用性；

缺点：装饰器引入了额外的函数调用开销，对于执行时间非常短的函数，可能会影响性能测量的准确性；

参考文献

- [1] Wikipedia contributors. Decorator pattern — Wikipedia, the free encyclopedia, 2023. [Online; accessed 1-December-2023].