Module 4

1. Написать декоратор, который откладывает исполнение функции на 3 секунды

Пример:

@delay

def print\_string(output\_str):

print(output\_str)

print\_string() # исполнится через 3 сек

1. Написать программу, которая выводит переданное текстовое сообщение. Написать декораторы, которые могут изменить эту надпись сделав ее жирной, наклонной, или подчеркнутой добавив соответствующие HTML теги.

Пример:

@underline

def print\_styled(styling\_string):

return str(styling\_string)

print\_styled("test") # <u>test</u>

@bold

@italic

@underline

def print\_styled(styling\_string):

return str(styling\_string)

print\_styled("Hello world") # <i><b><u>Hello world</u></b></i>'

1. Написать декоратор который удостовериться что только определенный тип ошибки будет выводиться в декорируемой функции.

Пример:

@raises(FileNotFoundError)

def return\_str():

return "test"+ 7

return\_str("Hello world") # Ошибка FileNotFoundError

1. Сохранить все декораторы в отдельных файлах. Подключить в качестве пакета к файлу с основной функцией.

python/module4

├── main.py  
 └── utils  
 ├── \_\_init\_\_.py  
 ├── delay.py  
 ├── html\_dec.py  
 ├── raises.py