

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЁТ**  
**по лабораторной работе №6**  
Дисциплина: «Языки программирования»

Выполнил: студент 2 курса  
группы ИТС-б-о-20-1

Попов Данила Владимирович

Проверил:

к.ф.-м.н., доцент

кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: \_\_\_\_\_

Ставрополь, 2022

Цель работы: приобретение навыков по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.x

<https://github.com/12W300/Six.git>

Ход работы:

Пример.

```
def benchmark(func):
    import time

    def wrapper(*args, **kwargs):
        start = time.time()
        return_value = func(*args, **kwargs)
        end = time.time()
        print('[*] Время выполнения: {} секунд.'.format(end-start))
        return return_value
    return wrapper

@benchmark
def fetch_webpage(url):
    import requests
    webpage = requests.get(url)
    return webpage.text

webpage = fetch_webpage('https://google.com')
print(webpage)
```

Рисунок 1 – код примера

```
[*] Время выполнения: 1.4475083351135254 секунд.
<!doctype html><html itemscope="" itemtype="http://schema.org/webPage">
```

Рисунок 2 – результат работы примера

Задание.

3. Вводятся два списка (каждый с новой строки) из слов, записанных через пробел. Имеется функция, которая преобразовывает эти две строки в два списка слов и возвращает эти списки. Определите декоратор для этой функции, который из этих двух списков формирует словарь, в котором ключами являются слова из первого списка, а значениями - соответствующие элементы из второго списка. Полученный словарь должен возвращаться при вызове декоратора. Примените декоратор к первой функции и вызовите ее. Результат (словарь) отобразите на экране.

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == "__main__":
5      def decorator(fn):
6          def wrapper(st1, st2):
7              print(st1)
8              print(st2)
9              arrkey, arrval = fn(st1, st2)
10             return dict(zip(arrkey, arrval))
11         return wrapper
12
13
14     @decorator
15     def str_to_list(s1, s2):
16         lst1 = list(s1.split())
17         lst2 = list(s2.split())
18         return lst1, lst2
19
20
21     str1 = 'key1 key2 key3 key4 key5'
22     str2 = 'val1 val2 val3 val4 val5'
23
24     print(str_to_list(str1, str2))
25

```

Рисунок 2 – код задания

```

key1 key2 key3 key4 key5
val1 val2 val3 val4 val5
{'key1': 'val1', 'key2': 'val2', 'key3': 'val3', 'key4': 'val4', 'key5': 'val5'}

```

Рисунок 3 – результат работы кода

### Контрольные вопросы:

#### 1. Что такое декоратор?

Декоратор — это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.

#### 2. Почему функции являются объектами первого класса?

Объектами первого класса в контексте конкретного языка программирования называются элементы, с которыми можно делать всё то же, что и с любым другим объектом: передавать как параметр, возвращать из функции и присваивать переменной.

#### 3. Каково назначение функций высших порядков?

Функции высших порядков — это такие функции, которые могут принимать в качестве аргументов и возвращать другие функции.

#### 4. Как работают декораторы?

Декоратор — это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.

#### 5. Какова структура декоратора функций?

Функция `decorator` — это, как можно понять по названию, декоратор. Она принимает в качестве параметра функцию. Внутри функции объявляется другая под названием `wrapper`, которая что-то выполняет. В конце возвращается функция `wrapper`.

Вывод по проделанной работе: приобретены навыки по работе с декораторами при написании программ с помощью языка программирования Python версии 3.