

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.12
дисциплины «Программирование на Python»
Вариант №2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Декораторы функций в языке Python

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

Ход работы

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def benchmark(func):
    import time

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

    return wrapper

@benchmark
def fetch_webpage():
    import requests
    webpage = requests.get('https://google.com')

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

```
[*] Время выполнения: 5.485761880874634 секунд.
```

Рисунок 1. Результат выполнения программы

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

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

if __name__ == "__main__":
    webpage = fetch_webpage('https://google.com')
    print(webpage)

```

```

[*] Время выполнения: 5.197993278503418 секунд.
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage"

```

Рисунок 2. Результат выполнения программы

Индивидуальное задание:

2. На вход программы поступает строка из целых чисел, записанных через пробел. Напишите функцию `get_list`, которая преобразовывает эту строку в список из целых чисел и возвращает его. Определите декоратор для этой функции, который сортирует список чисел, полученный из вызываемой в нем функции. Результат сортировки должен возвращаться при вызове декоратора. Вызовите декорированную функцию `get_list` и отобразите полученный отсортированный список на экране.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def sort_func(func):
    def wrapper():
        lst = sorted(func())
        print(f"Отсортированный список: {lst}")

    return wrapper()

@sort_func
def get_list():
    s = input("Строка: ")
    lst = [int(n) for n in s.split(" ")]
    return lst

```

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

```
Строка: 3 -2 7 -5 6 9 8 -1 1  
Отсортированный список: [-5, -2, -1, 1, 3, 6, 7, 8, 9]
```

Рисунок 3. Результат выполнения программы

Контрольные вопросы:

1. Декоратор - это функция, которая позволяет обернуть другую функцию для расширения её функциональности без непосредственного изменения её кода.
2. Функции — это объекты первого класса, так как мы можем сохранять функции в переменные, передавать их в качестве аргументов и возвращать из других функций. Можно даже определить одну функцию внутри другой.
3. Функции высших порядков — это такие функции, которые могут принимать в качестве аргументов и возвращать другие функции.
4. Функция-декоратор является функцией высшего порядка, так как принимает функцию в качестве аргумента, а также возвращает функцию. Внутри функции-декоратора мы определяем другую функцию, обертку, так сказать, которая обертывает функцию-аргумент и затем изменяет её поведение. Декоратор возвращает эту обертку.
5. Чтобы получить декоратор, в который можно передать аргументы, нужно из функции с параметрами вернуть функциональный объект, который может действовать как декоратор.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.x.

