

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.12

Декораторы функций в языке Python

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Декораторы функций в языке Python.

Цель работы: приобретение навыков по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.10.

Порядок выполнения работы:

1. Проработка примеров:

Код программы:

```
#!/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))

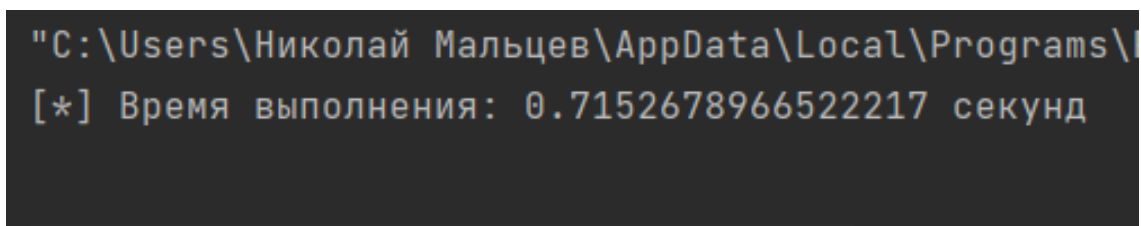
    return wrapper

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

fetch_webpage()

if __name__ == '__main__':
    print("")
```

Результат работы программы:



```
"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\python.exe"
[*] Время выполнения: 0.7152678966522217 секунд
```

Рисунок 1. Результат работы программы из примера 1

Код 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
    wrapper = requests.get(url)
    return wrapper.text

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

Результат работы программы:

```
"C:\Users\Николай Мальцев\AppData\Local\Programs\Python\Python310\python.exe"
[*] Время выполнения: 0.596172571182251 секунд
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="ru">
<head><meta charset="utf-8"><title>Google</title><script>
var f=this||self;var h,k=[];function l(a){for(var b;a&&(!a.getAttribute||!(b=
function n(a,b,c,d,g){var e="";c||-1!==b.search("&ei=")||e="&ei="+l(d),-1===
google.y={};google.sy=[];google.k=function(a,b){if(a)var c=a.id;else{do c=Mat
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.t
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;over
var h=this||self;var k,l=null!=(k=h.mei)?k:1,n,p=null!=(n=h.sdo)?n:!0,q=0,r,t
a.fileName;g&&(0<g.index0f("-extension:/")&&(e=3),c+="&script="+b(g),f&&g===v
if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.fc
})
})());</script><div id="mngb"><div id=gbar><noabr><b class=gb1>&#1055;&#1086;&#
else top.location='/doodles/';}})();</script><input value="AJiK0e8AAAAAY0vgf
var a,b="1";if(document&&document.getElementById)if("undefined"!==typeof XMLHt
```

Рисунок 2. Результат работы программы из примера 2

2. Выполнение индивидуального задания (вариант 1):

Листинг программы:

```
#!/usr/bin/env python3
# *_ coding: utf-8 *_
"""
Объявите функцию с именем get_sq , которая вычисляет площадь прямоугольника
по двум
параметрам: width и height – ширина и высота прямоугольника и возвращает
результат.
Определите декоратор для этой функции с именем (внешней функции) func_show ,
который
```

```

отображает результат на экране в виде строки (без кавычек): "Площадь
прямоугольника:
<значение>". Вызовите декорированную функцию get_sq .

"""

def func_show(func):
    def wrapper(width, height):
        res = func(width, height)
        print(f"Площадь прямоугольника: {res}")
        return res

    return wrapper

@func_show
def get_sq(width, height):
    return width * height

if __name__ == '__main__':
    w = int(input("Введите ширину: "))
    h = int(input("Введите высоту: "))
    get_sq(w, h)

```

Результат выполнения программы:

```

"C:\Users\Николай Мальцев\AppData\Local\
Введите ширину: 5
Введите высоту: 6
Площадь прямоугольника: 30

```

Рисунок 4. Результат выполнения программы к индивидуальному заданию

Ответы на вопросы:

1. Что такое декоратор?

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

2. Почему функции являются объектами первого класса?

С функциями можно производить те же действия, что и с объектами: передавать их в качестве параметров, присваивать переменным, возвращать. Функции являются объектами.

3. Каково назначение функций высших порядков?

Функции высших порядков могут принимать в качестве аргументов и возвращать другие функции.

4. Как работают декораторы?

Декораторы позволяют расширить возможности функции без непосредственного изменения в коде.

5. Какова структура декоратора функций?

Декоратор представляет из себя функцию, в которую передаётся декорируемая функция в качестве аргумента. Внутри декоратора находится функция-обёртка, которая и декорирует функцию.

6. Самостоятельно изучите как можно передать параметры декоратору, а не декорируемой функции?

Декораторы – это функции, которые возвращают функции. Для передачи аргумента декоратору необходимо вызвать функцию, которая возвращает декоратор, который возвращает функцию. В итоге мы получаем функцию с двумя вложенными в неё функциями.

Вывод: в ходе работы были приобретены навыки по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.10.