

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 14 (2.12)
Декораторы функций в языке Python

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Злыгостев И.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

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

Ход работы

1. После изучения теоретической части, приступил к разбору примера.

```
def decorator_function(func):  
    def wrapper():  
        print('Функция-обёртка!')  
        print('Оборачиваемая функция: {}'.format(func))  
        print('Выполняем обёрнутую функцию...')  
        func()  
        print('Выходим из обёртки')  
    return wrapper  
  
@decorator_function  
def hello_world():  
    print("Hello, world!")  
  
hello_world()
```

Рисунок 14.1 – Разбор примера

2. Затем приступил к выполнению индивидуального задания

Индивидуальное задание

Вариант 5

Вводится строка целых чисел через пробел. Напишите функцию, которая преобразовывает эту строку в список чисел и возвращает их сумму. Определите декоратор для этой функции, который имеет один параметр start – начальное значение суммы. Примените декоратор со значением start=5 к

функции и вызовите декорированную функцию. Результат отобразите на экране.

```
def decorate_function(func):
    start = 5

    def arg(message):
        summ = 0
        lst = message.split(' ')
        for i in lst:
            i = int(i)
            summ += i
        summ += start
        func(summ)
    return arg

@decorate_function
def main(message):
    print("Сумма чисел со start=5: ", message)

if __name__ == '__main__':
    main(input('Введите целые числа через пробел: '))
```

Рисунок 14.2 – Код индивидуального задания

```
Введите целые числа через пробел: 1 2 3
Сумма чисел со start=5: 11
```

Рисунок 14.3 – Результат выполнения 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?

Потому что с ними можно работать как с переменными, могут быть переданы как аргумент процедуры, могут быть возвращены как результат выполнения процедуры, могут быть включены в другие структуры данных.

3. Каково назначение функций высших порядков?

Основной задачей функций высших порядков является возможность принимать в качестве аргументов и возвращать другие функции.

4. Как работают декораторы?

Они берут декорируемую функцию в качестве аргумента и позволяют совершать с ней какие-либо действия до и после того, что сделает эта функция, не изменяя её.

5. Какова структура декоратора функций?

Функция `decorator` принимает в качестве аргумента функцию `func`, внутри функции `decorator` другая функция `wrapper`. В конце декоратора происходит возвращение функции `wrapper`.

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

Достаточно обернуть функцию декоратор в другую функцию, которая будет принимать аргументы. И сделать вывод функций `wrapper` и `decorator`.

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.