

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2.12**

**Декораторы функций в языке Python**

**по дисциплине «Технологии программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

### Ход работы: Примеры

<https://github.com/Mirror-Shard/L2.12>

1. Проработал примеры из учебника, внешний вид декоратора:

```
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
```

Рисунок 1 – Декоратор

2. Результат работы примера:

```
Функция-обёртка!  
Оборачиваемая функция: <function hello_world at 0x000002500E04BCA0>  
Выполняем обёрнутую функцию...  
Hello, world!  
Выходим из обёртки
```

Рисунок 2 – Результат

### Индивидуальное задание 1

Объявите функцию, которая принимает строку на кириллице и преобразовывает ее в латиницу, используя следующий словарь для замены русских букв на соответствующее латинское написание. Функция должна возвращать преобразованную строку. Замены делать без учета регистра (исходную строку перевести в нижний регистр – малые буквы).

1. Создал функцию, которая делает транслит переданной в неё строки:

```
Введите строку русскими буквами: антошка
Транслитерация строки: antoshka
```

Рисунок 3 – Транслит

2. Создал добавил декоратор, которые меняет все ! и ? на - :

```
Введите строку русскими буквами: антонио!?
Транслитерация строки: antonio-
```

Рисунок 4 – Замена ! и ? на -

3. Декоратор также удаляет подряд идущие '-' и оставляет только 1:

```
Введите строку русскими буквами: Святослав!!!!
Транслитерация строки: svyatoslav-
```

Рисунок 5 – Удаление лишних '-'

### 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 позволяет

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 работы приобрёл навыки по работе с функциями с замыканиями при написании программ с помощью языка программирования Python версии 3.