

**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 Python3.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Малышев А.Ю. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверила Воронкин Р.А. _____

(подпись)

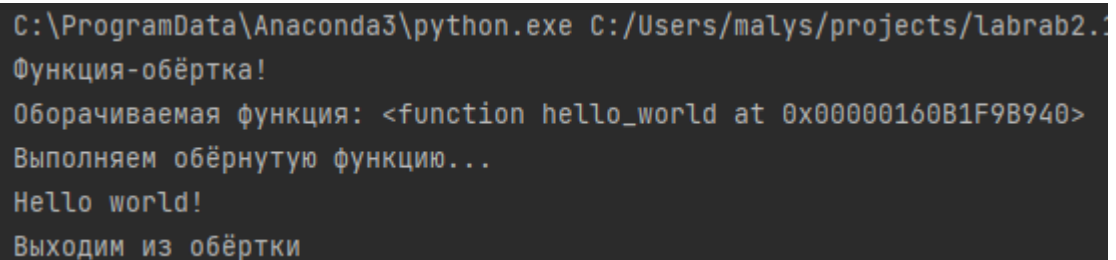
Ставрополь 2021

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

Ссылка на репозиторий: <https://github.com/AlexandrM333/labrab2.12.git>

Ход работы:

1. Проработал пример из теоретической части:



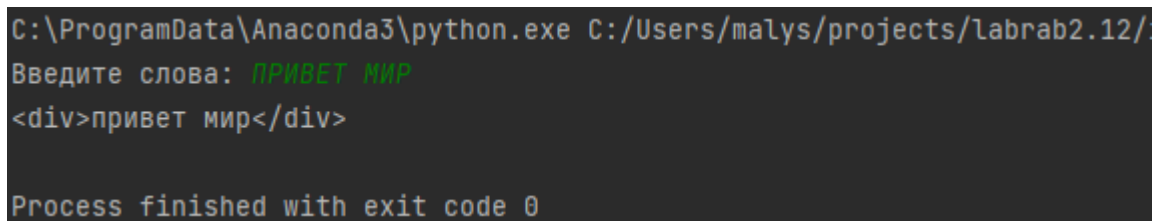
```
C:\ProgramData\Anaconda3\python.exe C:/Users/malys/projects/labrab2.12/
Функция-обёртка!
Оборачиваемая функция: <function hello_world at 0x00000160B1F9B940>
Выполняем обёрнутую функцию...
Hello world!
Выходим из обёртки
```

Рисунок 1. Пример №1.

2. Выполнил индивидуальное задание:

(Вариант – 6)

Объявите функцию, которая возвращает переданную ей строку в нижнем регистре (с малыми буквами). Определите декоратор для этой функции, который имеет один параметр tag, определяющий строку с названием тега (начальное значение параметра tag равно h1). Этот декоратор должен заключать возвращенную функцией строку в тег tag и возвращать результат. Пример заключения строки "python" в тег h1: <h1>python</h1>. Примените декоратор со значением tag="div" к функции и вызовите декорированную функцию. Результат отобразите на экране.



```
C:\ProgramData\Anaconda3\python.exe C:/Users/malys/projects/labrab2.12/
Введите слова: ПРИВЕТ МИР
<div>привет мир</div>

Process finished with exit code 0
```

Рисунок 2. Индивидуальное задание.

Контрольные вопросы:

1. Что такое декоратор? Декораторы – один из самых полезных инструментов в Python, – это функция, которая позволяет обернуть другую

функцию для расширения её функциональности без непосредственного изменения её кода.

2. Почему функции являются объектами первого класса? В Python всё является объектом, этот факт открывает перед нами множество возможностей. Мы можем сохранять функции в переменные, передавать их в качестве аргументов и возвращать из других функций. Можно даже определить одну функцию внутри другой. Иными словами, функции – это объекты 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. Функции высших порядков, вроде дифференциального оператора, принимают на входе функцию и возвращают другую функцию, производную от исходной. Функции высших порядков в программировании работают точно так же – они либо принимают функцию(и) на входе и/или возвращают функцию(и).

4. Как работают декораторы? Внутри функции-декоратора определяется другая функция, обёртка, так сказать, которая обёртывает функцию-аргумент и затем изменяет её поведение. Декоратор возвращает эту обёртку. Просто добавив `@decorator` перед определением функции, её поведение меняется. Однако выражение с `@` является всего лишь синтаксическим сахаром. Иными словами, выражение `@decorator` вызывает функцию-декоратор с функцией в качестве аргумента.

5. Какова структура декоратора функций? В начале объявляется функция в которой будет использоваться декорируемая функция. Далее объявляется декорируемая функция перед которой используется конструкция `@decorator`, где “decorator” имя функции, использующей декорируемую функцию.

6. Самостоятельно изучить как можно передать параметры декоратору, а не декорируемой функции? Для этого после объявления декоратора в скобках нужно указать необходимые параметры: `@decorator(p1)`

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