12Министерство 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 №15**

**дисциплины «****Программирование на python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Кожуховский Виктор Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем  », очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Декораторы функций в языке Python

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

# Методика и порядок выполнения работы

1. Изучил теоретический материал работы.

2.Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

3. Выполнил клонирование созданного репозитория.

4. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

5. Организовал свой репозиторий в соответствие с моделью ветвления git-flow.

6. Создал проект в папке репозитория.

7. Проработал примеры лабораторной работы.

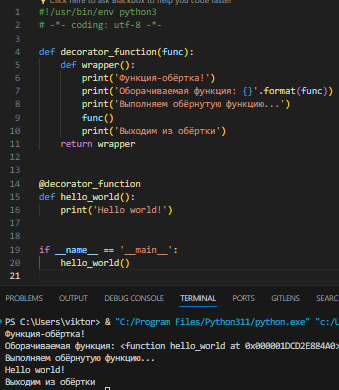


Рисунок 1. Код примера 1 и его выполнение

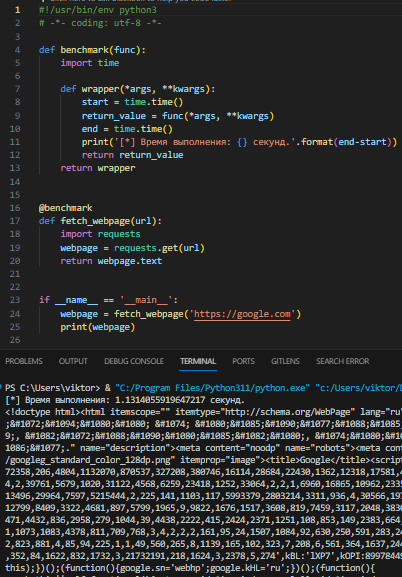


Рисунок 2. Код примера 2 и его выполнение

8. Выполнил индивидуальное задание.

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.

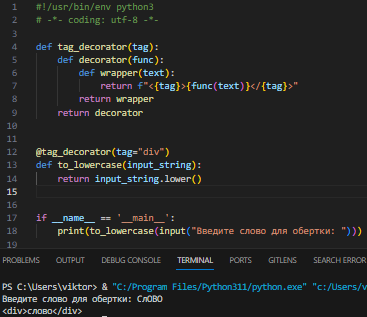


Рисунок 3. Код индивидуального задания и его выполнение

9. Зафиксировал сделанные изменения в репозитории.

10. Добавил отчет по лабораторной работе в формате PDF в папку doc репозитория. Зафиксировал изменения.

11. Выполнил слияние ветки для разработки с веткой master/main.

12. Отправил сделанные изменения на сервер GitHub.

Вопросы для защиты работы

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?

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.

4. Как работают декораторы?

Декораторы позволяют изменять или расширять поведение функций или классов без изменения их кода. Они "оборачивают" исходную функцию, предоставляя дополнительную функциональность до и/или после основной функции, не изменяя её поведение напрямую.

5. Какова структура декоратора функций?

Структура декоратора функций в Python обычно состоит из трех уровней вложенности:

Внешняя функция, которая является самим декоратором. Она принимает функцию в качестве аргумента и возвращает обернутую функцию (внутреннюю функцию).

Внутренняя функция (обернутая функция), которая выполняет действия перед и/или после вызова декорируемой функции. Она принимает аргументы декорируемой функции \*args и \*\*kwargs, вызывает декорируемую функцию и возвращает ее результат.

Сама декорируемая функция, которая выполняет основную работу.

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

Чтобы передать параметры декоратору, нам нужно использовать двусмысленность — первый вызов декоратора должен принимать аргументы, и внутри него должна быть определена и возвращена внутренняя функция, которая является самим декоратором.