МИНИCTEPCTBO 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 алгоритмизация»**

Выполнил студент группы ИВТ-б-о-20-1

Погорелов Д.Н. « »\_\_\_\_\_\_\_\_\_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. Создал общедоступный репозиторий, клонировал его локальный сервер.
2. Изучил теоретический материал и проработал примеры.

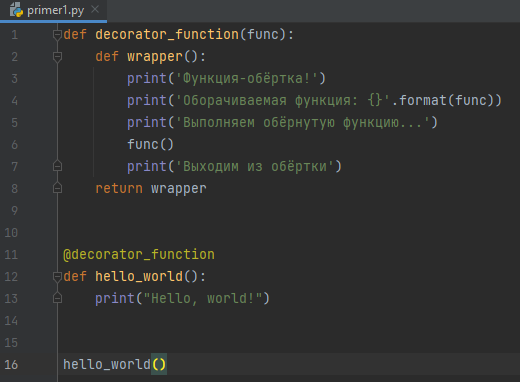


Рисунок 1 – Пример кода с декораторами

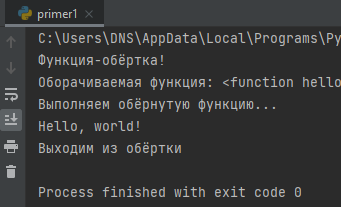


Рисунок 2 – Результат работы кода

1. Проработал второй пример с использованием декоратора.

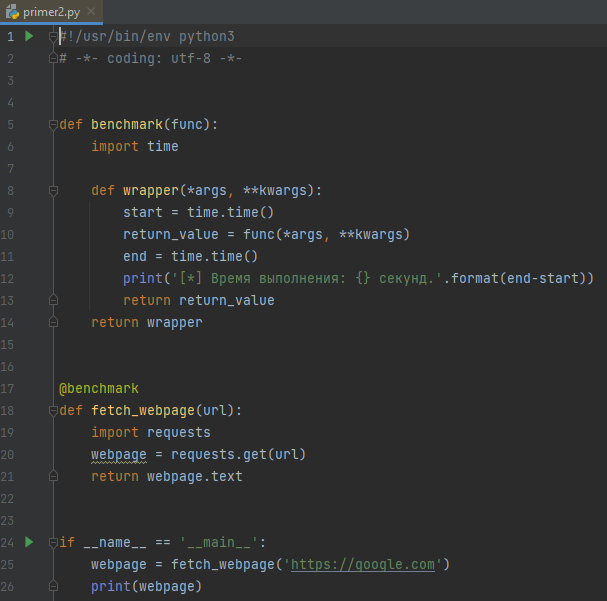


Рисунок 3 – Код второго примера

1. Приступил к выполнению индивидуального задания.

**Индивидуальное задание. Вариант 1.**

Условие: 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.

1. Объявил внешнюю функцию func\_show, в ней вызвал функцию обертку, где обратился к оригинальной функции для получения результата.
2. Объявил декоратор с именем get\_sq.
3. Сделал вызов декоративной функции.

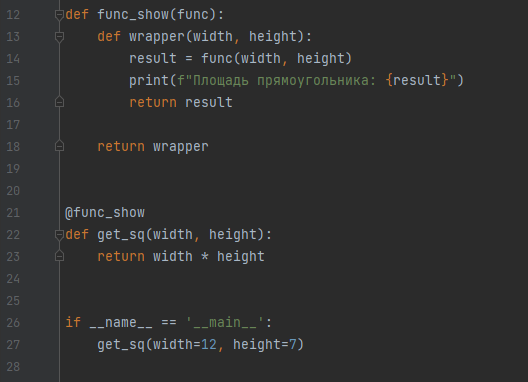


Рисунок 4 – Код выполненного результата

Рисунок 5 – 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.

1. 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.

1. Каково назначение функций высших порядков?

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

1. 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.

1. Какова структура декоратора функций?

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

1. 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 приобрел навыки по работе с декораторами функций при написании программ с помощью языка программирования Python версии 3.x.