

Министерство 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 №2.12
дисциплины «Основы кроссплатформенного программирования»
Вариант ____

Выполнил:
Дмитриченко Марат Заурович
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Декораторы функций в языке Python.

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

Ход работы

Условие примера: посмотреть и понять, как работает декоратор

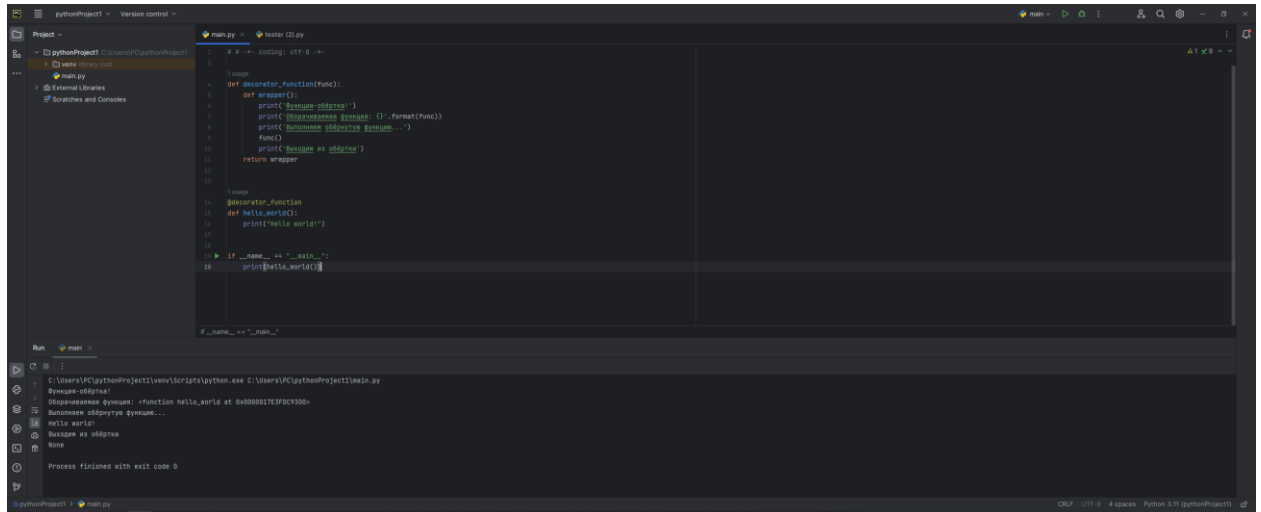


Рисунок 1. Реализация примера

Индивидуальное задание

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.

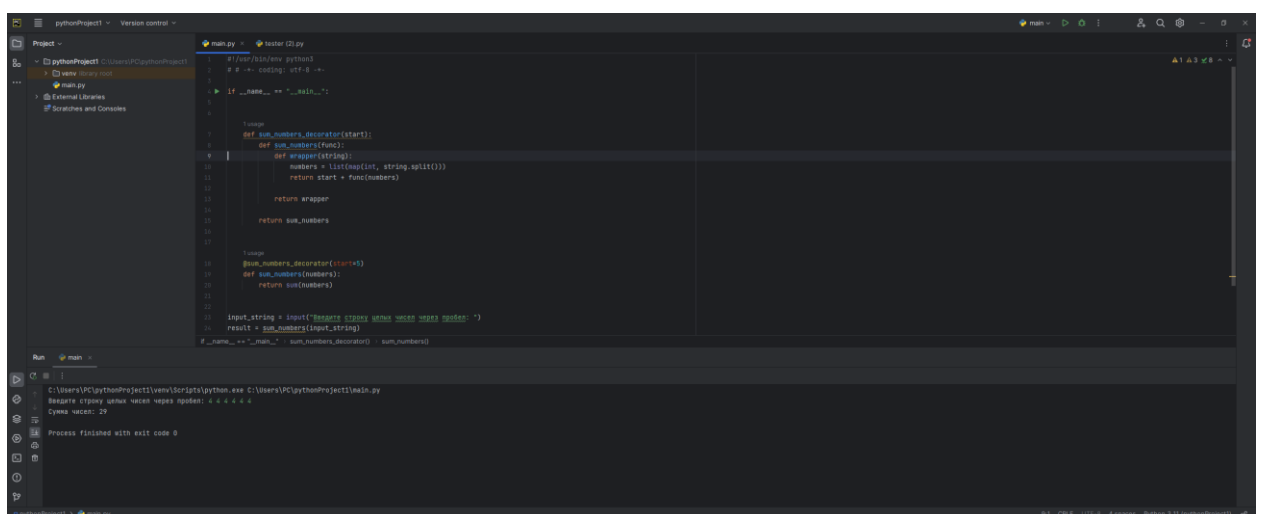


Рисунок 2. Реализация индивидуального задания

Ответы на контрольные вопросы

1. Что такое декоратор?

Декоратор - это функция, которая принимает другую функцию и расширяет ее функциональность, не изменяя ее исходный код. Декоратор позволяет добавлять новое поведение для функции, не затрагивая ее исходный код, что делает код более читаемым, понятным и модульным.

2. Почему функции являются объектами первого класса?

Функции являются объектами первого класса, потому что в Python они могут быть созданы, использованы и переданы как любой другой объект, например, числа, строки или списки. Функции могут быть присвоены переменной, переданы как аргументы в другую функцию, возвращены из функции и использованы в выражениях.

3. Каково назначение функций высших порядков?

Функции высших порядков - это функции, которые принимают одну или несколько функций как аргументы, либо возвращают другую функцию в качестве результата. Назначение функций высших порядков заключается в создании абстракций, которые могут использоваться для управления поведением других функций, делая код более гибким и модульным.

4. Как работают декораторы?

При использовании декоратора, функция или метод класса передаются в качестве аргумента в другую функцию, которая и выполняет нужное действие, дополняя тем самым исходную функциональность. Для создания декоратора в Python используют функцию, которая принимает в качестве аргументов другую функцию и возвращает функцию, которая представляет из себя декорированную версию первоначальной функции.

5. Какова структура декоратора функций?

Структура декоратора функций в Python состоит из двух функций:

1) Внешняя функция, которая принимает в качестве аргумента функцию, которую необходимо декорировать, и возвращает внутреннюю функцию.

2) Внутренняя функция, которая принимает те же аргументы, что и декорируемая функция, и содержит вызов декорируемой функции. Эта

функция может изменять аргументы, передаваемые декорируемой функции, а также результат, возвращаемый этой функцией.

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

В Python параметры декоратору могут быть переданы с помощью вложенной функции-обертки (wrapper), которая будет вызвана после декорирования функции, но перед выполнением ее кода.

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