

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 №15
дисциплины «Программирование на Python»
Вариант 5.

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент, доцент кафедры
инфокоммуникаций

(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.

Порядок выполнения работы.

1. Было выполнено индивидуальное задание согласно варианту 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.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def sum_decorator(start):
6      """
7      Декоратор для функции, который принимает начальное значение суммы.
8      """
9      def decorator(func):
10         def wrapper(string_of_numbers):
11             numbers = [int(num) for num in string_of_numbers.split()]
12             result_sum = func(numbers)
13             return result_sum + start
14         return wrapper
15     return decorator
16
17
18
19
20 @sum_decorator(start=5)
21 def calculate_sum(numbers):
22     """
23     Функция для подсчета суммы списка чисел.
24     """
25     return sum(numbers)
26
27
28 if __name__ == "__main__":
29     input_string = input("Введите строку целых чисел через пробел: ")
30
31     result = calculate_sum(input_string)
32
33     print(f"Сумма чисел с учетом начального значения: {result}")
34
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
python3: error: SDK "macosx" cannot be located
Введите строку целых чисел через пробел: 12 2 3 4 1
Сумма чисел с учетом начального значения: 27
(base) mikheeva@MacBook-Pro-Elena Python_2.12 %
```

Рисунок 1. Программа для индивидуального задания

Ответы на 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 они могут быть присвоены переменной, переданы в качестве аргумента другой функции, возвращены из функции и сохранены в структурах данных.

3. Каково назначение функций высших порядков?

могут принимать другие функции в качестве аргументов или возвращать их в качестве результатов. Они позволяют абстрагировать действия и оперировать функциями, как данными.

4. Как работают декораторы?

Работают путем обертывания другой функции, позволяя добавлять новое поведение к этой функции без изменения ее кода. Декораторы принимают функцию в качестве аргумента, возвращают другую функцию и обычно используются с символом `@`.

5. Какова структура декоратора функций?

Структура декоратора функций обычно выглядит следующим образом:

```
def decorator_function(original_function):  
    def wrapper_function(*args, **kwargs):  
        # Добавление нового поведения  
        return original_function(*args, **kwargs)  
    return wrapper_function
```

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

Параметры могут быть переданы декоратору, а не декорируемой функции, путем добавления еще одного уровня вложенной функции в

декораторе. Этот уровень может принимать параметры и передавать их в обернутую функцию.