

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.12 Декораторы
функций в языке Python»**

**ОТЧЕТ
по лабораторной работе №15
дисциплины
«Основы программной инженерии»**

Выполнил:

Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 2.12 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.

Ход работы:

Индивидуальное задание

8. Объявите функцию, которая вычисляет площадь круга и возвращает вычисленное значение. В качестве аргумента ей передается значение радиуса. Определите декоратор для этой функции, который выводит на экран сообщение: «Площадь круга равна = <число>». В строке выведите числовое значение с точностью до сотых. Примените декоратор к функции и вызовите декорированную функцию.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def output_circle(func):
8      def f(*args):
9          result = func(*args)
10         print("Площадь круга равна:", '%.2f' %result)
11
12     return f
13
14
15  @output_circle
16  def circle(r):
17      return math.pi * r * r
18
19
20  if __name__ == "__main__":
21      circle(3)
22
```

Рисунок 1 – Код программы для индивидуального задания

```
E:\GitHub\laba15\user\Scripts\pyth
Площадь круга равна: 28.27

Process finished with exit code 0
```

Рисунок 2 – Результат работы программы

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

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? Тот факт, что всё является объектами, 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. Из определения в Википедии: 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 оборачивают другие функции, расширяя её возможности без изменений функционального кода.

5. Какова структура декоратора функций?

```
def decorator_function(func):  
    def wrapper():  
        print('Функция-обёртка!')  
        print('Оборачиваемая функция: {}'.format(func))  
        print('Выполняем обёрнутую функцию...')  
        func()  
        print('Выходим из обёртки')  
    return wrapper
```

6. Самостоятельно изучить как можно передать параметры декоратору, а не декорируемой функции? Декоратор может принимать функции и возвращать их, чтобы передать параметры декоратору, а не декорируемой функции, можно поместить декоратор внутрь обычной функции, которая возвращает декоратор. И при вызове обычной функции передать в неё параметры.