

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО - КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
по лабораторной работе №6
Дисциплина: «Языки программирования»

Выполнил: студентка 2 курса
группы ИТС-б-о-20-1
Игнатова Елизавета Сергеевна

Проверил:
к.ф.-м.н., доцент
кафедры инфокоммуникаций
Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2022

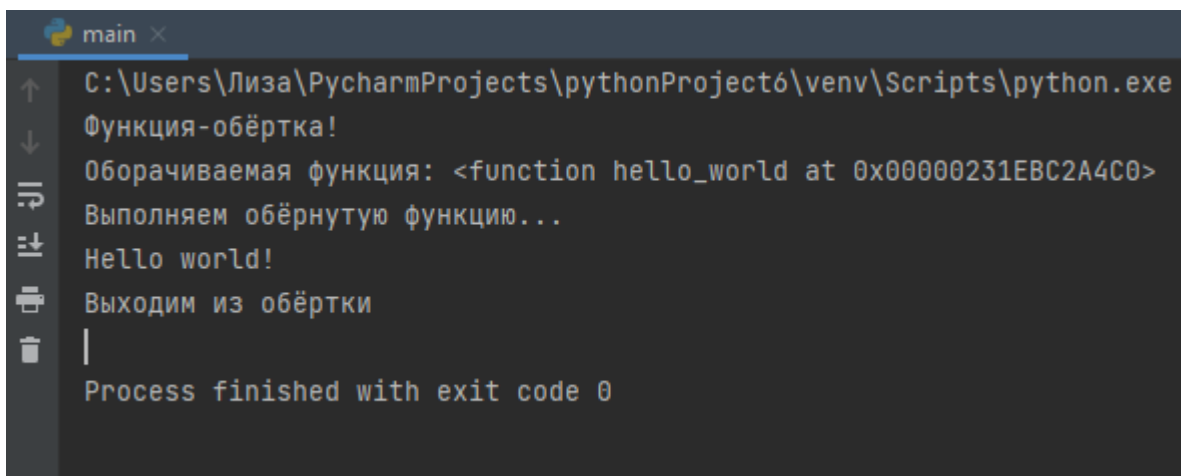
Цель работы: приобрести навыки по работе с декораторами функций при написании программ с помощью языка программирования Python.

Ссылка на репозиторий: <https://github.com/Nebula139/Sky2.6>

Проработан пример лабораторной

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def decorator_function(func):
5          def wrapper():
6              print('Функция-обёртка!')
7              print('Оборачиваемая функция: {}'.format(func))
8              print('Выполняем обёрнутую функцию...')
9              func()
10             print('Выходим из обёртки')
11         return wrapper
12
13
14     @decorator_function
15     def hello_world():
16         print('Hello world!')
17
18
19     hello_world()
20     |
```

Рисунок 1 – код примера



```
main x
C:\Users\Лиза\PycharmProjects\pythonProject6\venv\Scripts\python.exe
Функция-обёртка!
Оборачиваемая функция: <function hello_world at 0x00000231EBC2A4C0>
Выполняем обёрнутую функцию...
Hello world!
Выходим из обёртки
|
Process finished with exit code 0
```

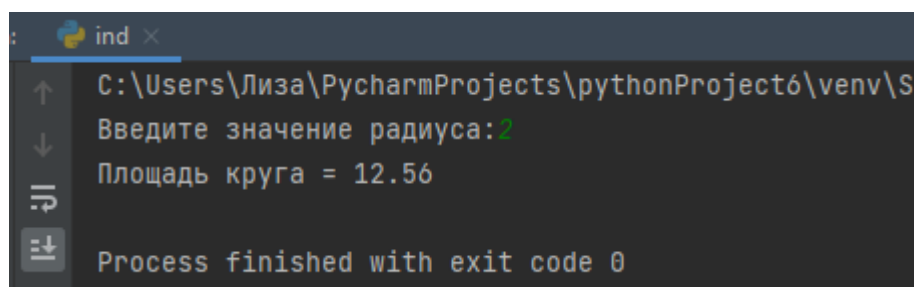
Рисунок 2 – результат работы кода

Индивидуальное задание. Вариант – 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 =». В строке выведите числовое значение с точностью до сотых. Примените декоратор к функции и вызовите декорированную функцию

```
1  ▶  #!/usr/bin/venv python3
2      # -*- coding: utf-8 -*-
3
4      def decorator(func):
5          def decorator1(x):
6              c = "Площадь круга"
7              ret = func(x)
8              print(f'{c} = {ret}')
9              return ret
10         return decorator1
11
12
13     @decorator
14     def add(x):
15         pi = 3.14
16         return x * x * pi
17
18
19  ▶  if __name__ == "__main__":
20      r = float(input("Введите значение радиуса:"))
21      add(r)
22
```

Рисунок 3 – код программы



```
ind x
C:\Users\Лиза\PycharmProjects\pythonProject6\venv\Scripts
Введите значение радиуса:2
Площадь круга = 12.56
Process finished with exit code 0
```

Рисунок4 – результат работы кода

Контрольные вопросы:

1. Что такое декоратор?

Декоратор – это функция, которая позволяет обернуть другую функцию для расширения ее функциональности без непосредственного изменения ее кода.

2. Почему функции являются объектами первого класса?

Объектами первого класса в контексте конкретного языка программирования называются элементы, с которыми можно делать все то же, что и с любыми другими объектами: передавать как параметр, возвращать из функции и присваивать переменной

3. Каково назначение функций высших порядков?

Функции высших порядков – это функции, которые могут принимать в качестве аргументов и возвращать другие функции.

4. Как работают декораторы?

Они позволяют обворачивать другие функции при помощи символа @.

5. Какова структура декоратора функций?

Для начала записывается декоратор. Далее идет вызов с помощью @, и потом основная функция, которую обворачивает декоратор.

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

Если декоратор вызван без скобок, то единственным его параметром будет декорируемая функция (что подпадает под определение чистого декоратора), которая попадет в переменную `_func`, а иначе она будет равна `None`. Если же, мы передаем только именованный параметр, например, `n=10`,

то остается `_func = None` и благодаря этому декоратор понимает вызвали его с параметром или нет.

Вывод по работе: в ходе лабораторной работы были приобретены навыки по работе с декораторами функций при написании программ с помощью языка программирования Python.