

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Основы кроссплатформенного программирования
Отчет по лабораторной работе №2.12**

Декораторы функций в языке Python

Выполнил студент группы
ИТС-б-о-21-1 (2)

Якупов Э.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил к.т.н., доцент

Кафедры инфокоммуникаций

Воронкин Р.А.

(подпись)

Ставрополь 2022

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

Ссылка на репозиторий - <https://github.com/Blekroyt/Fox2.git>

Ход работы:

Пример 1.

```
>>> def hello_world():
...     print('Hello world!')
...
>>> type(hello_world)
<class 'function'>
>>> class Hello:
...     pass
...
>>> type(Hello)
<class 'type'>
>>> type(10)
<class 'int'>
>>> hello = hello_world
>>> hello()
Hello world!
```

Рисунок 1. Результат выполнения программы

Пример 2.

```
>>> 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
...
>>> @decorator_function
... def hello_world():
...     print('Hello world!')
...
>>> hello_world()
Функция-обёртка!
Оборачиваемая функция: <function hello_world at 0x0000019900CAFAC0>
Выполняем обёрнутую функцию...
Hello world!
Выходим из обёртки
```

Рисунок 2. Результат выполнения программы

Индивидуальное задание. 8 вариант. 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  # -*- кодировка: utf-8 -*-
3  # 8. Объявите функцию, которая вычисляет площадь круга и возвращает вычисленное значение.
4  # радиуса. Определите декоратор для этой функции, который выводит на экран сообщение:
5  # «Площадь круга равна = <число>». В строке выведите числовое значение с точностью до
6  # сотых. Примените декоратор к функции и вызовите декорированную функцию.
7  import sys
8
9
10 def print_info(func):
11     def f(*args, **kwargs):
12         result = func(*args, **kwargs)
13         print(f"Площадь круга равна = {result:1.2f}")
14
15     return f
16
17
18 @print_info
19 def area(x):
20     return 3.1415926 * x * x
21
22
23 r = float(input("Введите радиус"))
24 area(r)
25 |
```

Рисунок 3. код программы



Рисунок 4. результат программы

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?

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.

В Python всё является объектом, а не только объекты, которые вы создаёте из классов. Это значит, что в Python всё это — объекты:

- числа;
- строки;
- классы;
- функции.

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?

Декоратор — это функция, которая позволяет обернуть другую функцию с помощью символа «@»

5. Какова структура декоратора функций?

Сначала записывается функция — декоратор. Потом идет его вызов с помощью @, а затем основная функция, которую оборачивает декоратор.

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

Используя замыкание функций.

Вывод: в ходе лабораторной работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.