

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 (15)**  
**дисциплины**  
**«Программирование на Python»**

Выполнил:  
Ибрагимов Муса Айнуудинович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

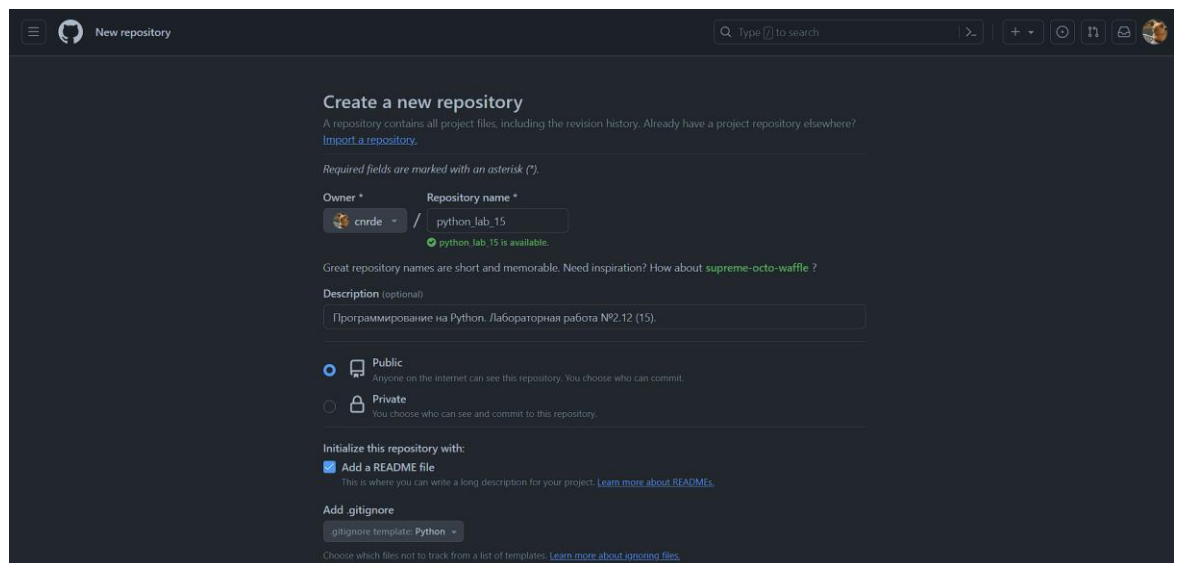
Ставрополь, 2024 г.

Тема: Функции с переменным числом параметров в Python.

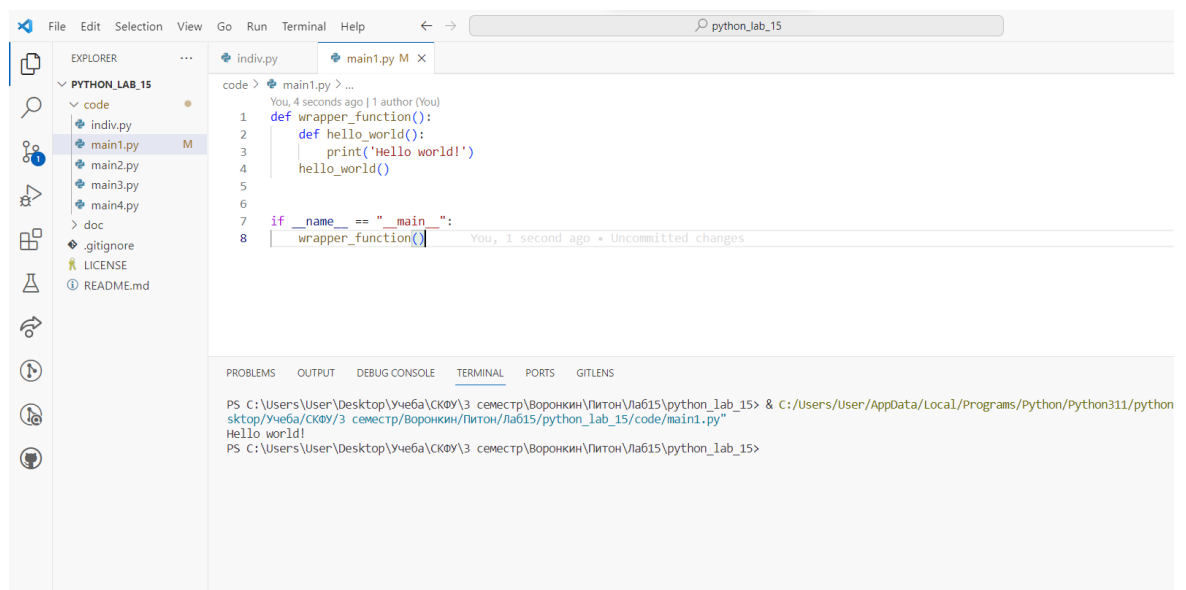
Цель: Приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

### Практическая часть:

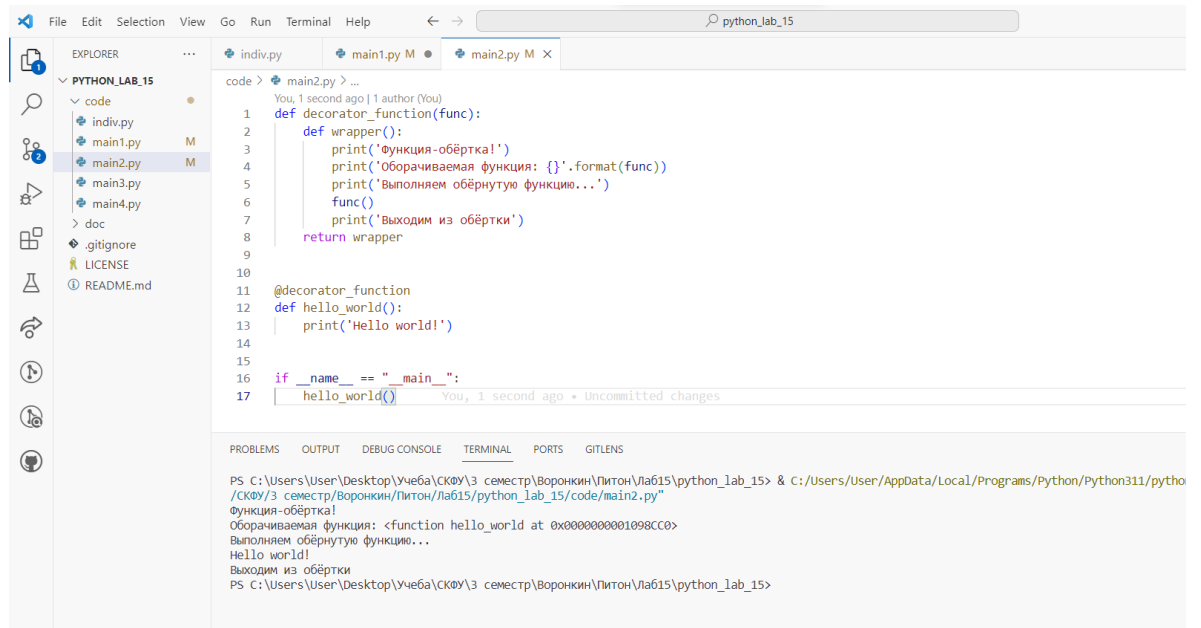
1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



3. Проработал первый пример.



## 4. Проработал второй пример.



```
File Edit Selection View Go Run Terminal Help python_lab_15
```

EXPLORER

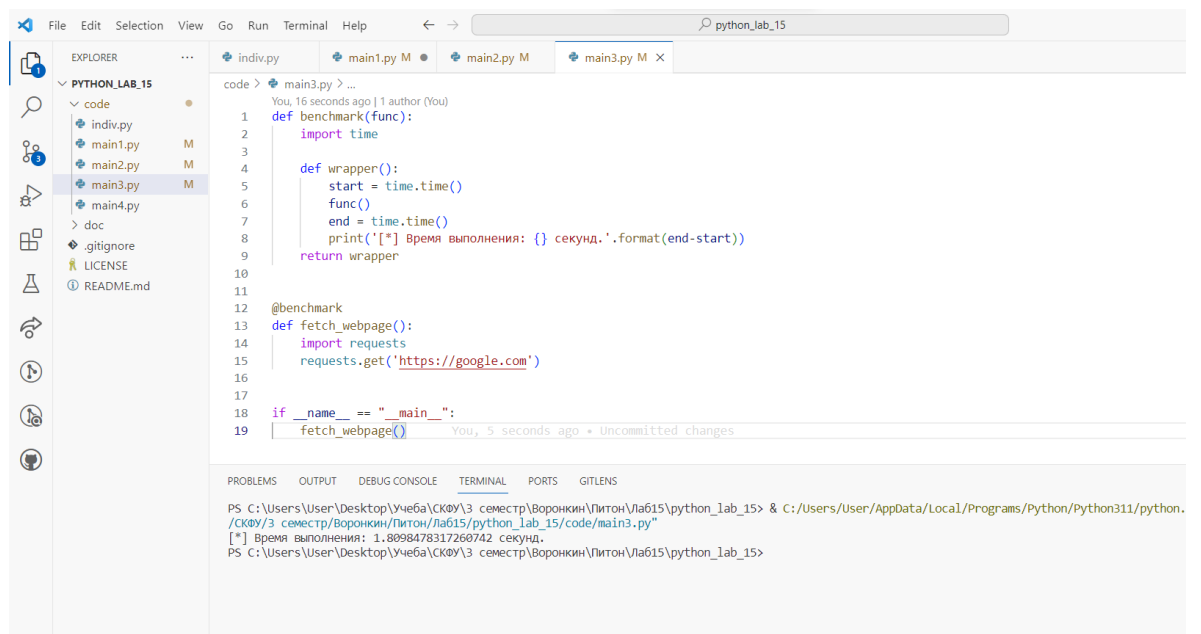
- PYTHON\_LAB\_15
  - code
    - indiv.py
    - main1.py M
    - main2.py M
    - main3.py
    - main4.py
  - doc
  - .gitignore
  - LICENSE
  - README.md

```
code > main2.py > ...
You, 1 second ago | 1 author (You)
1 def decorator_function(func):
2     def wrapper():
3         print('функция-обёртка!')
4         print('оборачиваемая функция: {}'.format(func))
5         print('выполняем обёрнутую функцию...')
6         func()
7         print('выходим из обёртки')
8     return wrapper
9
10
11 @decorator_function
12 def hello_world():
13     print('Hello world!')
14
15
16 if __name__ == "__main__":
17     hello_world()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\User\Desktop\Учеба\СКОУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15> & C:/Users/User/AppData/Local/Programs/Python/Python311/python
/СКОУ/3 семестр\Воронкин\Питон\Лаб15\python_lab_15/code/main2.py"
функция-обёртка!
оборачиваемая функция: <function hello_world at 0x0000000001098C0>
выполняем обёрнутую функцию...
Hello world!
выходим из обёртки
PS C:\Users\User\Desktop\Учеба\СКОУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15>
```

## 5. Проработал третий пример.



```
File Edit Selection View Go Run Terminal Help python_lab_15
```

EXPLORER

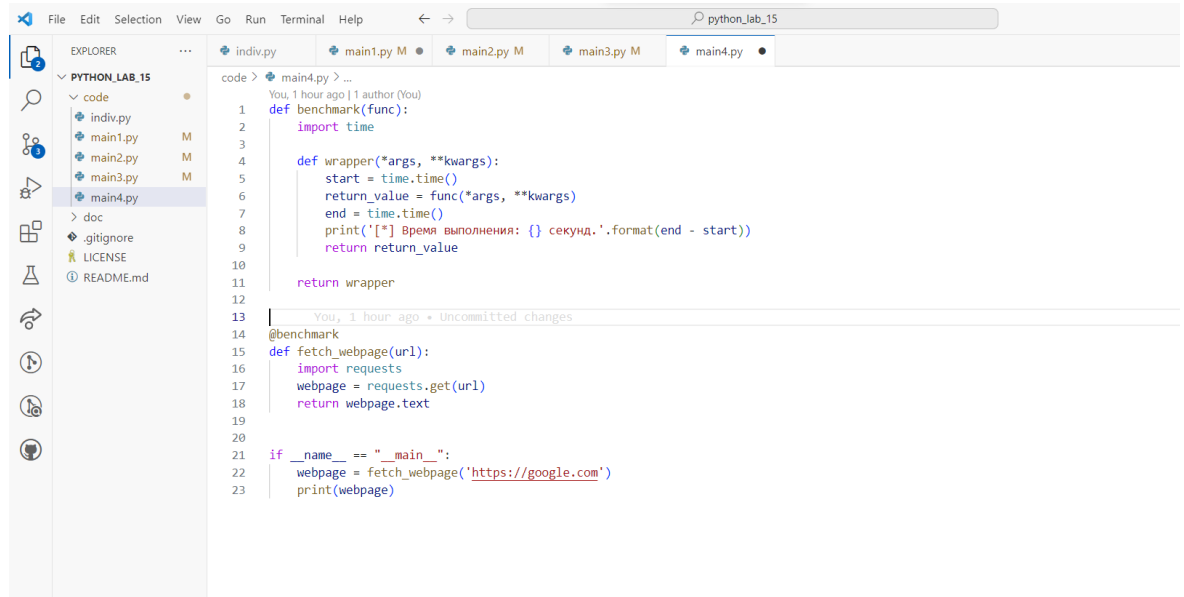
- PYTHON\_LAB\_15
  - code
    - indiv.py
    - main1.py M
    - main2.py M
    - main3.py M
    - main4.py
  - doc
  - .gitignore
  - LICENSE
  - README.md

```
code > main3.py > ...
You, 16 seconds ago | 1 author (You)
1 def benchmark(func):
2     import time
3
4     def wrapper():
5         start = time.time()
6         func()
7         end = time.time()
8         print('[*] Время выполнения: {} секунд.'.format(end-start))
9     return wrapper
10
11
12 @benchmark
13 def fetch_webpage():
14     import requests
15     requests.get('https://google.com')
16
17
18 if __name__ == "__main__":
19     fetch_webpage()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\User\Desktop\Учеба\СКОУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.
/СКОУ/3 семестр\Воронкин\Питон\Лаб15\python_lab_15/code/main3.py"
[*] Время выполнения: 1.8098478317260742 секунд.
PS C:\Users\User\Desktop\Учеба\СКОУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15>
```

## 6. Проработал четвертый пример.

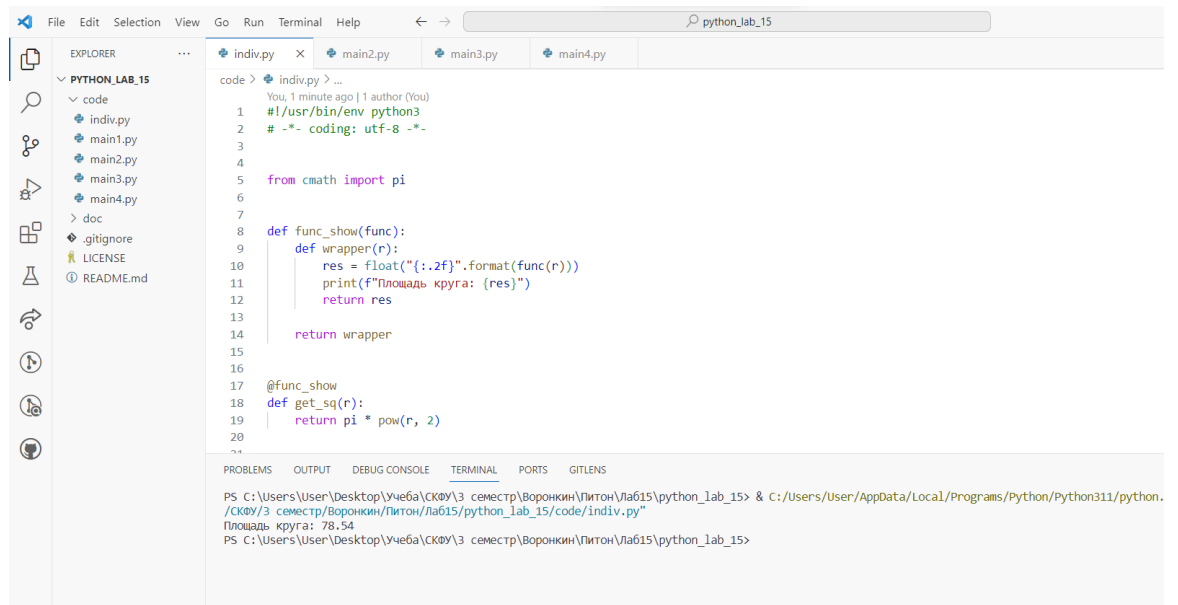


```
File Edit Selection View Go Run Terminal Help
python_lab_15

EXPLORER
PYTHON_LAB_15
├── code
│   ├── indiv.py
│   ├── main1.py
│   ├── main2.py
│   ├── main3.py
│   └── main4.py
├── doc
├── .gitignore
├── LICENSE
└── README.md

code > main4.py > ...
You, 1 hour ago | 1 author (You)
1 def benchmark(func):
2     import time
3
4     def wrapper(*args, **kwargs):
5         start = time.time()
6         return_value = func(*args, **kwargs)
7         end = time.time()
8         print('[*] Время выполнения: {} секунд.'.format(end - start))
9         return return_value
10
11     return wrapper
12
13 You, 1 hour ago * Uncommitted changes
14 @benchmark
15 def fetch_webpage(url):
16     import requests
17     webpage = requests.get(url)
18     return webpage.text
19
20
21 if __name__ == "__main__":
22     webpage = fetch_webpage('https://google.com')
23     print(webpage)
```

7. Выполнил 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 = ». В строке выведите числовое значение с точностью до сотых. Примените декоратор к функции и вызовите декорированную функцию.



```
File Edit Selection View Go Run Terminal Help
python_lab_15

EXPLORER
PYTHON_LAB_15
├── code
│   ├── indiv.py
│   ├── main1.py
│   ├── main2.py
│   ├── main3.py
│   └── main4.py
├── doc
├── .gitignore
├── LICENSE
└── README.md

code > indiv.py > ...
You, 1 minute ago | 1 author (You)
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 from cmath import pi
6
7
8 def func_show(func):
9     def wrapper(r):
10         res = float("{:.2f}".format(func(r)))
11         print(f"Площадь крyга: {res}")
12         return res
13
14     return wrapper
15
16
17 @func_show
18 def get_sq(r):
19     return pi * pow(r, 2)
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
PS C:\Users\User\Desktop\Учеба\СкФУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15> & C:\Users\User\AppData\Local\Programs\Python\Python311\python.
/СкФУ/3 семестр\Воронкин\Питон\Лаб15\python_lab_15/code/indiv.py
Площадь крyга: 78.54
PS C:\Users\User\Desktop\Учеба\СкФУ\3 семестр\Воронкин\Питон\Лаб15\python_lab_15>
```

## **Ответы на контрольные вопросы**

### **1. Что такое замыкание?**

Замыкание – это функция, в теле которой присутствуют ссылки на переменные, объявленные вне тела этой функции.

### **2. Как реализованы замыкания в языке программирования Python?**

Замыканием в языке Python называется функция, вложенная в другую функцию и использующая переменные внешней функции.

### **3. Что подразумевает под собой область видимости Local?**

Переменные с областью видимости Local (локальные переменные) могут быть использованы только внутри того блока кода, где она была объявлена.

### **4. Что подразумевает под собой область видимости Enclosing?**

Для вложенных функций переменные из функции более высокого уровня имеют данную область видимости.

### **5. Что подразумевает под собой область видимости Global?**

Область видимости Global означает, что данная переменная может быть использована (видна) во всём модуле (файле с расширением .py).

### **6. Что подразумевает под собой область видимости Build-in?**

Это переменный уровень интерпретатора. Для их использования не нужно импортировать модули.

**Вывод:** Приобрел навыки по работе с функциями с переменным числом параметров при написании программ помощью языка программирования Python версии 3.x.