

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное  
учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт цифрового развития**

**Кафедра информационных систем и технологий**

Отчет по лабораторной работе №2,8.

Дисциплина: «Основы программной инженерии»

**Выполнил:**

Студент группы ПИЖ-б-о-22-  
1,

направление подготовки:  
09.03.04

«Программная инженерия»

ФИО: Рядская Мария

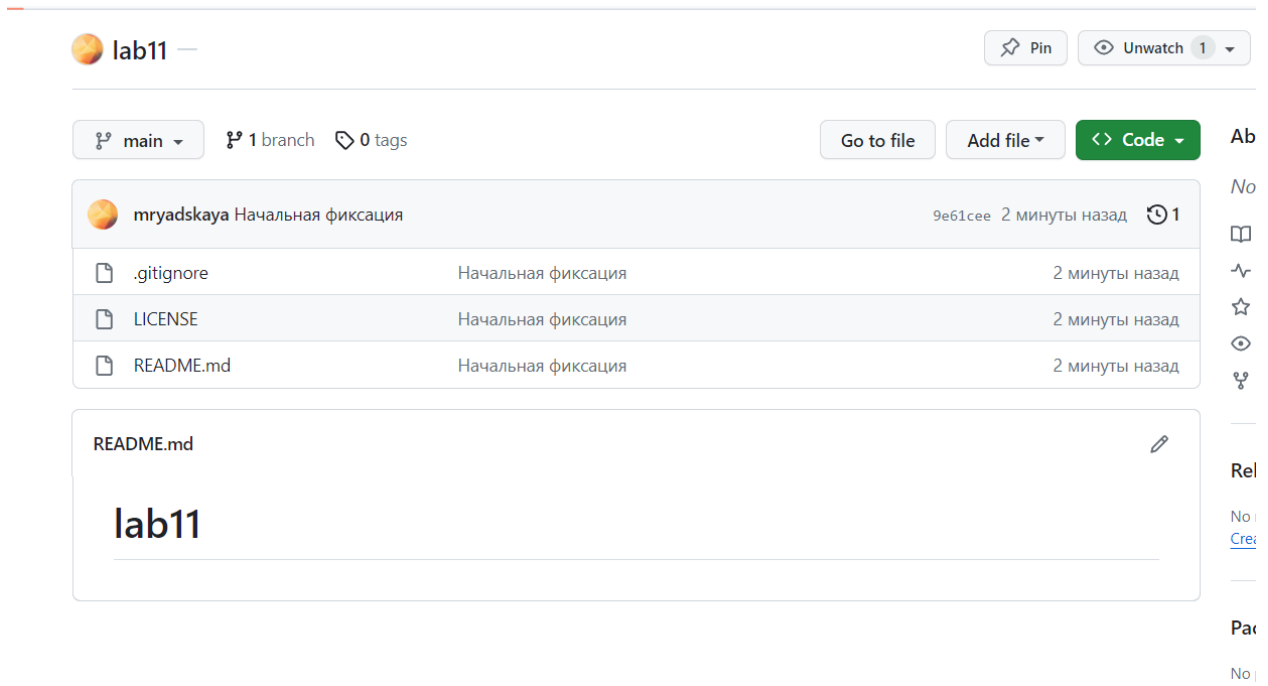
Александровна **Проверил:**

Воронкин Р. А.

Ставрополь 2023

Тема: Лабораторная работа 2.8 Работа с функциями в языке Python.

Цель работы: приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.  
Создала репозиторий на git.hub.



- Организовать свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\git1\lab10>cd ..

C:\git1>git clone https://github.com/mryadskaya/lab11.git
Cloning into 'lab11'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\git1>cd lab11

C:\git1\lab11>git branch develop

C:\git1\lab11>git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/mryadskaya/lab11/pull/new/develop
remote:
To https://github.com/mryadskaya/lab11.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.

C:\git1\lab11>git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

C:\git1\lab11>
```

Проработать примеры из методички.

```
pr1.py  пример.py ×
1  ✓ 1 #!/usr/bin/env python3
2  2 # -*- coding: utf-8 -*-
3
4  3 import sys
5
6  4 def get_worker():
7  5     """
8  6     Запросить данные о работнике.
9  7     """
10  8     name = input("Фамилия и инициалы? ")
11  9     post = input("Должность? ")
12 10     year = int(input("Год поступления? "))
13
14 11     # Создать словарь.
15 12     return {
16 13         'name': name,
17 14         'post': post,
18 15         'year': year,
19 16     }
20
21
22 17 def display_workers(staff):
23 18     """
24 19     Отобразить список работников.
25 20     """
26 21     # Проверить, что список работников не пуст.
27 22     if staff:
28 23         # Заголовок таблицы.
29 24         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
30 25             *args: '-' * 4,
31 26
32 27 if __name__ == '__main__':
```

```
C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab4\PyCharm\пример.py
>>> add
Фамилия и инициалы? ryadskaya
Должность? student
Год поступления? 2022
>>> add
Фамилия и инициалы? asdfghjk
Должность? oiy
Год поступления? 852
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | asdfghjk                 | oiy                 | 852           |
|  2 | ryadskaya                 | student             | 2022          |
+-----+-----+-----+-----+
Список работников пуст.
>>>
```

## Задание 1

Основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции `test()` и инструкции `if __name__ == '__main__':`. В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция `positive()`, тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция `negative()`, ее тело содержит выражение вывода на экран слова "Отрицательное"

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  def test():
7      number = int(input("Введите целое число: "))
8      if number > 0:
9          positive()
10     elif number < 0:
11         negative()
12
13
14  def positive():
15      print("Положительное")
16
17
18  def negative():
19      print("Отрицательное")
20
21
22  if __name__ == '__main__':
23      test()
24
25  if __name__ == '__main__':
```

Run задание1 x

C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab4\PyCharm\задание1.py  
Введите целое число: 55  
Положительное

## Задание 2

В основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле  $\pi r^2$ . В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле  $2\pi rh$ , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def cylinder():
8
9      def circle(r):
10         return math.pi * r**2
11
12         r = float(input("Введите радиус цилиндра: "))
13         h = float(input("Введите высоту цилиндра: "))
14         side_area = 2 * math.pi * r * h
15         full_area = side_area + 2 * circle(r)
16
17         choice = input("Хотите получить только площадь боковой поверхности цилиндра? (да/нет): ")
18         if choice.lower() == "да":
19             print(f"Площадь боковой поверхности цилиндра: {side_area}")
20         else:
21             print(f"Полная площадь цилиндра: {full_area}")
22
23
24 if __name__ == '__main__':
25     cylinder()
```

Run задание2 x

C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab4\PyCharm\задание2.py

Введите радиус цилиндра: 12

Введите высоту цилиндра: 18

Хотите получить только площадь боковой поверхности цилиндра? (да/нет): да

Площадь боковой поверхности цилиндра: 1357.1680263507906


.




### Задание 3

Напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.

```
1  ✓#!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import math
5
6  ✓def proverka():
7      n = int(input())
8      s = n
9
10     ✓while n != 0:
11         n = int(input())
12         s += n
13
14     print(s)
15
16  ▶ if __name__ == '__main__':
17     proverka()
```

if \_\_name\_\_ == '\_\_main\_\_'

Run  задание 3 ×

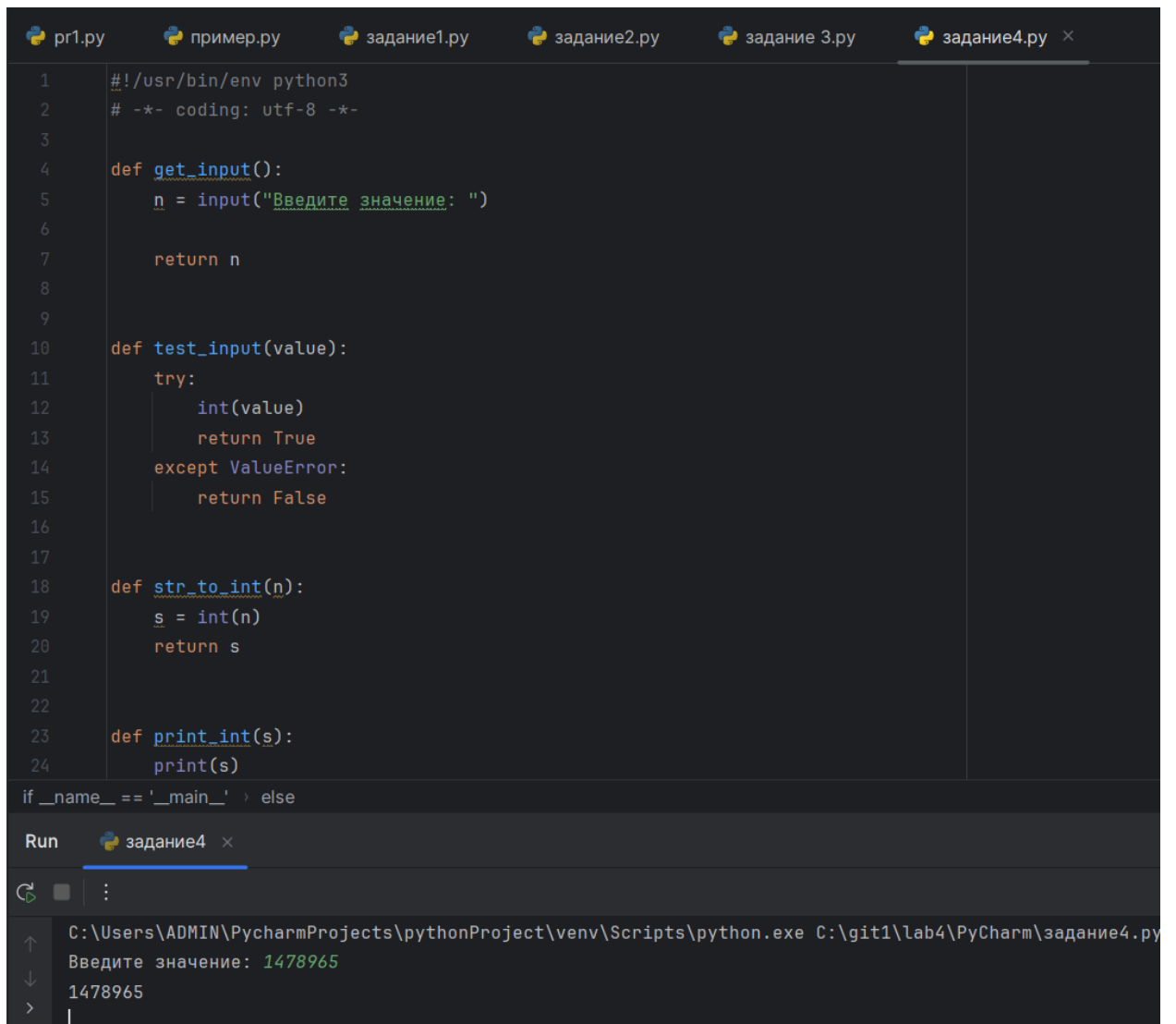
↻   

↑ 12  
↓ 96  
↺ 78  
↻ 45  
⬇ 20  
🖨 12  
🗑 0  
272

## Задание 4

Напишите программу, в которой определены четыре Функции





The image shows a PyCharm IDE window with several tabs: 'pr1.py', 'пример.py', 'задание1.py', 'задание2.py', 'задание 3.py', and 'задание4.py'. The 'задание4.py' tab is active, displaying a Python script. The script defines four functions: `get_input()` for user input, `test_input(value)` for validating if a value is an integer, `str_to_int(n)` for converting a string to an integer, and `print_int(s)` for printing an integer. The main block of the script uses these functions to prompt the user for a value, test it, convert it, and print it. The output window at the bottom shows the command prompt path and the execution of the script, which prompts the user to enter a value. The user has entered '1478965', which is displayed in green text.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def get_input():
5      n = input("Введите значение: ")
6
7      return n
8
9
10 def test_input(value):
11     try:
12         int(value)
13         return True
14     except ValueError:
15         return False
16
17
18 def str_to_int(n):
19     s = int(n)
20     return s
21
22
23 def print_int(s):
24     print(s)
25
26 if __name__ == '__main__':
27     else
28
29 Run  задание4  ×
30
31 C:\Users\ADMIN\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\git1\lab4\PyCharm\задание4.py
32 Введите значение: 1478965
33 1478965
34 >
```

1. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.

```
pr1.py  пример.py  задание1.py  задание2.py  задание 3.py  задание4.py  индивиду...

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  def help1():
7      """
8      Функция для вывода списка команд
9      """
10     # Вывести справку о работе с программой.
11     print("Список команд:\n")
12     print("add - добавить информацию;")
13     print("list - вывести список;")
14     print("select <тип> - вывод на экран фамилия, имя; знак Зодиака; дата рождения ")
15     print("help - отобразить справку;")
16     print("exit - завершить работу с программой.")
17
18
19  def add1():
20      """
21      Функция для добавления информации о новых рейсах
22      """
23     # Запросить данные о работнике.
24     name = input("Фамилия и имя? ")
25     zodiac = input("Знак зодиака? ")
26     date = input("дата рождения? ")
27     # Создать словарь.
28     i = {
29         'name': name,
30         'zodiac': zodiac,
```

```
list()  > for idx, i in enumerate(birthda...

add - добавить информацию;
list - вывести список ;
select <тип> - вывод на экран фамилия, имя; знак Зодиака; дата рождения
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и имя? ryadsksysa
Знак зодиака? рак
дата рождения? 17.07.2005
>>> add
Фамилия и имя? svetloкова
Знак зодиака? рак
дата рождения? 17.07.2004
>>> add
Фамилия и имя? skichko
Знак зодиака? рыбы
дата рождения? 25.02.2003
>>> list
+-----+-----+-----+
| 1 | ryadsksysa | рак | 17.07.2005 |
| 2 | skichko | рыбы | 25.02.2003 |
| 3 | svetloкова | рак | 17.07.2004 |
+-----+-----+-----+
>>>
```

- Зафиксировал все изменения в github в ветке develop.
- Слила ветки.

```

C:\git1\lab11>git add .

C:\git1\lab11>git status
On branch develop
Your branch is up to date with 'origin/develop'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\265 3.
py"
    new file:   "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2651.p
y"
    new file:   "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.p
y"
    new file:   "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2654.p
y"
    new file:   "PyCharm/\320\270\320\275\320\264\320\270\320\262\320\270\320\264\32
1\203\320\260\320\273\321\214\320\275\320\276\320\265.py"
    new file:   "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\200.py"

C:\git1\lab11>git commit -m "сщхранение"
[develop 2086775] сщхранение
 6 files changed, 363 insertions(+)
 create mode 100644 "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\265 3.
py"
 create mode 100644 "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2651.p
y"
 create mode 100644 "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2652.p
y"
 create mode 100644 "PyCharm/\320\267\320\260\320\264\320\260\320\275\320\270\320\2654.p
y"
 create mode 100644 "PyCharm/\320\270\320\275\320\264\320\270\320\262\320\270\320\264\32
1\203\320\260\320\273\321\214\320\275\320\276\320\265.py"
 create mode 100644 "PyCharm/\320\277\321\200\320\270\320\274\320\265\321\200.py"

C:\git1\lab11>git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 4.49 KiB | 417.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mryadskaya/lab11.git
 9e61cee..2086775 develop -> develop

C:\git1\lab11>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

```

Вывод: приобрел навыки по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

Контрольные вопросы:

### 1. Каково назначение функций в языке программирования Python?

Функция в программировании представляет собой обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был

назван. При вызове происходит выполнение команд тела функции.  
Внедрение

функций позволяет решить проблему дублирования кода в разных местах программы. Благодаря им можно исполнять один и тот же участок кода не сразу,

а только тогда, когда он понадобится.

### 2. Каково назначение операторов `def` и `return`?

Оператор `def` в Python используется для определения функции. Он начинает заголовок функции и может принимать ноль или более аргументов, которые могут

использоваться в теле функции. Оператор `return` используется для возврата результата выполнения функции. Он может быть необязательным, так как функция может ничего не возвращать. Оператор `return` не только возвращает значение, но и производит выход из функции. Поэтому он должен определяться

после остальных инструкций. Если функция не возвращает никакого значения,

после оператора `return` не ставится никакого возвращаемого значения.

### 3. Каково назначение локальных и глобальных переменных при написании функций в Python?

Соответственно, локальные переменные видны только в локальной области видимости, которой может выступать отдельно взятая функция. Глобальные переменные видны во всей программе. "Видны" – значит, известны, доступны. К

ним можно обратиться по имени и получить связанное с ними значение. К глобальной переменной можно обратиться из локальной области видимости. К

локальной переменной нельзя обратиться из глобальной области видимости, потому что локальная переменная существует только в момент выполнения тела

функции. При выходе из нее, локальные переменные исчезают.

Компьютерная

память, которая под них отводилась, освобождается. Когда функция будет снова

вызвана, локальные переменные будут созданы заново.

4. Как вернуть несколько значений из функции Python?

В Питоне позволительно возвращать из функции несколько объектов, перечислив их через запятую после команды return.

5. Какие существуют способы передачи значений в функцию?

```
figure4 = cylinder(r=2, h=10)
```

```
def cylinder(h, r=1):
```

```
figure4 = cylinder(i, h)(передаются значения глобальных переменных)
```

6. Как задать значение аргументов функции по умолчанию?

```
def cylinder(h, r=1):
```

7. Каково назначение lambda-выражений в языке Python?

Lambda-выражения в Python, также известные как “анонимные функции”,

используются для создания небольших функций без необходимости

использования ключевого слова def. Они представляют собой компактный способ

определения функции.

8. Как осуществляется документирование кода согласно PEP257?

Все модули должны, как правило, иметь строки документации, и все функции и классы, экспортируемые модулем также должны иметь строки документации. Публичные методы (в том числе `__init__`) также должны иметь

строки документации. Пакет модулей может быть документирован в `__init__.py`.

Для согласованности, всегда используйте `"""triple double quotes"""` для строк документации. Многострочные строки документации состоят из однострочной

строки документации с последующей пустой строкой, а затем более подробным

описанием. Первая строка может быть использована автоматическими средствами индексации, поэтому важно, чтобы она находилась на одной строке и

была отделена от остальной документации пустой строкой. Первая строка может

быть на той же строке, где и открывающие кавычки, или на следующей строке.

Вся документация должна иметь такой же отступ, как кавычки на первой строке

9. В чем особенность однострочных и многострочных форм строк документации?

Однострочные строки документации используются для краткого описания функции, метода, класса или модуля, что делает и какие аргументы принимает.

Они заключаются в тройные кавычки и пишутся в императивной форме.

Многострочные строки документации используются для более подробного описания функции, метода, класса или модуля, включая их параметры, типы, возвращаемые значения, исключения, примеры и другие детали. Они также

закljučаются в тройные кавычки, но имеют определенный формат и стиль, в зависимости от выбранной конвенции.