

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 №11**  
**дисциплины «Основы программной инженерии»**

Выполнил:  
Матвеев Александр Иванович  
2 курс, группа ПИЖ-б-о-22-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка и  
сопровождение программного  
обеспечения», очная форма обучения

\_\_\_\_\_  
(подпись)

Проверил Воронкин Роман Александрович

\_\_\_\_\_  
(подпись)

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

Ставрополь, 2023 г.

## Тема: 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.

Ход работы.

1. Создание нового репозитория с лицензией MIT.


### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---


*Required fields are marked with an asterisk (\*).*

Owner \*

 SashkaHacker ▾

Repository name \*


laba10

 laba10 is available.


Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#) ?

Description (optional)

---

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python ▾


Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

 You are creating a public repository in your personal account.

---

Create repository

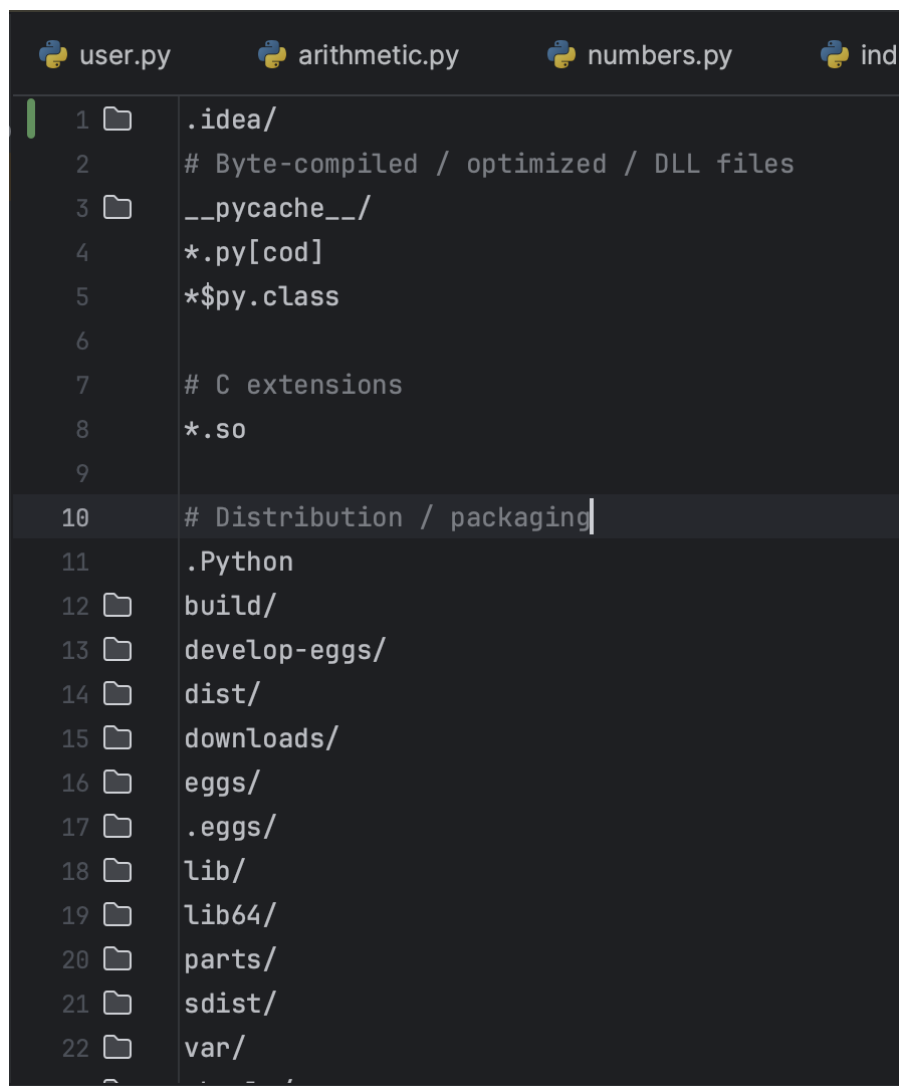
Рисунок 1 – Создание репозитория

2. Клонировал репозиторий на рабочий ПК.

```
Sashka@DESKTOP-U4RPSBI MINGW64 ~/Documents/GitHub
$ git clone https://github.com/SashkaHacker/laba10.git
Cloning into 'laba10'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.



The image shows a code editor window with a dark theme. At the top, there are four tabs: 'user.py', 'arithmetic.py', 'numbers.py', and 'ind'. The active tab is 'user.py'. The editor displays the content of a '.gitignore' file. The text is as follows:

```
1  .idea/
2  # Byte-compiled / optimized / DLL files
3  __pycache__/
4  *.py[cod]
5  *$py.class
6
7  # C extensions
8  *.so
9
10 # Distribution / packaging
11 .Python
12 build/
13 develop-eggs/
14 dist/
15 downloads/
16 eggs/
17 .eggs/
18 lib/
19 lib64/
20 parts/
21 sdist/
22 var/
```

Рисунок 3 – Файл .gitignore

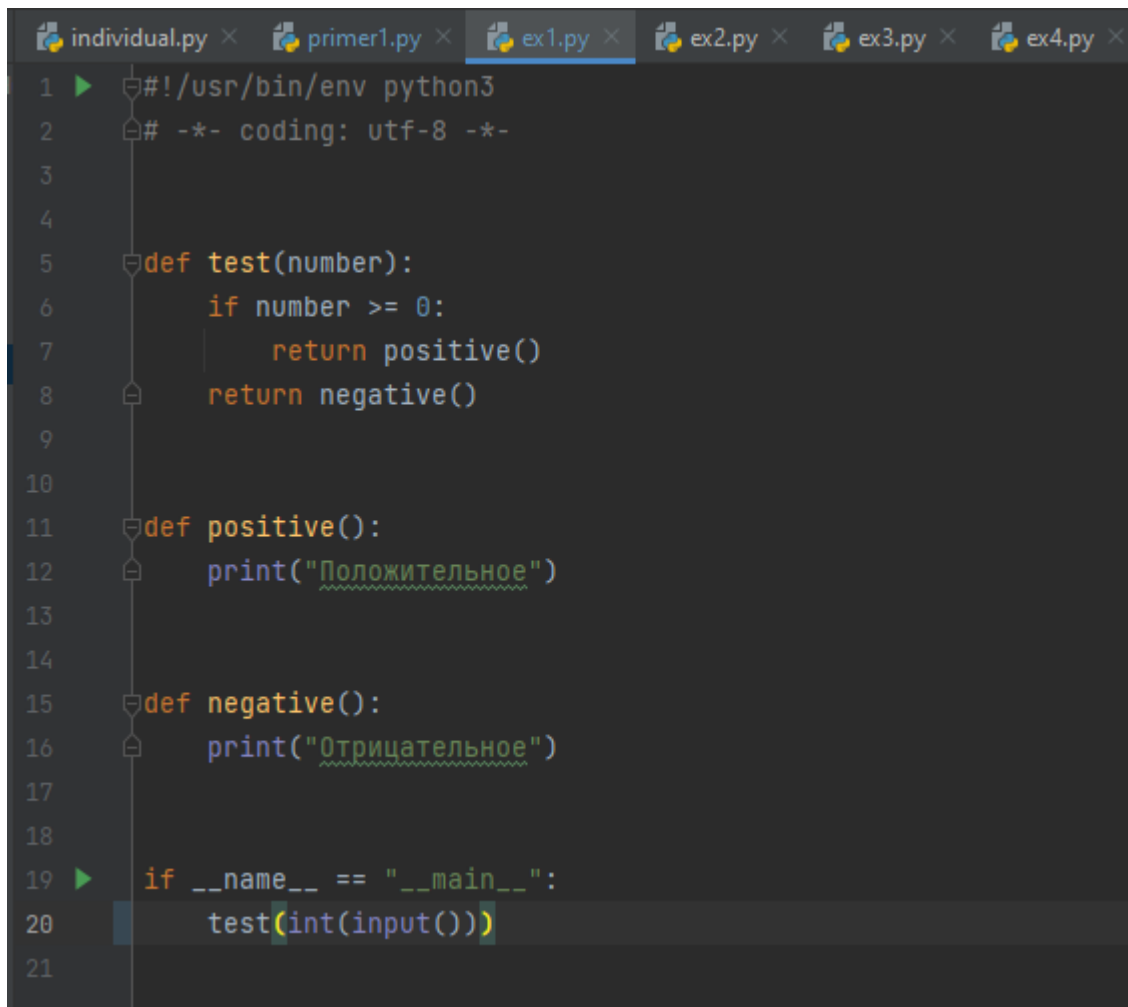
#### 4. Проработка задания №1 из методических указаний.

```
C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Sa
>>> add
Фамилия и инициалы? Матвеев А.И.
Должность? owner
Год поступления? 2020
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Матвеев А.И. | owner | 2020 |
+-----+-----+-----+-----+
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
```

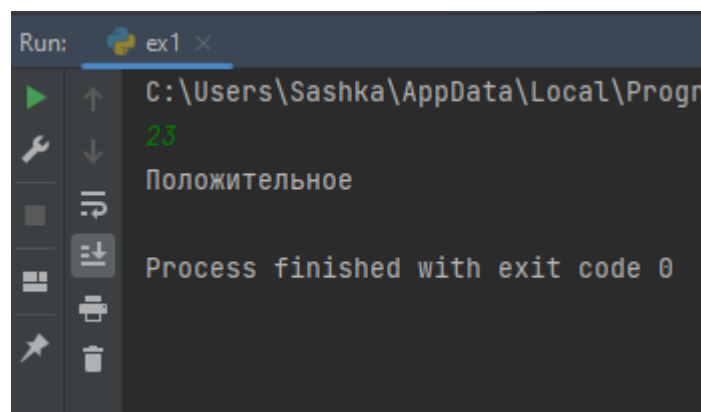
Рисунок 4 – Пример выполнения программы

#### 5. Выполнение задания №1.



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

Рисунок 5 – Код программы



```
Run: ex1 x
C:\Users\Sashka\AppData\Local\Progr
23
Положительное
Process finished with exit code 0
```

Рисунок 6 – Пример выполнения программы

6. Выполнение задания №2.

```
individual.py x primer1.py x ex1.py x ex2.py x ex3.py x ex4.py x
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5      from math import pi
6
7
8      def cylinder(radius, height):
9          command = input('Напишите, что хотите получить: "Площадь боковой '
10                          'поверхности" или "Полная площадь": ')
11          match command:
12              case "Площадь боковой поверхности":
13                  return 2 * pi * radius * height
14              case "Полная площадь":
15                  return 2 * circle(radius) + 2 * pi * radius * height
16
17
18      def circle(radius):
19          return pi * radius ** 2
20
21
22  ▶  if __name__ == "__main__":
23      print(cylinder(float(input("Введите радиус: ")),
24                  float(input("Введите высоту: "))))
25
```

Рисунок 7 – Код программы

```
Run: ex2 x
C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Sashka/Documents/GitHub/Laba11/ex2.py
Введите радиус: 32
Введите высоту: 10
Напишите, что хотите получить: "Площадь боковой поверхности" или "Полная площадь": Площадь боковой поверхности
2010.6192982974676
Process finished with exit code 0
```

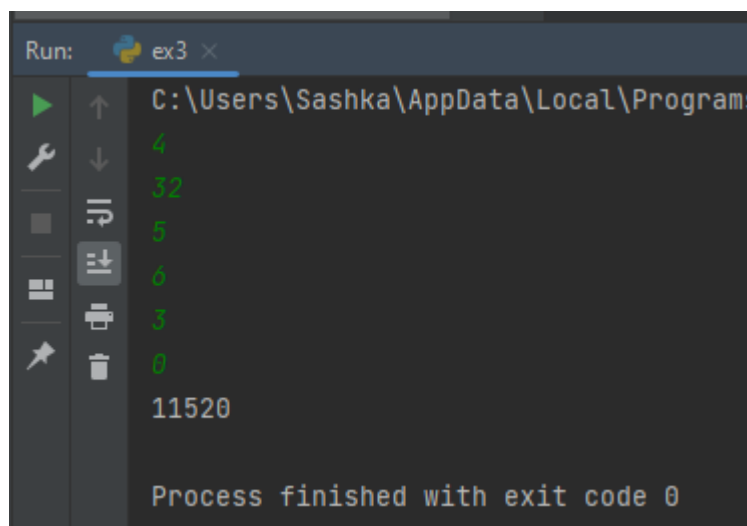
Рисунок 8 – Пример выполнения программы

## 7. Выполнение задания №3



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def multiplication():
5      number = int(input())
6      result = 1
7      while number != 0:
8          result *= number
9          number = int(input())
10     return result
11
12
13 if __name__ == '__main__':
14     print(multiplication())
15
```

Рисунок 9 – Код программы



```
Run: ex3 x
C:\Users\Sashka\AppData\Local\Program
4
32
5
6
3
0
11520

Process finished with exit code 0
```

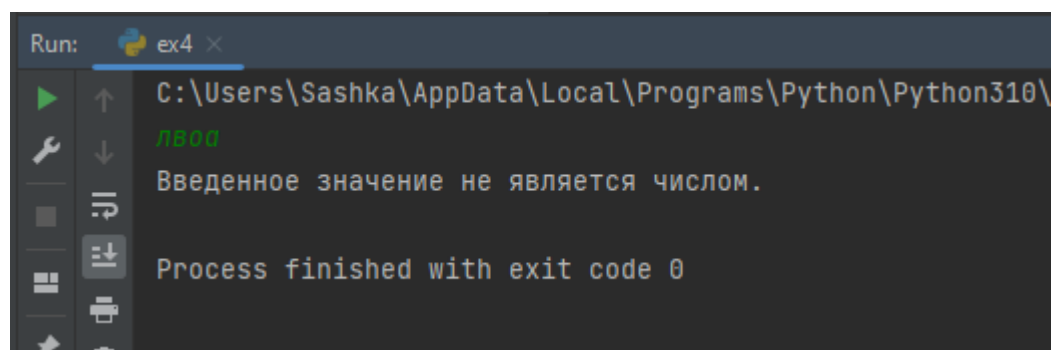
Рисунок 10 – Пример выполнения программы

8. Выполнение задания №4.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def get_input():
6      return input()
7
8
9  def test_input(string):
10     try:
11         if int(string):
12             return True
13     except ValueError:
14         return False
15
16
17  def str_to_int(number):
18     return int(number)
19
20
21  def print_int(number):
22     print(number)
23
24
25  if __name__ == '__main__':
26     result1 = get_input()
27     if test_input(result1):
28         result2 = str_to_int(result1)
29         print_int(result2)
30     else:
31         print('Введенное значение не является числом.')
```

Рисунок 11 – Код программы

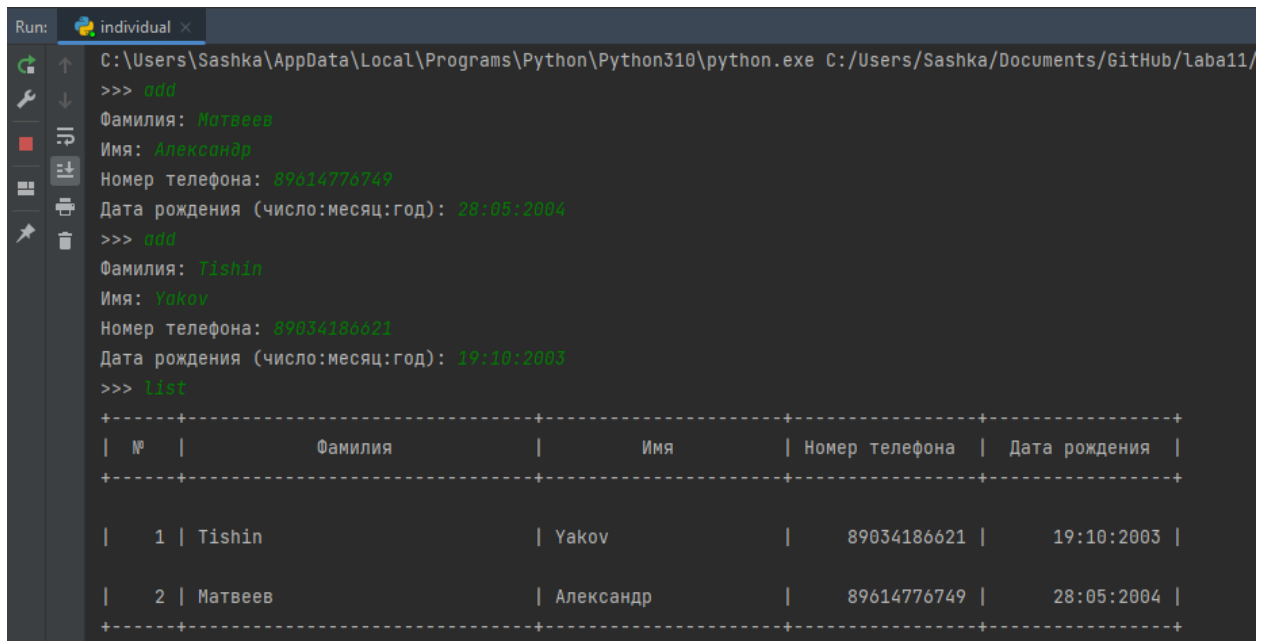


```
Run: ex4 x
C:\Users\Sashka\AppData\Local\Programs\Python\Python310\
лвоо
Введенное значение не является числом.
Process finished with exit code 0
```

Рисунок 12 – Пример выполнения программы



## 9. Выполнение индивидуального задания.



```
Run: individual x
C:\Users\Sashka\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Sashka/Documents/GitHub/lab11/
>>> add
Фамилия: Матвеев
Имя: Александр
Номер телефона: 89614776749
Дата рождения (число:месяц:год): 28:05:2004
>>> add
Фамилия: Tishin
Имя: Yakov
Номер телефона: 89034186621
Дата рождения (число:месяц:год): 19:10:2003
>>> list
+-----+-----+-----+-----+-----+
| № | Фамилия | Имя | Номер телефона | Дата рождения |
+-----+-----+-----+-----+-----+
| 1 | Tishin | Yakov | 89034186621 | 19:10:2003 |
| 2 | Матвеев | Александр | 89614776749 | 28:05:2004 |
+-----+-----+-----+-----+-----+
```

Рисунок 13 – Результат выполнения программы

### Контрольные вопросы:

1. Каково назначение функций в языке программирования Python?

О: Функции в языке Python используются для обособления участка кода, который можно неограниченно вызывать в ходе программы.

2. Каково назначение операторов def и return ?

О: def используется для определения функции, return используется для возврата значения из функции, а так же остановки ее выполнения.

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

О: локальные переменные используются внутри функций и существуют в памяти программы только во время выполнения кода функции, к ним можно обращаться только внутри функции. Глобальные же переменные хранятся в памяти постоянно и к ним можно обратиться из любого места.

4. Как вернуть несколько значений из функции Python?

О: перечислить их через запятую после `return`, тогда в результате выполнения функции вернется кортеж (`tuple`).

5. Какие существуют способы передачи значений в функцию?

О: при помощи позиционируемых аргументов; при помощи ключевых аргументов; при помощи аргументов по умолчанию;

6. Как задать значение аргументов функции по умолчанию?

О: при помощи оператора `(=)`.

7. Каково назначение `lambda`-выражений в языке Python?

О: `lambda`-выражения – это анонимные функции, которые используются чаще всего с функциями `map()` и `filter()`.

8. Как осуществляется документирование кода согласно PEP257?

О: согласно PEP257, строка документации является строковым литералом, который встречается в качестве первого оператора в модуле, функции, классе или определении метода. В Python строки документации должны быть окружены тройными двойными скобками. Существуют: однострочная и многострочная строка документации.

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

О: Однострочные строки документации используются для очевидных случаев. Многострочные строки документации используются для более сложных случаев