

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 2
по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-24-1:

Безруков Даниил Андреевич

«2» ноября 2025г.

Подпись студента _____ Безруков _____

Работа защищена « » _____ 2025г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2025

Тема: Основы языка Python

Цель: исследование процесса установки и базовых возможностей языка Python 3.x

Порядок выполнения работы:

1. Создаём публичный репозиторий
2. Клонировем репозиторий на компьютер
3. Создаём файл .gitignore и вносим изменения для PyCharm

```
.. ----  
# option (not reco  
.idea/  
.venv/
```

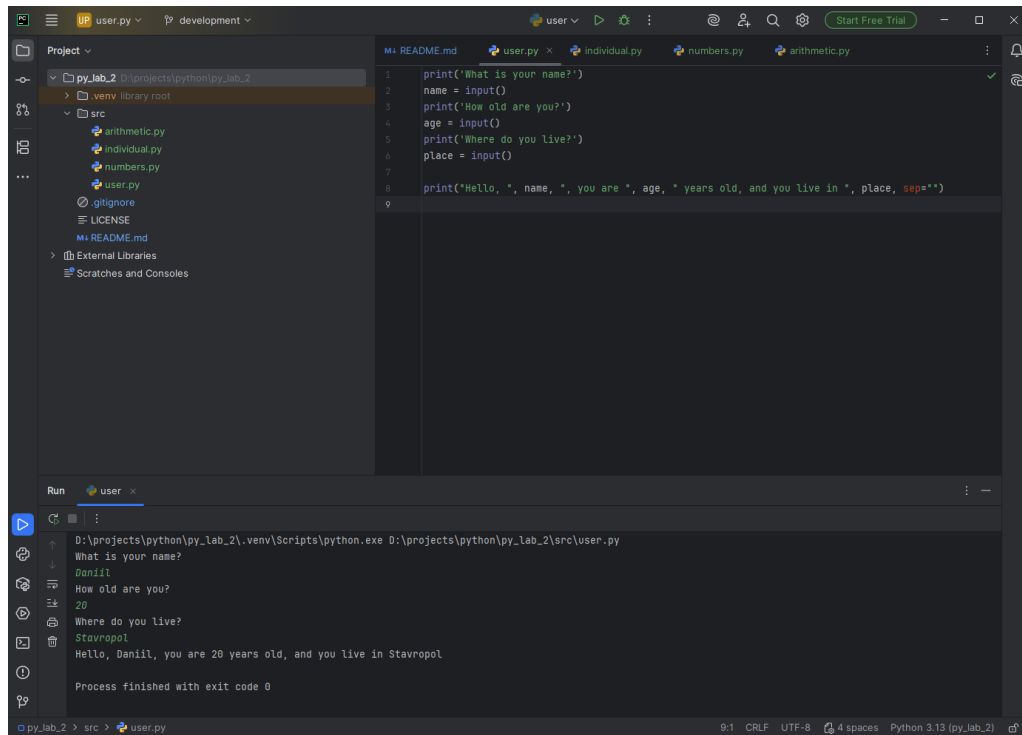
Рисунок 1. Файл .gitignore

4. Создаём ветку для разработки (development)

```
user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)  
$ git branch development  
  
user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)  
$ git switch development  
M       .gitignore  
M       README.md  
A       src/arithmetic.py  
A       src/individual.py  
A       src/numbers.py  
A       src/user.py  
Switched to branch 'development'  
  
user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (development)  
$
```

Рисунок 2. Создание ветки

5. Напишем программу user.py согласно заданию 8



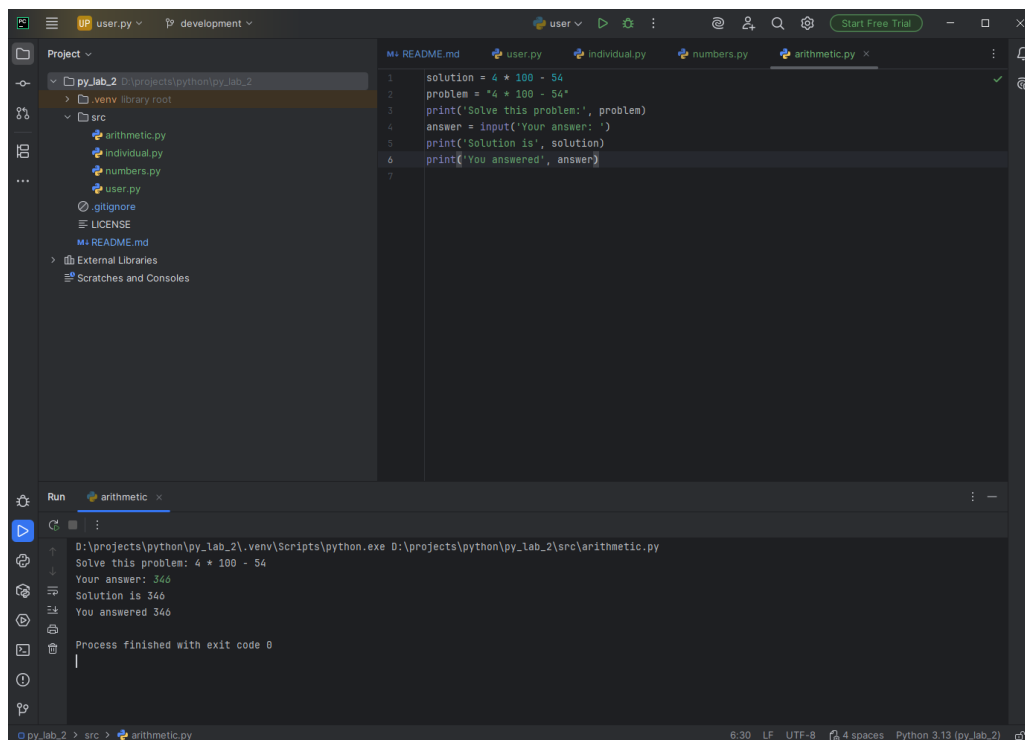
```
1 print('What is your name?')
2 name = input()
3 print('How old are you?')
4 age = input()
5 print('Where do you live?')
6 place = input()
7
8 print('Hello, ', name, ', you are ', age, ' years old, and you live in ', place, sep='')
9
```

Run user

```
D:\projects\python\py_lab_2\venv\Scripts\python.exe D:\projects\python\py_lab_2\src\user.py
What is your name?
Daniil
How old are you?
20
Where do you live?
Stavropol
Hello, Daniil, you are 20 years old, and you live in Stavropol
Process finished with exit code 0
```

Рисунок 3. Выполнение программы

6. Напишем программу arithmetic.py согласно заданию 9



```
1 solution = 4 * 100 - 54
2 problem = "4 * 100 - 54"
3 print('Solve this problem:', problem)
4 answer = input('Your answer: ')
5 print('Solution is', solution)
6 print('You answered', answer)
7
```

Run arithmetic

```
D:\projects\python\py_lab_2\venv\Scripts\python.exe D:\projects\python\py_lab_2\src\arithmetic.py
Solve this problem: 4 * 100 - 54
Your answer: 340
Solution is 340
You answered 340
Process finished with exit code 0
```

Рисунок 4. Результат выполнения программы

7. Напишем программу numbers.py согласно заданию 10

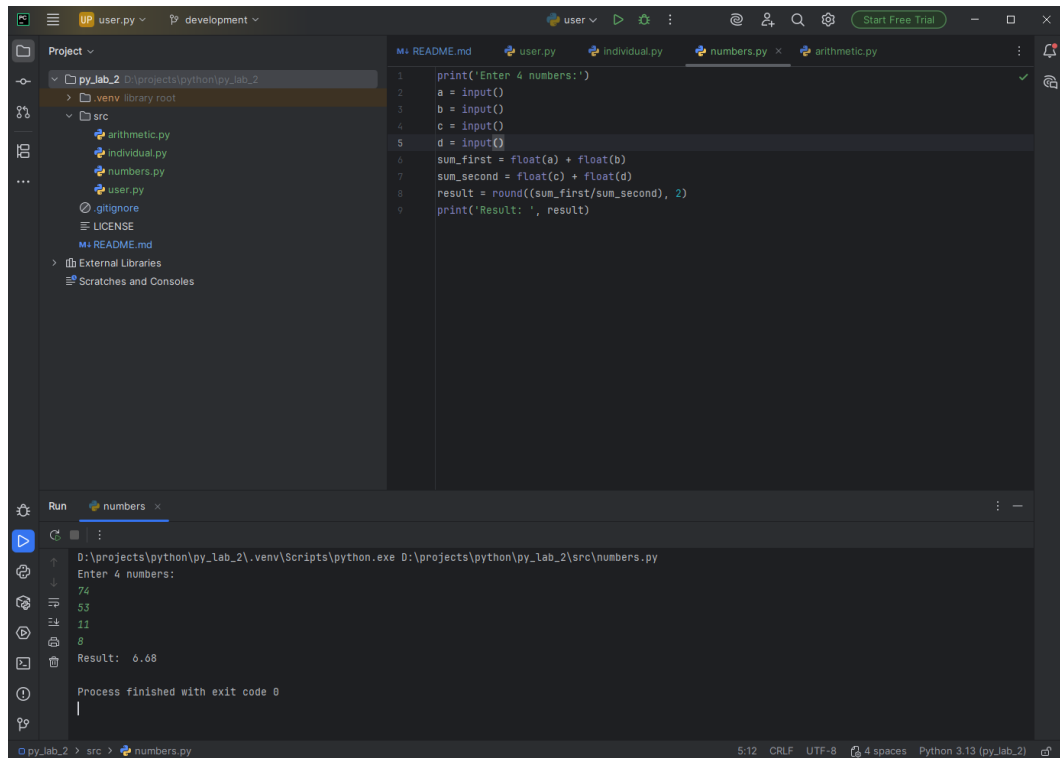


Рисунок 5. Выполнение программы

8. Напишем программу для индивидуального задания (individual.py) согласно варианту 2

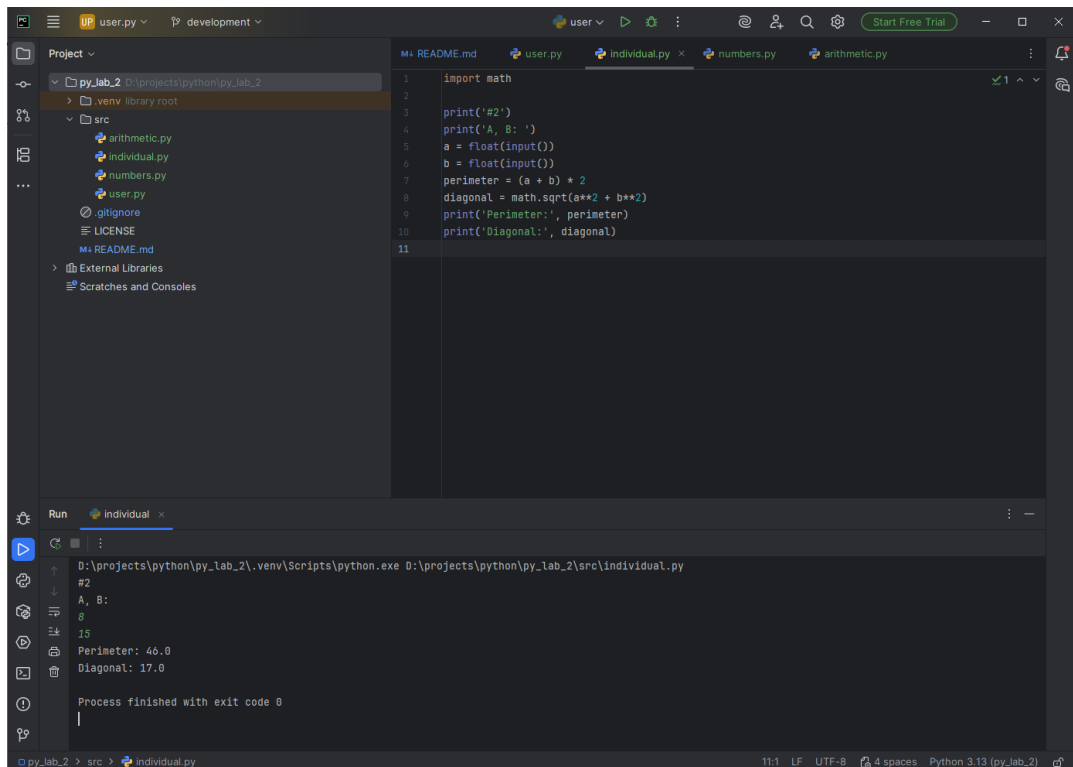
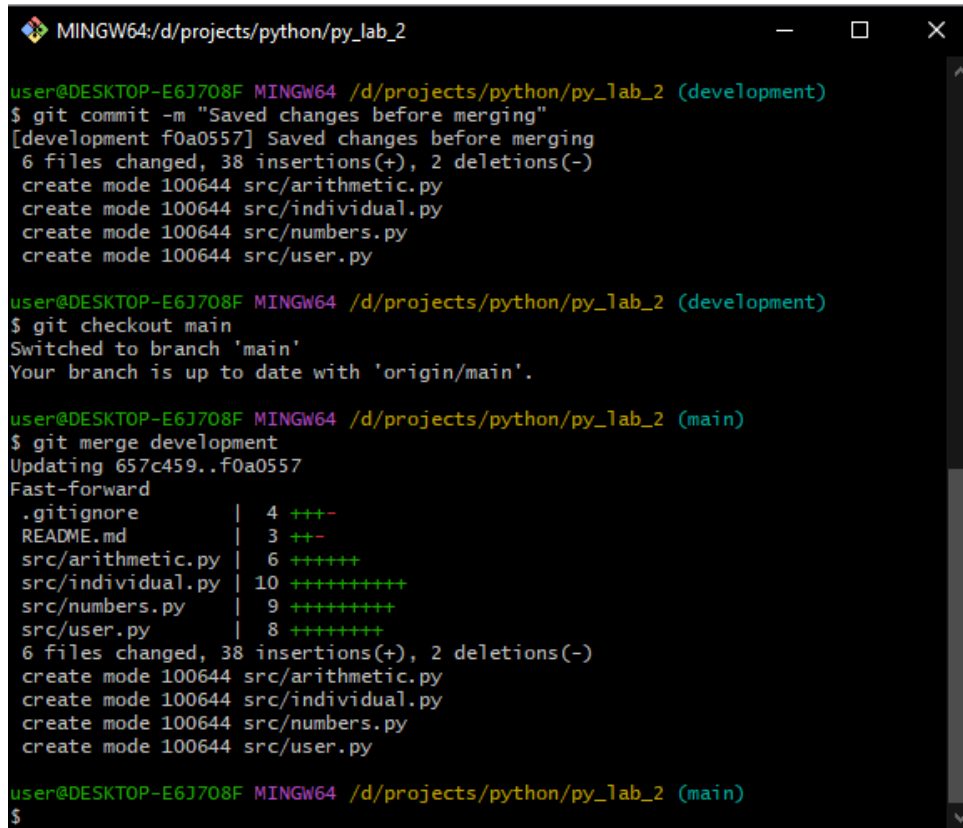


Рисунок 6. Результат выполнения программы

9. Выполняем слияние ветки develop с веткой main



```
MINGW64:/d/projects/python/py_lab_2

user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (development)
$ git commit -m "Saved changes before merging"
[development f0a0557] Saved changes before merging
6 files changed, 38 insertions(+), 2 deletions(-)
create mode 100644 src/arithmetic.py
create mode 100644 src/individual.py
create mode 100644 src/numbers.py
create mode 100644 src/user.py

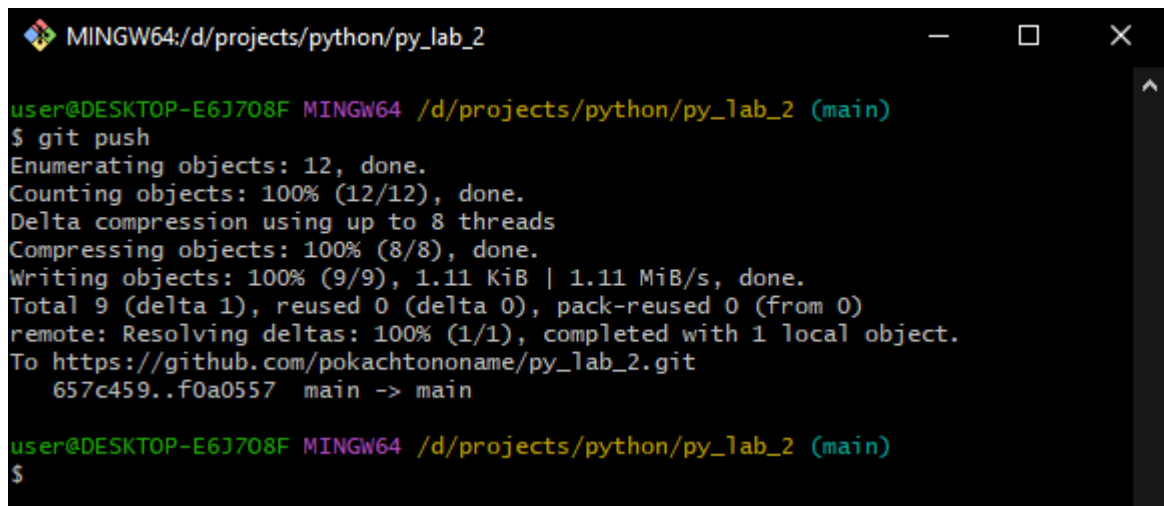
user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (development)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)
$ git merge development
Updating 657c459..f0a0557
Fast-forward
 .gitignore      | 4 +++-
 README.md       | 3 ++-
 src/arithmetic.py | 6 +++++
 src/individual.py | 10 ++++++++
 src/numbers.py   | 9 ++++++++
 src/user.py      | 8 ++++++
 6 files changed, 38 insertions(+), 2 deletions(-)
 create mode 100644 src/arithmetic.py
 create mode 100644 src/individual.py
 create mode 100644 src/numbers.py
 create mode 100644 src/user.py

user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)
$
```

Рисунок 7. Слияние выполнено

10. Отправим изменения на сервер GitHub



```
MINGW64:/d/projects/python/py_lab_2

user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)
$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.11 KiB | 1.11 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/pokachtononame/py_lab_2.git
 657c459..f0a0557  main -> main

user@DESKTOP-E6J708F MINGW64 /d/projects/python/py_lab_2 (main)
$
```

Рисунок 8. Отправка изменений. Адрес репозитория:

https://github.com/pokachtononame/py_lab_2/

Ответы на контрольные вопросы:

1. Опишите основные этапы установки Python в Windows и Linux.

Windows:

- 1) Скачать установщик с официального сайта python.org
- 2) Запустить .exe файл и отметить галочку "Add Python to PATH"
- 3) Выбрать тип установки (рекомендуется обычная)
- 4) Завершить установку и проверить через команду `python --version`

Linux:

- 1) Обновить пакеты: `sudo apt update`
- 2) Установить Python: `sudo apt install python3`
- 3) Проверить установку: `python3 --version`

2. В чем отличие пакета Anaconda от пакета Python, скачиваемого с официального сайта?

Anaconda — это дистрибутив, включающий:

- 1) Интерпретатор Python
- 2) Предустановленные библиотеки для Data Science (NumPy, Pandas, Matplotlib, Jupyter)
- 3) Менеджер пакетов Conda для управления окружениями
- 4) Графический интерфейс Anaconda Navigator

Официальный Python — только интерпретатор и базовые библиотеки.

3. Как осуществить проверку работоспособности пакета Anaconda?

- 1) Открыть командную строку или Anaconda Prompt
- 2) Ввести команды: `bash`, `conda –version`, `python –version`, `jupyter --version`

4. Как задать используемый интерпретатор языка Python в IDE PyCharm?

- 1) Открыть Settings (Ctrl+Alt+S)
- 2) Перейти в раздел: Project -- Python Interpreter
- 3) Нажать на шестерёнку -- Add Interpreter
- 4) Выбрать путь к интерпретатору

5. Как осуществить запуск программы с помощью IDE PyCharm?

- 1) Открыть файл с кодом (.py)
- 2) Нажать сочетание клавиш Ctrl+Shift+F10
- 3) Или правой кнопкой по файлу -- Run

6. В чем суть интерактивного и пакетного режимов работы Python?

- 1) Интерактивный режим — выполнение кода построчно с немедленным результатом (например, в Jupyter Notebook).
- 2) Пакетный режим — запуск готовой программы из файла целиком

7. Почему язык программирования Python называется языком динамической типизации?

Типы переменных определяются автоматически во время выполнения программы, а не при компиляции.

8. Какие существуют основные типы в языке программирования Python?

- 1) int (целые числа)
- 2) float (вещественные числа)
- 3) str (строки)
- 4) bool (логический тип)
- 5) list (списки)
- 6) tuple (кортежи)
- 7) dict (словари)
- 8) set (множества)

9. Как создаются объекты в памяти? Каково их устройство? В чем заключается процесс объявления новых переменных и работа операции присваивания?

При создании переменной:

- 1) В памяти выделяется место для объекта
- 2) Переменная становится ссылкой на этот объект
- 3) Оператор присваивания (=) связывает имя переменной с объектом

10. Как получить список ключевых слов в Python?


```
python
import keyword
print(keyword.kwlist)
```

11. Каково назначение функций id() и type()?

1) id() — возвращает уникальный идентификатор объекта в памяти

2) type() — возвращает тип объекта

12. Что такое изменяемые и неизменяемые типы в Python.

1) Неизменяемые: int, float, str, tuple — нельзя изменить после создания.

2) Изменяемые: list, dict, set — можно изменять содержимое.

13. Чем отличаются операции деления и целочисленного деления?

1) Обычное деление: / — возвращает float ($7 / 2 = 3.5$)

2) Целочисленное деление: // — возвращает int ($7 // 2 = 3$)

14. Какие имеются средства в языке Python для работы с комплексными числами?

1) print(z.real) # действительная часть

2) print(z.imag) # мнимая часть

15. Каково назначение и основные функции библиотеки (модуля) `math`?

`math` — математические функции:

- `math.sqrt()` — квадратный корень
- `math.sin()`, `math.cos()` — тригонометрия
- `math.pi`, `math.e` — константы

`cmath` — аналогичные функции для комплексных чисел.

16. Каково назначение именованных параметров `sep` и `end` в функции `print()`?

- 1) `sep` — разделитель между аргументами (по умолчанию пробел)
- 2) `end` — что выводить в конце (по умолчанию перевод строки)

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python?

`format()` — подстановка значений в строку:

```
python "{} + {} = {}".format(2, 3, 5)
```

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?

```
x = int(input("Введите целое число: "))  
y = float(input("Введите вещественное число: "))
```

Выводы: в результате выполнения работы были освоены основы программирования на Python и базовые навыки работы с Git и IDE PyCharm