

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины «Программирование на Python»

Выполнил:
Щегольков Савва Игоревич
1 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Автоматизированные системы
обработки информации и
управления», очная форма обучения

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Основы языка Python

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

Порядок выполнения работы:

Ссылка на репозиторий: <https://github.com/xouixao/lab2>

Создание репозитория в GitHub и дальнейшая работа с ним

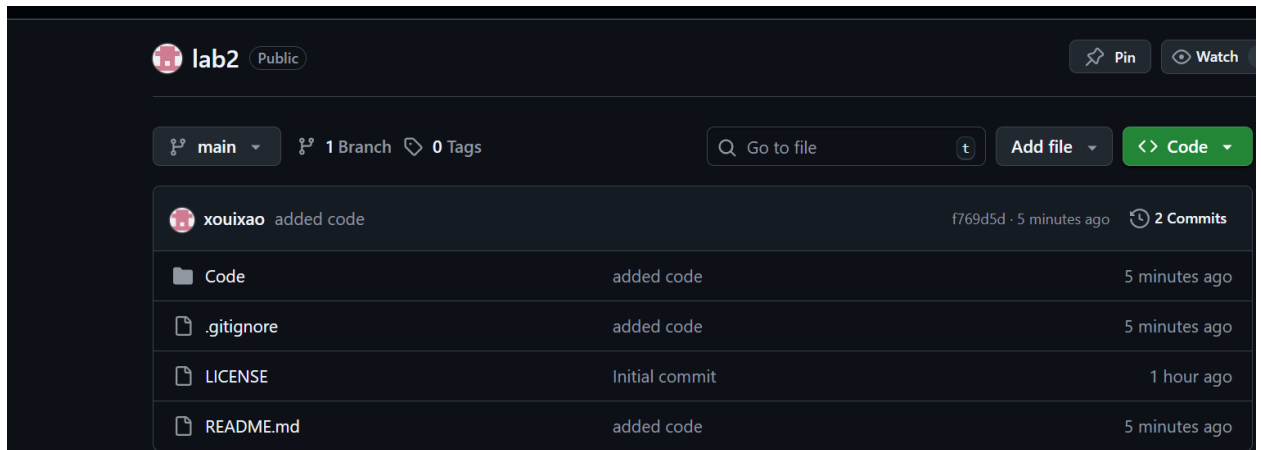


Рисунок 1 – созданный репозиторий.

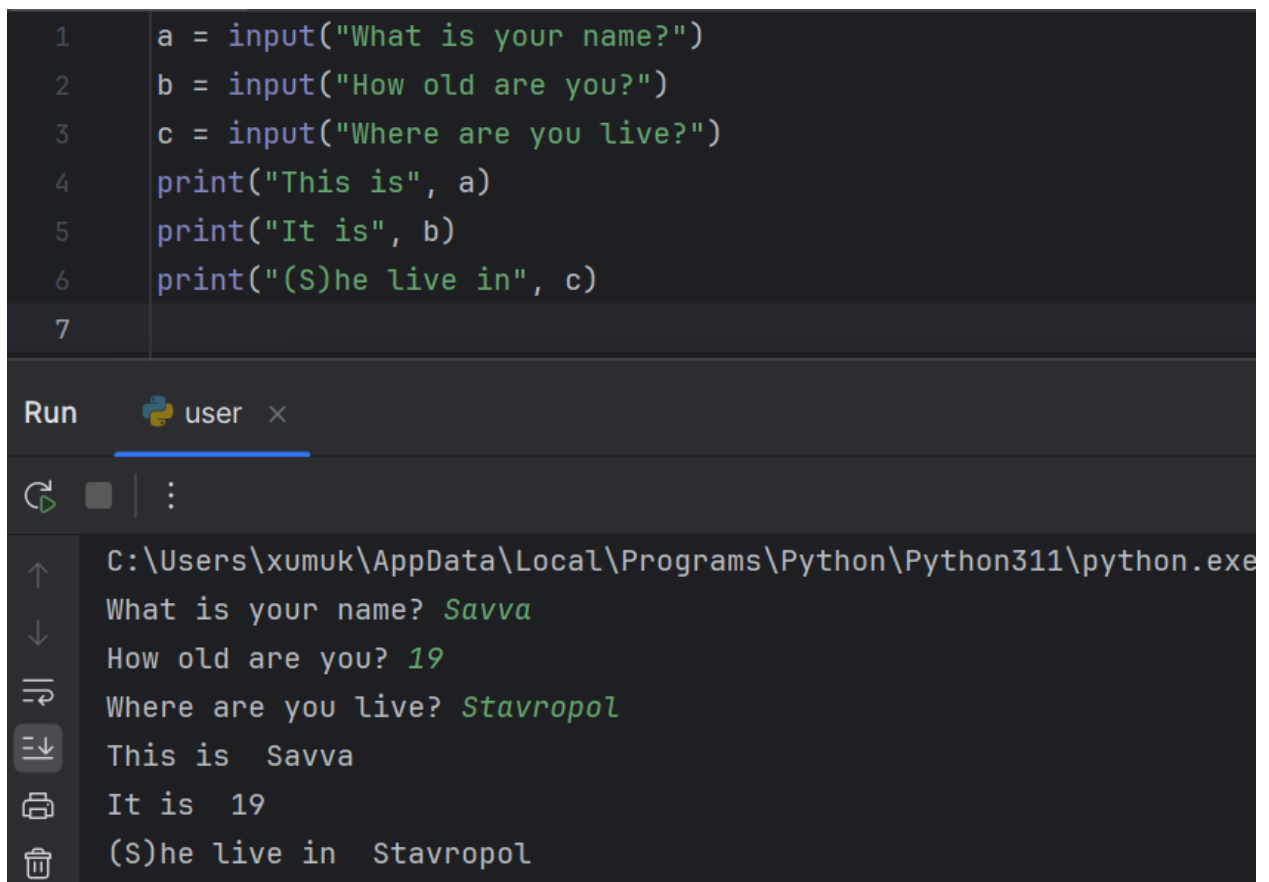


Рисунок 2 – Задание 1.

```
user.py arithmetic.py × numbers.py individual.py hardc
1 a = input("4 * 100 - 54 = ")
2 print("Correct answer = 346")
3 print("Your answer =", a)
4

Run arithmetic ×
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\python.exe C:
4 * 100 - 54 = 2
Correct answer = 346
Your answer = 2
Process finished with exit code 0
```

Рисунок 3 – Задание 2.

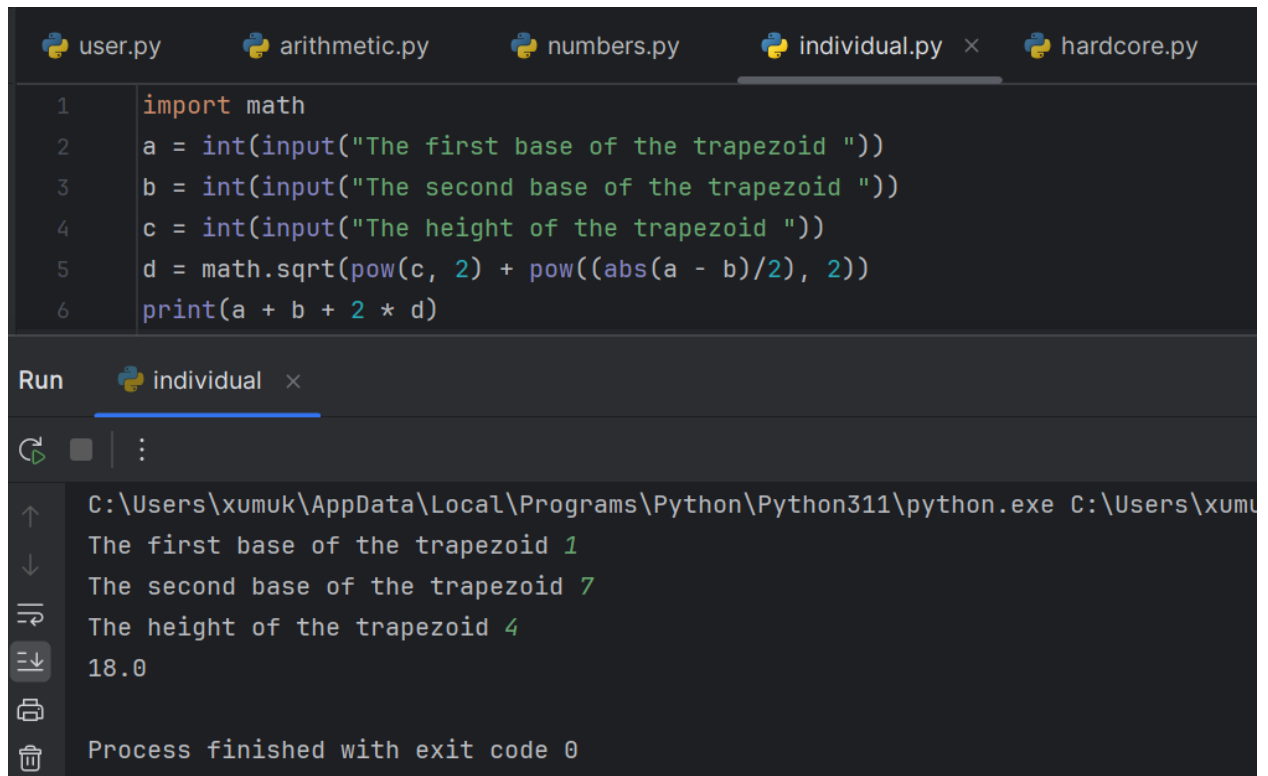
```
user.py arithmetic.py numbers.py × individual.py hardcore.py README.md
1 a = input("Enter first number ")
2 b = input("Enter second number ")
3 c = input("Enter third number ")
4 d = input("Enter forth number ")
5 a = int(a)
6 b = int(b)
7 c = int(c)
8 d = int(d)
9 a += b
10 c *= d
11 a /= c
12 print("%.2f" % (a))

Run numbers ×
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\python.exe C:\Users\xumuk\lab2\Code\numbers.py
Enter first number 2
Enter second number 4
Enter third number 3
Enter forth number 1
1.50
Process finished with exit code 0
```

Рисунок 4 – Задание 3.

Индивидуальное задание заключается в том, чтобы найти периметр равнобедренной трапеции, если известны ее основания и высота, а повышенной сложности в установлении факта делимости одного числа на другое без остатка или наоборот, без использования циклов или условных

операторов, при делимости нужно вывести 1, а при неделимости любое отличное от 1 число.



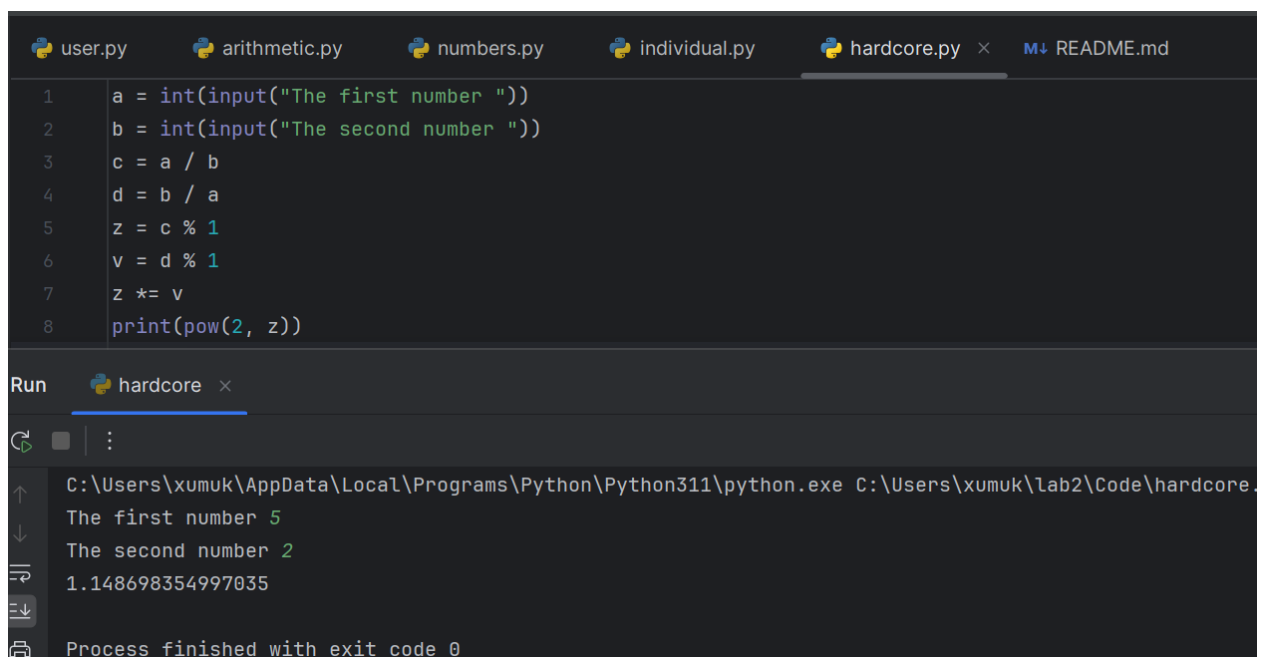
The screenshot shows a Python IDE with several tabs: user.py, arithmetic.py, numbers.py, individual.py (selected), and hardcore.py. The code in individual.py is as follows:

```
1 import math
2 a = int(input("The first base of the trapezoid "))
3 b = int(input("The second base of the trapezoid "))
4 c = int(input("The height of the trapezoid "))
5 d = math.sqrt(pow(c, 2) + pow((abs(a - b)/2), 2))
6 print(a + b + 2 * d)
```

Below the code editor, the 'Run' button is visible, and the output window shows the execution results:

```
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\python.exe C:\Users\xumuk\
The first base of the trapezoid 1
The second base of the trapezoid 7
The height of the trapezoid 4
18.0
Process finished with exit code 0
```

Рисунок 5 – Индивидуальное задание (вариант 24).



The screenshot shows a Python IDE with several tabs: user.py, arithmetic.py, numbers.py, individual.py, hardcore.py (selected), and README.md. The code in hardcore.py is as follows:

```
1 a = int(input("The first number "))
2 b = int(input("The second number "))
3 c = a / b
4 d = b / a
5 z = c % 1
6 v = d % 1
7 z *= v
8 print(pow(2, z))
```

Below the code editor, the 'Run' button is visible, and the output window shows the execution results:

```
C:\Users\xumuk\AppData\Local\Programs\Python\Python311\python.exe C:\Users\xumuk\lab2\Code\hardcore.
The first number 5
The second number 2
1.148698354997035
Process finished with exit code 0
```

Рисунок 6 – Задание повышенной сложности (вариант 8).

Контрольные вопросы:

1. Python — интерпретируемый язык высокого уровня с динамической типизацией, поддерживающий ООП, функциональное и

процедурное программирование; для установки в Windows скачайте с python.org, запустите установщик и добавьте в PATH; в Linux используйте менеджер пакетов (`apt install python3` в Ubuntu) или Anaconda; общие шаги — скачать, установить, проверить версию командой `python --version`.

2. Запуск Anaconda осуществляется через Anaconda Navigator (иконка в меню или поиск), где выбираете среду (например, Jupyter или Spyder), или в терминале командой `conda activate base` для базовой среды; предварительно установите Anaconda с официального сайта.

3. Проверка доступности пакета Anaconda: в терминале введите `conda list` для списка установленных пакетов или `conda search <пакет>` для поиска; если Anaconda установлен, команда `conda --version` покажет версию.

4. Запуск IDE PyCharm: после установки с jetbrains.com откройте через меню (PyCharm Community/Professional) или в терминале `pycharm.sh` (Linux) / `pycharm.exe` (Windows); при первом запуске настройте интерпретатор Python.

5. Запуск программы в PyCharm: откройте проект, создайте/откройте файл `.py`, напишите код, нажмите зеленую стрелку (Run) в toolbar или правой кнопкой на файле > Run; отладка через Debug с точками останова.

6. Суть интегрированной среды (IDE) Python — в объединении редактора, отладчика, терминала и инструментов (например, PyCharm); интерактивная среда (REPL как IDLE или Jupyter) позволяет выполнять код по строкам, тестировать выражения в реальном времени для быстрой разработки и экспериментов.

7. Основные типы в Python: целые числа (`int`), вещественные (`float`), строки (`str`), списки (`list`), кортежи (`tuple`), словари (`dict`), множества (`set`), булевы (`bool`), `NoneType`; они динамические, с автоматическим управлением памятью.

8. Объекты в Python создаются присваиванием (`a = 5` создает `int`-объект), типы — встроенные (`int`, `str`) и пользовательские (`class`); используются

для хранения данных, методов; всё в Python — объект с атрибутами и методами.

9. Объявление переменных в Python: просто присваивание ($x = 10$), без ключевых слов; используются для хранения значений, в выражениях, функциях; глобальные — вне функций, локальные — внутри; множественное: $a, b = 1, 2$.

10. Смена строк в Python: для многострочных строк используйте тройные кавычки ("""строка"""), или `\n` для переноса; в коде — обратный слеш `\` для продолжения на следующей строке.

11. Ключевые функции ввода/вывода в Python: `print()` для вывода, `input()` для ввода от пользователя; `open()` для файлов (`with open('file.txt', 'r') as f:`), `read()/write()` для чтения/записи.

12. Комментарии в Python: однострочные с `#` (`#` это комментарий), многострочные с тройными кавычками ("""коммент"""); размещаются для объяснения кода, игнорируются интерпретатором.

13. Ключевые ключевые слова и функции ввода/вывода в Python? (Вопрос кажется повтором 11, возможно опечатка; если о функциях — см. 11; ключевые слова: `if`, `for`, `def`, `class` и т.д.)

14. Как наносить структура и какие Python даёт возможности? Что анализирует `math`? (Возможно, "Как наносить структуру" — опечатка; структура кода: отступы для блоков; возможности: модули, библиотеки; `math` — модуль для математики: `sin()`, `cos()`, `sqrt()`, `pi`, `exp()` и т.д.)

15. Python? (Вопрос неполный; возможно, "Как используются циклы в Python?": `for i in range(5):` для итераций, `while` условие: для условий; `break/continue` для контроля.)

16. (Вопрос обрезан; если "Как оформлять Python?", то следуйте PEP 8: отступы 4 пробела, `snake_case` для переменных, `docstrings` для документации.)

Вывод: в ходе выполнения лабораторной работы были изучены установка и базовые возможности Python.