

**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:  
Оразов Тимур  
2 курс, группа ИВТ-б-о-24-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А.

---

(подпись)

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

Ставрополь, 2025 г.

## Вариант 16

Тема: Основы языка Python

Цель: исследование процесса установки и безопасных возможностей Python.

Ссылка на репозиторий с которого выполнялась работа:

[https://github.com/Tim120903/laba\\_2](https://github.com/Tim120903/laba_2)

Ход работы:

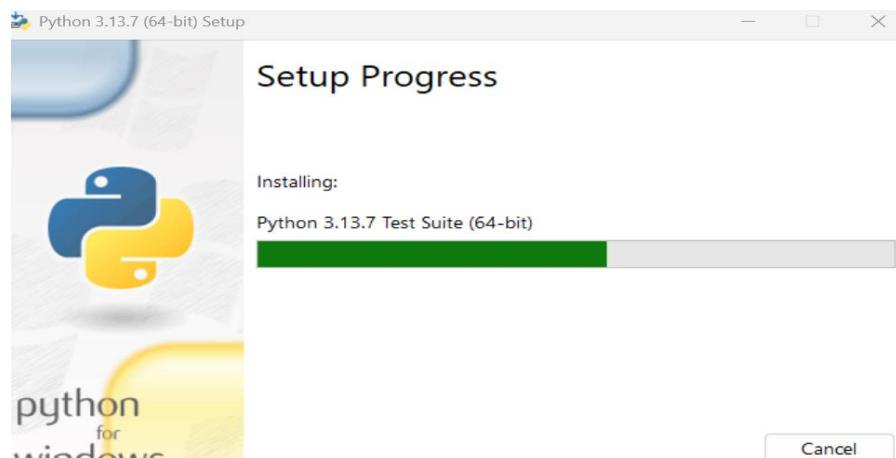
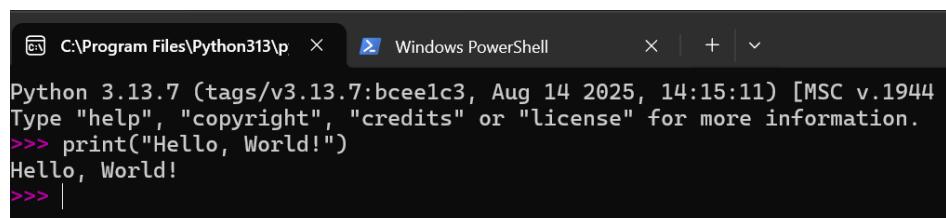


Рис. 1 - установка python.



```
C:\Program Files\Python313\p... >>> print("Hello, World!")
Hello, World!
>>> |
```

Рис. 2 - проверка работоспособности терминала Python

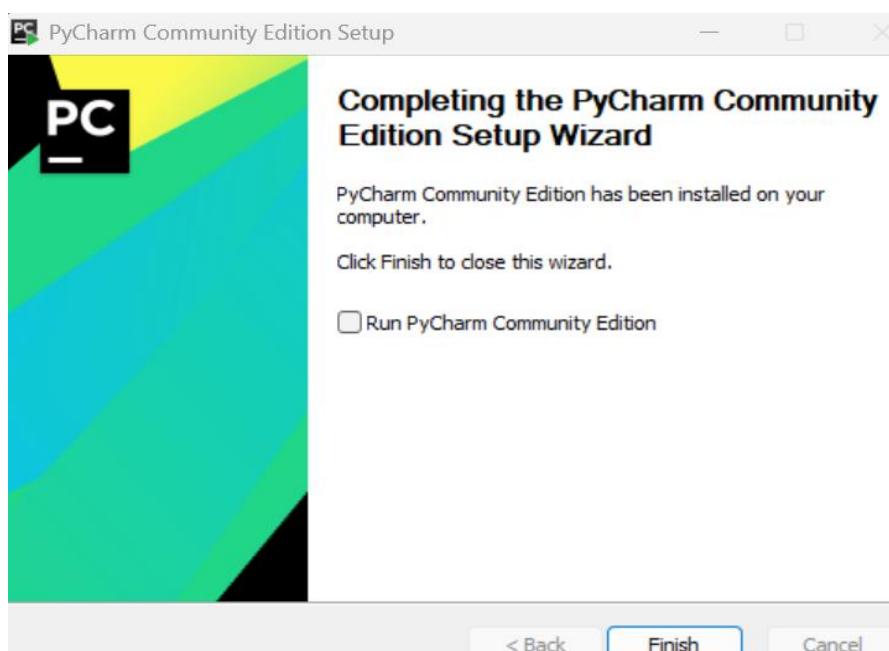


Рис. 3 – установка PyCharm Community.

```
Project C:\Users\user\PycharmProject Lab_2.py
Filetered C:\Users\user\PycharmProject Lab_2.py
Internal Libraries
Lab_2
print("Hello, World")
print(5+5)

"C:\Program Files\Python313\python.exe" C:/Users/user/PycharmProjects/untitled/Lab_2.py
Hello, World
10
```

Рис. 4 - проверка на работоспособность PyCharm.

### Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

1 General

Owner \* Repository name \*

Tim120903 / laba  
✓ laba is available.

Great repository names are short and memorable. How about [reimagined-octo-palm-tree](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility \* Public

Choose who can see and commit to this repository

Add README On

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

Add license

Licenses explain how others can use your code. [About licenses](#)

MIT License

Create repository

Рис. 5- создание общедоступного репозитория.

```
C:\Users\Тимур>git clone https://github.com/Tim120903/laba_2.git
```

Рис. 6 – копирование репозитория на устройство.

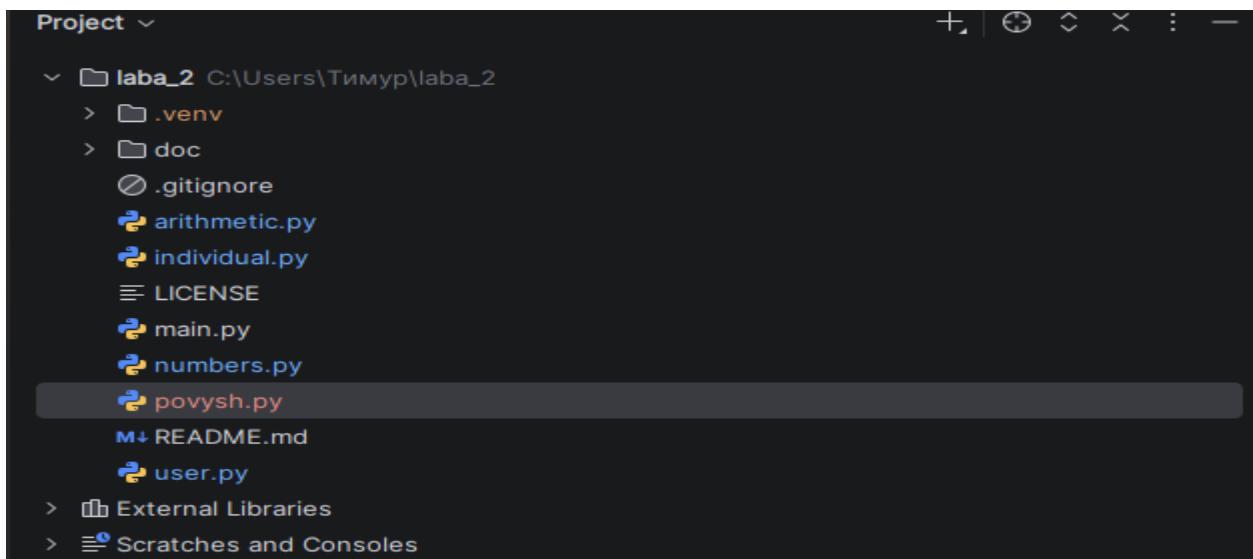


Рис. 7 – создание файла user в PyCharm в выбранном репозитории.

8. Напишите программу (файл *user.py*), которая запрашивала бы у пользователя:

- его имя (например, "What is your name?")
- возраст ("How old are you?")
- место жительства ("Where are you live?")

После этого выводила бы три строки:

```
"This is `имя`"  
"It is `возраст`"  
"(S)he live in `место_жительства`"
```

Вместо **имя**, **возраст**, **место\_жительства** должны быть данные, введенные пользователем. Примечание: можно писать фразы на русском языке, но если вы планируете стать профессиональным программистом, привыкайте к английскому.

Рис. 8 – Задание 1.

The screenshot shows the PyCharm interface. In the code editor, the 'user.py' tab is active, displaying the following Python code:

```
1 name = str(input("What is your name? "))  
2 age = int(input("How old are you? "))  
3 location = str(input("Where are you live? "))  
4 print("This is", name)  
5 print("It is", age)  
6 print("(S)he live in", location)
```

In the terminal window below, the code is run under the 'user' configuration. The user is prompted for their name, age, and location, and the program outputs three lines of text based on the input.

```
C:\Users\Тимур\laba_2\.venv\Scripts\python.exe C:\Users\Тимур\laba_2\user.py  
What is your name? Tim  
How old are you? 22  
Where are you live? Stavropol  
This is Tim  
It is 22  
(S)he live in Stavropol  
Process finished with exit code 0
```

Рис. 9 - решение к заданию 1.

Add files via upload

Tim120903 authored 2 hours ago

Verified

927d9fb



Рис. 10 - добавление коммита.

9. Напишите программу (файл *arithmetic.py*), которая предлагала бы пользователю решить пример  $4 * 100 - 54$ . Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

Рис. 11 – Задание 2.

```
1 q_u = input("Solve the example 4 * 100 - 54 and enter the answer: ")
2 u_a = int(q_u)
3 r_a = 4 * 100 - 54
4 print("Right answer:", r_a)
5 print("User answer:", u_a)

Run arithmetic <input>
C:\Users\Тимур\лаба_2\.venv\Scripts\python.exe C:\Users\Тимур\лаба_2\arithmetic.py
Solve the example 4 * 100 - 54 and enter the answer: 23
Right answer: 346
User answer: 23

Process finished with exit code 0
```

Рис. 12 - решение к заданию 2.

Решение примера

Tim120903 authored 1 hour ago

Verified

58aeela



Рис. 13 - добавление коммита.

10. Запросите у пользователя четыре числа (файл *numbers.py*). Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.

Рис. 14 – Задание 3.

```
1 a = float(input("enter the first number: "))
2 b = float(input("enter the second number: "))
3 c = float(input("enter the third number: "))
4 d = float(input("enter the fourth number: "))
5 sum_1 = a + b
6 sum_2 = c + d
7 if sum_2 != 0:
8     res = sum_1 / sum_2
9     print(f"The final result: {res:.2f}")
10 else:
11     print("Error!")
```

Run numbers

```
C:\Users\Тимур\laba_2\.venv\Scripts\python.exe C:\Users\Тимур\laba_2\numbers.py
enter the first number: 12.155185
enter the second number: 14.874748451
enter the third number: 12
enter the fourth number: 1.5878945
The final result: 1.99

Process finished with exit code 0
```

Рис. 15 - результат выполнения задания 3.

16. Известно значение температуры по шкале Цельсия. Найти соответствующее значение температуры по шкале:

- Фаренгейта;
- Кельвина.

Для пересчета по шкале Фаренгейта необходимо исходное значение температуры умножить на 1,8 и к результату прибавить 32, а по шкале Кельвина абсолютное значение нуля соответствует -273,15 градуса по шкале Цельсия.

Рис. 16 – индивидуальное задания.

```
1 cels = float(input("Please enter any temperature in celsius: "))
2 Farenheit = cels * 1.8 + 32
3 Kelvin = cels + 273.15
4 print("The temperature in Farenheit is:", Farenheit)
5 print("The temperature in Kelvin is:", Kelvin)
6
```

Run individual

```
C:\Users\Тимур\laba_2\.venv\Scripts\python.exe C:\Users\Тимур\laba_2\individual.py
Please enter any temperature in celsius: 18
The temperature in Farenheit is: 64.4
The temperature in Kelvin is: 291.15
```

Рис. 17- результат выполнения индивидуального задания.

8. Даны два целых числа  $a$  и  $b$ . Если  $a$  делится на  $b$  или  $b$  делится на  $a$ , то вывести 1, иначе – любое другое число. Условные операторы и операторы цикла не использовать.

Рис. 18 - задание повышенной сложности (В-16)

```
1 a = int(input("Enter a number: "))
2 b = int(input("Enter another number: "))
3 result = int((a % b == 0) | (b % a == 0))
4 print(result)
```

Run povysh

```
C:\Users\Тимур\laba_2\.venv\Scripts\python.exe C:\Users\Тимур\laba_2\povysh.py
Enter a number: 12
Enter another number: 4
1
```

Рис. 19 - результат выполнения задания повышенной сложности.

---

main1 Branch0 Tags

Go to filetAdd fileCode

---

 Tim120903	Задание повышенной сложности	97a6bb3 · 26 minutes ago	7 Commits
 .gitignore	Update .gitignore	2 hours ago	
 LICENSE	Initial commit	2 hours ago	
 README.md	Initial commit	2 hours ago	
 arithmetic.py	Решение примера	1 hour ago	
 individual.py	Индивидуальное задание	45 minutes ago	
 numbers.py	запрос чисел у пользователя	1 hour ago	
 povysh.py	Задание повышенной сложности	26 minutes ago	
 user.py	Add files via upload	2 hours ago	

---

Рис. 20 - добавленные файлы с коммитами на Github.

Ответы на контрольные вопросы:

**1. Для установки Python в Windows и Linux нужно:**

В Windows: скачать установщик с официального сайта python.org, запустить его и поставить галочку "Add Python to PATH"

В Linux: использовать команду sudo apt install python3 в терминале или установить из репозитория

**2. Пакет Anaconda отличается от стандартного Python тем, что:**

Anaconda включает в себя сам Python плюс более 1500 популярных научных библиотек (таких как numpy, pandas, matplotlib), а также менеджер пакетов conda для удобного управления зависимостями

**3. Проверку работоспособности пакета Anaconda осуществляют:**

Запуском команд conda --version и python --version в командной строке - они должны показать установленные версии

**4. Задать используемый интерпретатор Python в PyCharm нужно:**

Через меню File → Settings → Project → Python Interpreter, где выбирается нужная версия Python из установленных в системе

**5. Запуск программы в PyCharm осуществляют:**

Нажатием правой кнопки мыши на файле с кодом и выбором "Run" или использованием сочетания клавиш Shift+F10

**6. Суть интерактивного и пакетного режимов работы Python:**

Интерактивный режим: выполнение команд по одной в реальном времени, как в калькуляторе

Пакетный режим: выполнение всей программы из файла целиком

**7. Python называется языком динамической типизации потому, что:**

Типы переменных определяются во время выполнения программы, а не при компиляции, и одна переменная может менять тип в процессе работы

**8. Основные типы в Python включают:**

Целые числа (int), вещественные числа (float), строки (str), списки (list), кортежи (tuple), словари (dict), множества (set) и логический тип (bool)

**9. Объекты в памяти создаются и работают так:**

При создании переменной x = 5 в памяти создается объект 5, а переменная x становится ссылкой на него. Операция присваивания создает новую ссылку на объект

**10. Получить список ключевых слов в Python можно:**

С помощью команды import keyword; print(keyword.kwlist)

**11. Назначение функций id() и type():**

id() возвращает уникальный идентификатор объекта в памяти

type() показывает тип объекта

**12. Изменяемые и неизменяемые типы в Python:**

Неизменяемые: числа, строки, кортежи (созданные объекты нельзя изменить)

Изменяемые: списки, словари, множества (объекты можно изменять после создания)

### **13. Операции деления отличаются так:**

Обычное деление / возвращает вещественное число ( $7/2 = 3.5$ )

Целочисленное деление // возвращает целую часть ( $7//2 = 3$ )

### **14. Средства для работы с комплексными числами:**

Используется тип complex, например:  $z = 3 + 4j$ , с доступом к действительной и мнимой частям через  $z.real$  и  $z.imag$

### **15. Назначение библиотеки math:**

Предоставляет математические функции (тригонометрические, логарифмы, корни) для вещественных чисел. Модуль cmath предоставляет аналогичные функции для комплексных чисел

### **16. Назначение параметров sep и end в функции print():**

sep определяет разделитель между аргументами (по умолчанию пробел)

end определяет, что печатать в конце строки (по умолчанию перевод строки)

### **17. Назначение метода format() и средства форматирования:**

Метод format() позволяет вставлять значения в строку по шаблону.

```
name = "Анна"
```

```
age = 25
```

```
print("Меня зовут {} и мне {} лет".format(name, age))
```

Также существуют f-строки `f"{{переменная}}"` - более современный и удобный способ

### **18. Ввод значений с консоли осуществляют:**

Для целых чисел:  $x = \text{int}(\text{input}())$ , для вещественных:  $y = \text{float}(\text{input}())$

Вывод: в ходе работы были исследованы процессы установки и безопасных возможностей Python.