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

_____
(подпись)

Проверил:
Доцент кафедры инфокомунникаций
Воронкин Р.А.

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема работы: “Основы языка Python ”

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

Порядок выполнения работы:

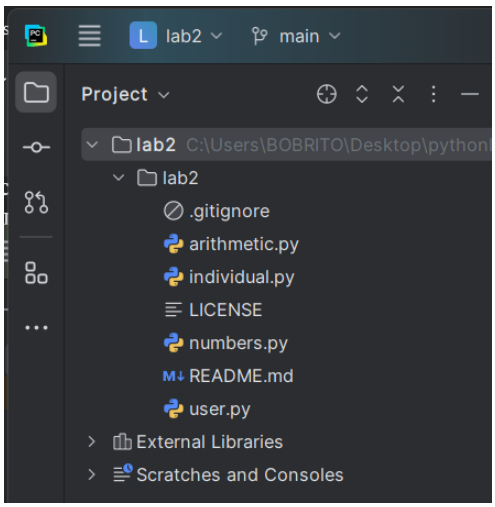

Рисунок 1. Скриншот окна проекта PyCharm

```
C:\Users\BOBRITO\Desktop\pythonlab\lab2\lab2>git checkout develop
M       .gitignore
Already on 'develop'
```

Рисунок 2. Создал ветку develop

```
# user.py

name = input("What is your name? ")
age = input("How old are you? ")
place = input("Where are you live? ")

print(f"This is {name}")
print(f"It is {age}")
print(f"(s)he live in {place}")
```

```
user
C:\Users\BOBRITO\AppData\Local\Programs\Python\Python312\python.exe C:\Users\BOBRITO\Desktop\pythonlab\lab2\lab2\user.py
What is your name? Rostislav
How old are you? 20
Where are you live? Nadezhda
This is Rostislav
It is 20
(s)he live in Nadezhda

Process finished with exit code 0
```

Рисунок 3. Результат работы программы user

```python
correct_answer = 4 * 100 - 54

print("Solve the example: 4 * 100 - 54")
user_answer = input("Your answer: ")

print(f"Correct answer: {correct_answer}")
print(f"Your answer: {user_answer}")
```

```
arithmetic ×
C:\Users\BOBRITO\AppData\Local\Programs\Python\Python312\python.exe C:\Users\BOBRITO\Desktop\pythonlab\lab2\lab2\arithmetic.py
Solve the example: 4 * 100 - 54
Your answer: 346
Correct answer: 346
Your answer: 346

Process finished with exit code 0
```

Рисунок 4. Результат работы программы arithmetic

```python
print("Enter 4 numbers:")

n1 = float(input("Number 1: "))
n2 = float(input("Number 2: "))
n3 = float(input("Number 3: "))
n4 = float(input("Number 4: "))

sum1 = n1 + n2
sum2 = n3 + n4

if sum2 != 0:
    result = sum1 / sum2

    print(f"Result: {result:.2f}")
else:
    print("Division by zero is not allowed.")
```

```
n numbers ×
C:\Users\BOBRITO\AppData\Local\Programs\Python\Python312\python.exe C:\Users\BOBRITO\Desktop\pythonlab\lab2\lab2\numbers.py
Enter 4 numbers:
Number 1: 10
Number 2: 7
Number 3: 2
Number 4: 3
Result: 3.40

Process finished with exit code 0
```

Рисунок 5. Результат работы программы numbers

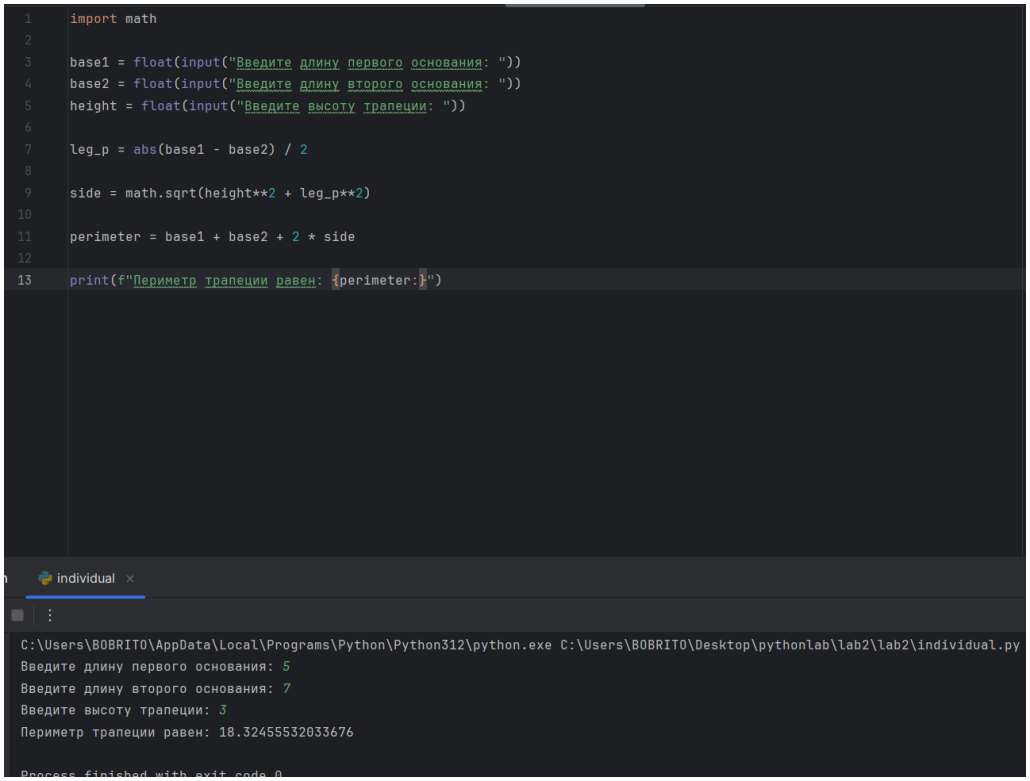

Рисунок 6. Результат работы программы individual по заданию 7

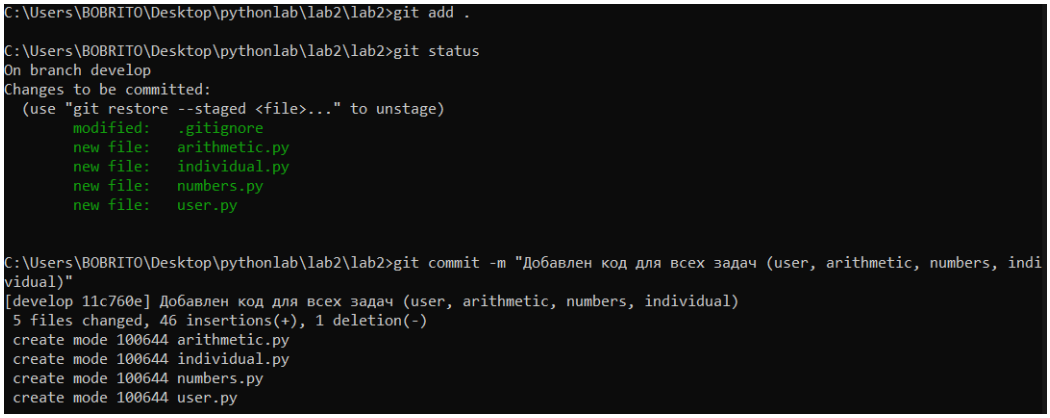

Рисунок 7. Отправил финальный результат на репозиторий

Вывод: В ходе работы были освоены основные операторы и синтаксис Python при решении практических задач, а также исследованы базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

Контрольные вопросы:

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

   Windows: Скачать установщик с python.org. Запустить. Важно: поставить галочку "Add Python to PATH" (чтобы запускать из консоли). Нажать "Install Now".

   Linux: Обычно Python уже установлен. Если нет - в терминале: sudo apt update и sudo apt install python3.

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

   Официальный Python: "Чистый" язык. Библиотеки нужно ставить вручную через pip.

   Anaconda: Дистрибутив для науки и анализа данных. Включает Python + сотни предустановленных библиотек (NumPy, Pandas и др.) + свой менеджер пакетов conda.

3. Как осуществить проверку работоспособности пакета Anaconda?

Открыть командную строку (или Anaconda Prompt) и ввести conda --version или python. Если Anaconda работает, при запуске python будет написано Anaconda, Inc. в приветственном сообщении.

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

File -> Settings (или Preferences на Mac) -> Project: <имя> -> Python Interpreter. Там в выпадающем списке выбрать нужный или нажать шестеренку -> Add, чтобы указать путь к python.exe.

5. Как осуществить запуск программы с помощью IDE PyCharm?

   Нажать зеленую стрелочку (Play) справа сверху.

   Нажать правой кнопкой мыши в коде -> Run 'имя_файла'.

   Горячие клавиши: Shift + F10.

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

Интерактивный: Вводите команду - сразу получаете ответ (консоль Python). Удобно для тестов.

Пакетный: Пишете весь код в файл .py, затем запускаете файл целиком. Используется для написания программ.

7. Почему Python называется языком динамической типизации?

Потому что тип переменной определяется автоматически в момент присваивания значения, и эту переменную можно позже перезаписать значением другого типа (например, сначала x = 5, потом x = "Hello"). Объявлять типы заранее (int x) не нужно.

8. Какие существуют основные типы в языке Python?

Числа: int (целые), float (дробные), complex (комплексные).

Строки: str.

Булево: bool (True/False).

Коллекции: list (список), tuple (кортеж), dict (словарь), set (множество).

NoneType (пустое значение None).

9. Как создаются объекты в памяти? Устройство, объявление и присваивание.

В Python всё является объектом.

Когда вы пишете a = 10, в памяти создается объект числа 10.

Переменная a - это просто ссылка (ярлык), указывающая на этот объект.

Присваивание (=) связывает имя переменной с объектом.

10. Как получить список ключевых слов в Python?

В консоли выполнить:

Python

import keyword

print(keyword.kwlist)

Или набрать help("keywords").

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

id(obj): Возвращает уникальный идентификатор объекта (фактически адрес в памяти).

type(obj): Возвращает тип данных объекта (например, <class 'int'>).

12. Что такое изменяемые и неизменяемые типы в Python?

Неизменяемые (Immutable): Нельзя изменить содержимое после создания (если попытаться, создастся новый объект). Примеры: int, float, str, tuple, bool.

Изменяемые (Mutable): Можно менять содержимое без создания нового объекта. Примеры: list (списки), dict (словари), set (множества).

13. Чем отличаются операции деления и целочисленного деления?

/ (обычное деление): Всегда возвращает float. Пример: 5 / 2 = 2.5.

// (целочисленное): Отбрасывает дробную часть. Пример: 5 // 2 = 2.

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

Встроенный тип complex. Создается через j: z = 3 + 4j. Поддерживает сложение, умножение и т.д.

15. Назначение и функции библиотеки math? Отличие от cmath.

math: Математические функции для вещественных (обычных) чисел (sin, cos, sqrt, pi). Если попробовать math.sqrt(-1), будет ошибка.

cmath: Функции для комплексных чисел. cmath.sqrt(-1) вернет 1j.

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

sep: Разделитель между аргументами. По умолчанию пробел. print(1, 2, sep="-") -> 1-2.

end: Что вывести в конце строки. По умолчанию перенос строки \n. Если сделать end="", следующая печать будет в той же строке.

17. Назначение метода format()? Средства форматирования строк.

format(): Подставляет значения в строку вместо {}. Пример: "Hi, {}".format(name).

f-строки (f-strings): Более современный и быстрый способ (появился в Python 3.6). Пишется буква f перед кавычками. Пример: f"Hi, {name}".

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

Функция input() всегда возвращает строку. Её нужно преобразовать:

Целое: x = int(input("Введите число: "))

Вещественное: y = float(input("Введите число: "))