

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2

Основы языка Python

Выполнил студент группы

ИТС-б-з-22-1

Рябов З.А. « » _____ 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Ставрополь, 2023 г.

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

Ход работы:

1. Создание общедоступного репозитория на GitHub (<https://github.com/DemoGood/zadanie2>).
2. Изучил теоретический материал.
3. Установил PyCharm.

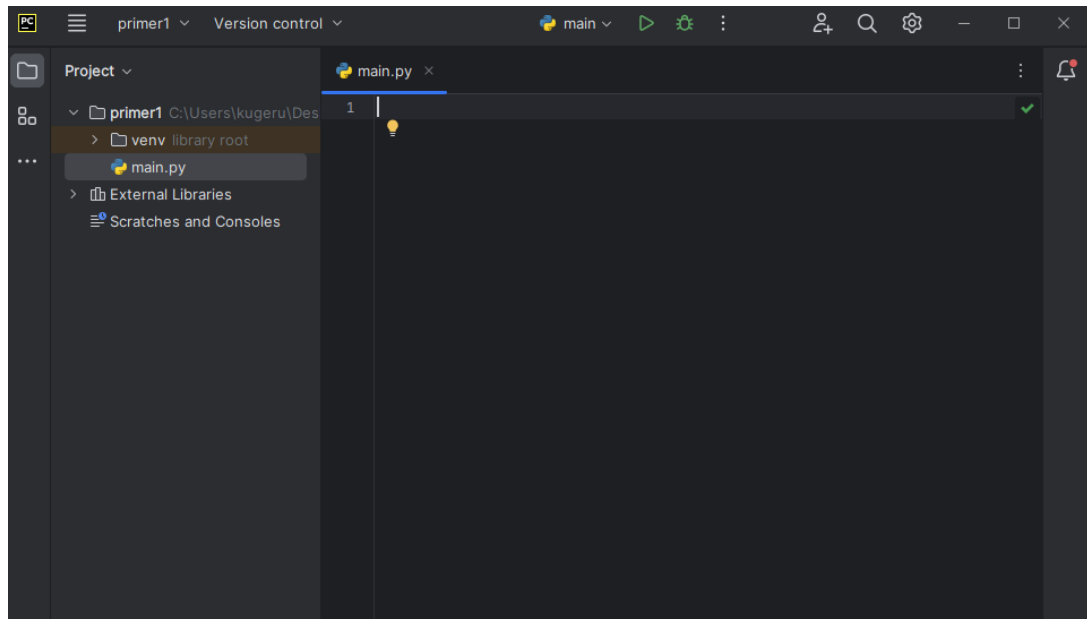
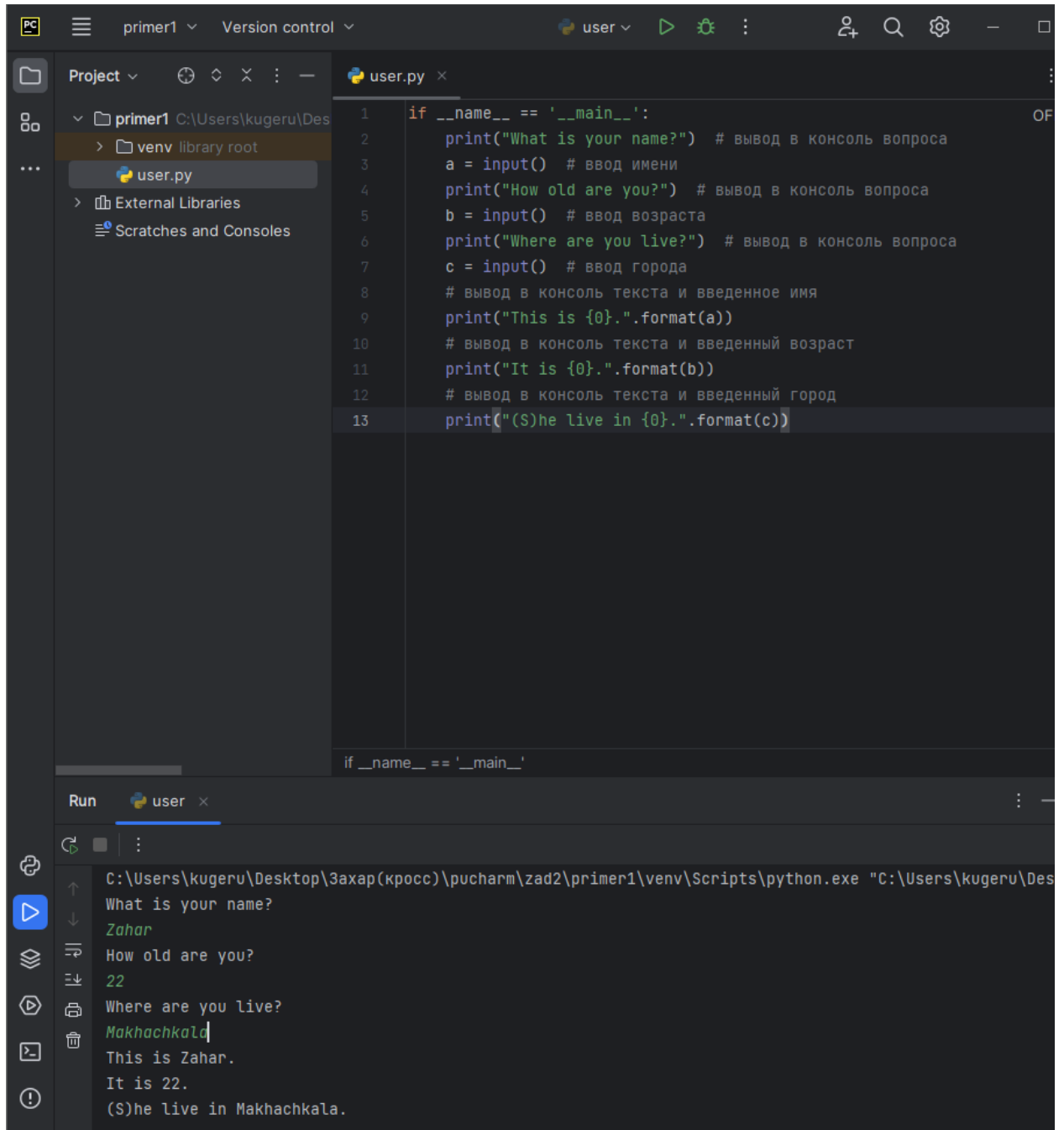


Рисунок 1 – Установка PyCharm

Задача 1. Напишите программу (файл user.py), которая запрашивала бы у пользователя: его имя, возраст, место жительства. После этого вывести три строки: «This is 'имя'», «It is 'возраст'», «(S)he live in 'город'».



The screenshot shows an IDE with a project named 'primer1'. The file 'user.py' is open in the editor. The code in 'user.py' is as follows:

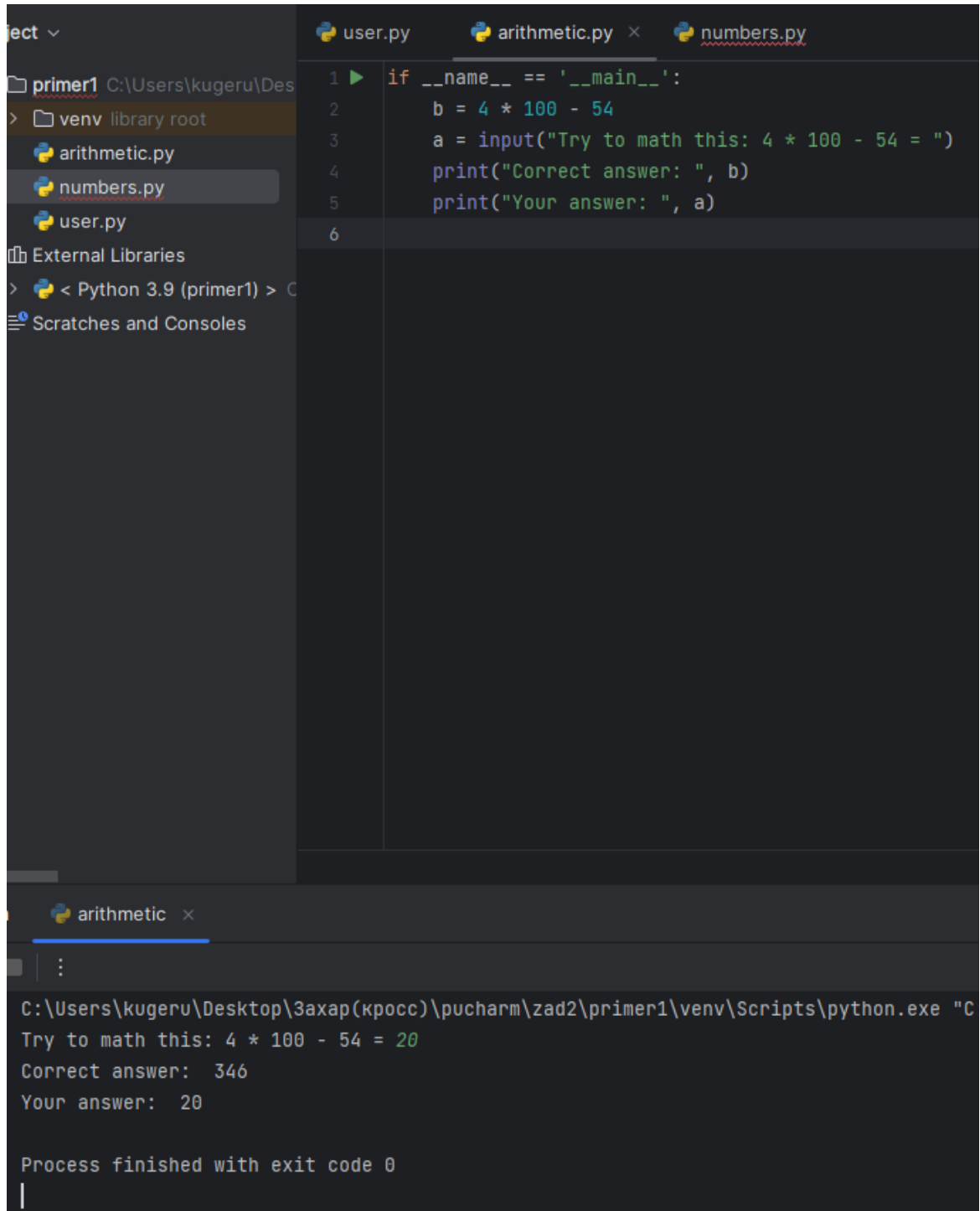
```
1 if __name__ == '__main__':
2     print("What is your name?") # вывод в консоль вопроса
3     a = input() # ввод имени
4     print("How old are you?") # вывод в консоль вопроса
5     b = input() # ввод возраста
6     print("Where are you live?") # вывод в консоль вопроса
7     c = input() # ввод города
8     # вывод в консоль текста и введенное имя
9     print("This is {0}.".format(a))
10    # вывод в консоль текста и введенный возраст
11    print("It is {0}.".format(b))
12    # вывод в консоль текста и введенный город
13    print("(S)he live in {0}.".format(c))
```

The Run window at the bottom shows the execution of the program. The command used is: `C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe"`. The output is:

```
What is your name?
Zahar
How old are you?
22
Where are you live?
Makhachkala
This is Zahar.
It is 22.
(S)he live in Makhachkala.
```

Рисунок 1 – Код для первой задачи и проверка его работоспособности.

Задача 2. 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.



The screenshot shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'primer1' with a 'venv' directory containing 'library root', 'arithmetic.py', 'numbers.py', and 'user.py'. The code editor shows the following code in 'arithmetic.py':

```
1 if __name__ == '__main__':
2     b = 4 * 100 - 54
3     a = input("Try to math this: 4 * 100 - 54 = ")
4     print("Correct answer: ", b)
5     print("Your answer: ", a)
6
```

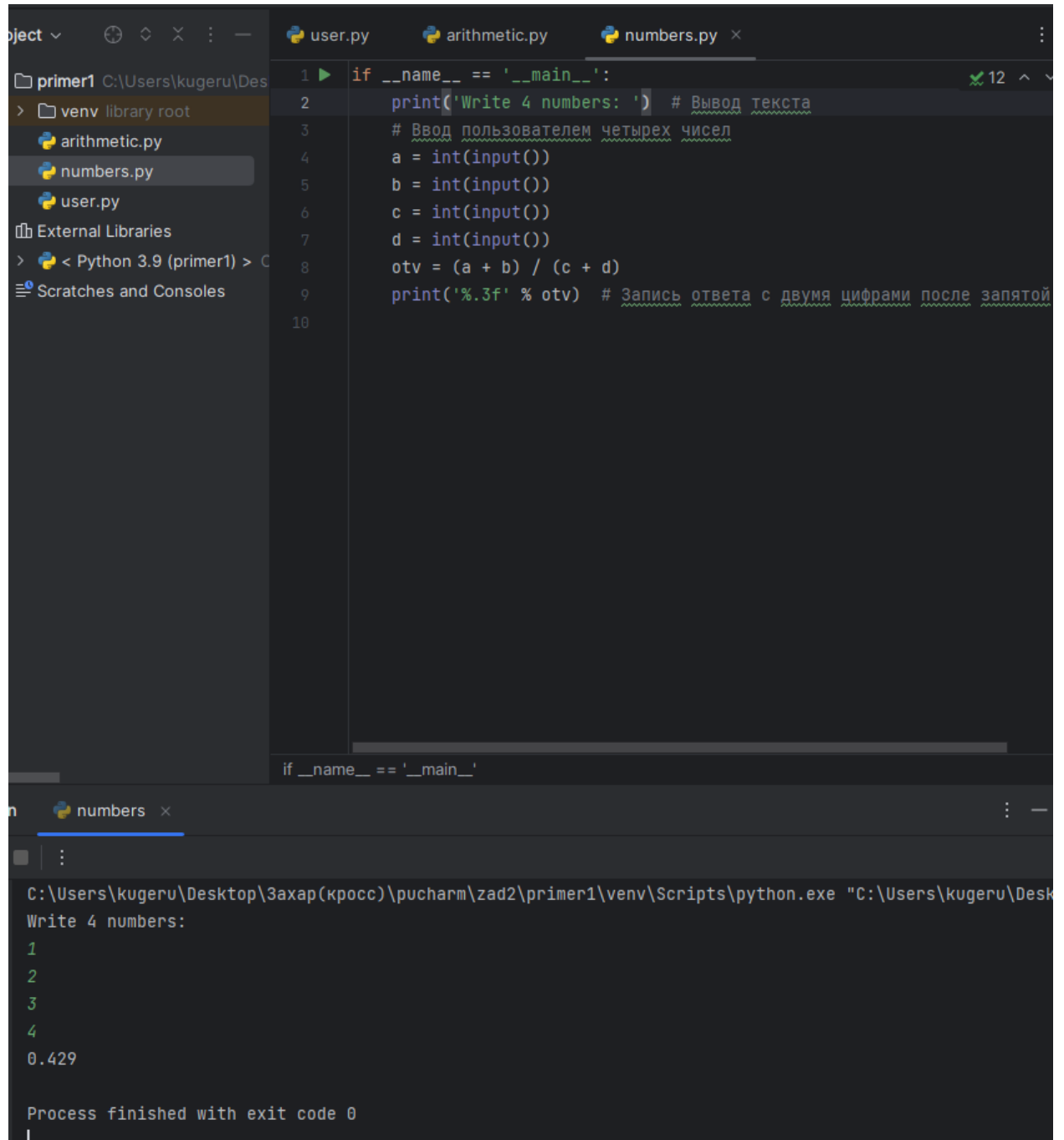
Below the code editor, the output of the program is shown in a console window. The output is:

```
C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe "C
Try to math this: 4 * 100 - 54 = 20
Correct answer:  346
Your answer:  20

Process finished with exit code 0
```

Рисунок 2 – Код для второй задачи и проверка его работоспособности.

Задача 3. 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.



The image shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'primer1' with a subdirectory 'venv' and files 'arithmetic.py', 'numbers.py', and 'user.py'. The code editor shows the contents of 'numbers.py' with the following code:

```
1 if __name__ == '__main__':
2     print('Write 4 numbers: ') # Вывод текста
3     # Ввод пользователем четырех чисел
4     a = int(input())
5     b = int(input())
6     c = int(input())
7     d = int(input())
8     otv = (a + b) / (c + d)
9     print('%.3f' % otv) # Запись ответа с двумя цифрами после запятой
10
```

Below the code editor, the output of the program is shown in a console window. The output is:

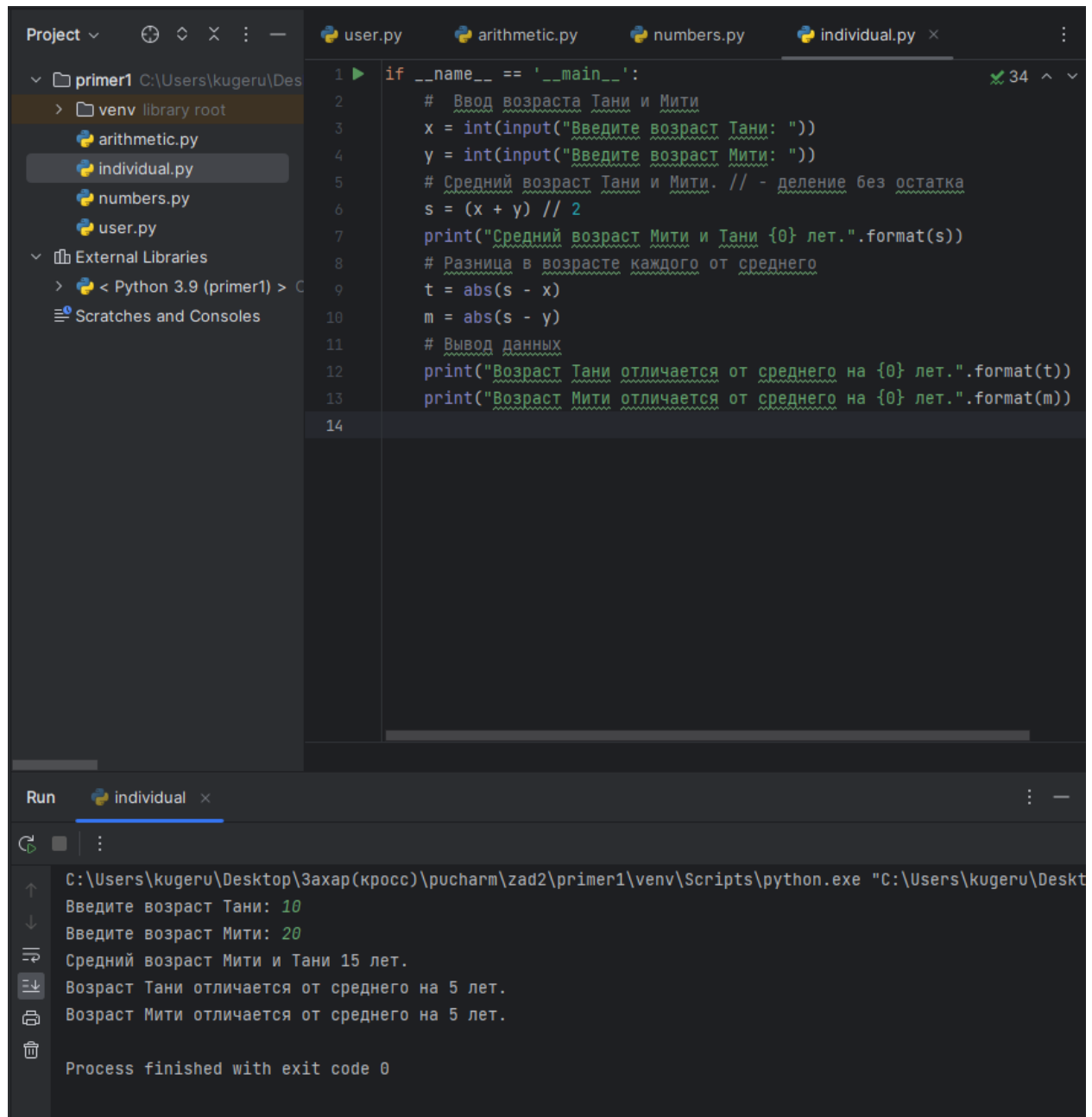
```
C:\Users\kugeru\Desktop\Запах(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Запах(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe"
Write 4 numbers:
1
2
3
4
0.429

Process finished with exit code 0
```

Рисунок 3 – Код для третьей задачи и проверка его работоспособности.

Задача 4. Напишите программу (файл individual.py) для решения индивидуального задания. Вариант индивидуального задания 13.

13. Возраст Тани – X лет, а возраст Мити – Y лет. Найти их средний возраст, а также определить, на сколько отличается возраст каждого ребенка от среднего значения.



```
1 if __name__ == '__main__':
2     # Ввод возраста Тани и Мити
3     x = int(input("Введите возраст Тани: "))
4     y = int(input("Введите возраст Мити: "))
5     # Средний возраст Тани и Мити. // - деление без остатка
6     s = (x + y) // 2
7     print("Средний возраст Мити и Тани {0} лет.".format(s))
8     # Разница в возрасте каждого от среднего
9     t = abs(s - x)
10    m = abs(s - y)
11    # Вывод данных
12    print("Возраст Тани отличается от среднего на {0} лет.".format(t))
13    print("Возраст Мити отличается от среднего на {0} лет.".format(m))
14
```

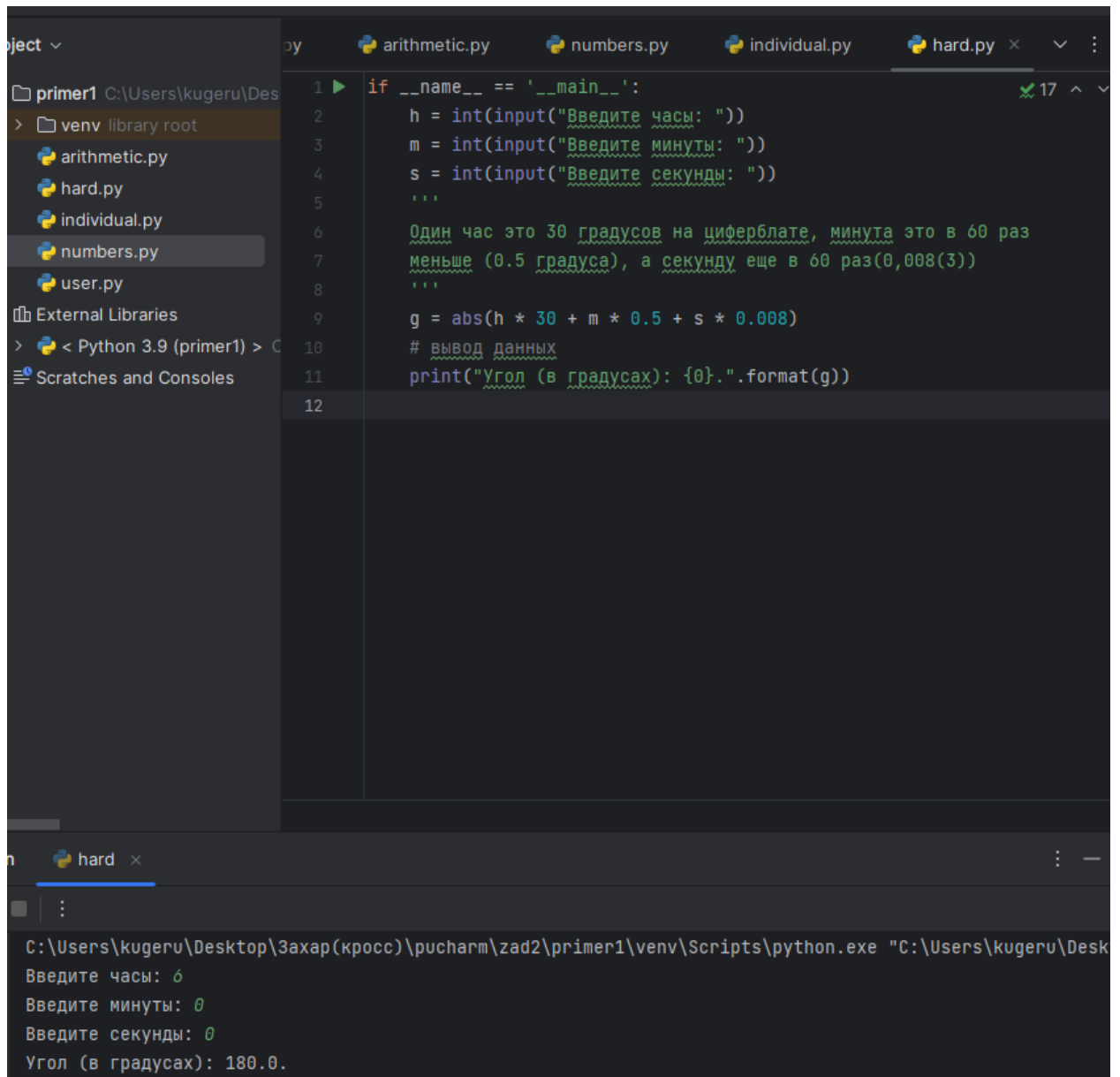
Run individual

```
C:\Users\kugeru\Desktop\Захав(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захав(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe"
Введите возраст Тани: 10
Введите возраст Мити: 20
Средний возраст Мити и Тани 15 лет.
Возраст Тани отличается от среднего на 5 лет.
Возраст Мити отличается от среднего на 5 лет.
Process finished with exit code 0
```

Рисунок 4 – Код для четвертой задачи и проверка его работоспособности.

Задача повышенной сложности:

5. Даны целые числа h, m, s ($0 < h \leq 23, 0 \leq m \leq 59, 0 \leq s \leq 59$), указывающие момент времени: « h часов, m минут, s секунд». Определить угол (в градусах) между положением часовой стрелки в начале суток и в указанный момент времени.



```
1  if __name__ == '__main__':
2      h = int(input("Введите часы: "))
3      m = int(input("Введите минуты: "))
4      s = int(input("Введите секунды: "))
5      ...
6      Один час это 30 градусов на циферблате, минута это в 60 раз
7      меньше (0.5 градуса), а секунду еще в 60 раз(0,008(3))
8      ...
9      g = abs(h * 30 + m * 0.5 + s * 0.008)
10     # вывод данных
11     print("Угол (в градусах): {0}.".format(g))
12
```

hard ×

C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe "C:\Users\kugeru\Desktop\Захар(кросс)\pucharm\zad2\primer1\venv\Scripts\python.exe"

Введите часы: 6

Введите минуты: 0

Введите секунды: 0

Угол (в градусах): 180.0.

Рисунок 5 – Код для задачи повышенной сложности (вариант 13 задача 5) и проверка его работоспособности.



Рисунок 6 – Страница GitHub с выполненными задачами

Ответы на контрольные вопросы:

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

Для начала скачиваем дистрибутив по ссылке. Запускаем скачанный файл. Выбираем способ установки (install now или customize installation). Отмечаем необходимые опции для установки. Выбираем место установки (доступно при выборе Customize installation). successful

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

Anaconda наиболее известна как дистрибутив Python со встроенным в него пакетным менеджером conda. Она позволяет изолировать окружение проекта от системной версии Python, который критически необходим для работы системы. Использование sudo pip считается плохой практикой. Также

conda позволяет без проблем переносить окружение с одной машины на другую. Кроме того, если вы что-то сломаете, то с Anaconda вы всегда сможете откатиться на более старую версию окружения. Конечно, если вы позаботитесь о регулярных бэкапах. С системной версией Python это гораздо сложнее и может потребовать переустановки системы.

3. Как осуществить проверку работоспособности пакета Anaconda?

Проверка Anaconda

Для выполнения проверки работоспособности Anaconda необходимо вначале запустить командный процессор с поддержкой виртуальных окружений Anaconda. В Windows это можно сделать выбрав следующий пункт главного меню системы *Пуск → Anaconda3 (64-bit) → Anaconda Prompt*. В появившейся командной строке необходимо ввести

```
> jupyter notebook
```

в результате чего отобразится процесс загрузки веб-среды Jupyter Notebook

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

Запускаем программу. И он не содержит ни одного файла с текстом программы. Создадим его. Нажимаем правую кнопку мыши, выбираем «New» -> «Python File». Вводим имя файла, например, ex1 и этот файл автоматически добавляется в наш проект. Здесь мы можем писать наши программы, на Питоне.

5. Как осуществить запуск программы с помощью IDE PyCharm?

Здесь при первом запуске необходимо создать новый проект. Нажимаем «Create New Project». В поле «Location» указывается расположение проекта и его имя.

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

Его суть в удобстве использования языка Питон, а именно его внешний вид и простота, сопровождаемая автоматикой действий, позволяющей пользователям работать с данными программами.

7. Почему язык программирования Python называется языком динамической типизации?

Потому что в Питоне тип переменной определяется непосредственно при выполнении программы.

8. Какие существуют основные типы в языке программирования Python?

В Python типы данных можно разделить на встроенные в интерпретатор (built-in) и не встроенные, которые можно использовать при импортировании соответствующих модулей.

К основным встроенным типам относятся:

1. None (неопределенное значение переменной)

2. Логические переменные (Boolean Type)

3. Числа (Numeric Type)

- int – целое число
- float – число с плавающей точкой
- complex – комплексное число

4. Списки (Sequence Type)

- list – список
- tuple – кортеж
- range – диапазон

5. Строки (Text Sequence Type)

- str

6. Бинарные списки (Binary Sequence Types)

- bytes – байты
- bytearray – массивы байт
- memoryview – специальные объекты для доступа к внутренним данным

объекта через protocol buffer

7. Множества (Set Types)

- set – множество
- frozenset – неизменяемое множество

8.Словари (Mapping Types)

- dict – словарь

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

Для того, чтобы объявить и сразу инициализировать переменную необходимо написать её имя, потом поставить знак равенства и значение, с которым эта переменная будет создана. Например строка: `b=5` Целочисленное значение 5 в рамках языка Python по сути своей является объектом. Объект, в данном случае – это абстракция для представления данных, данные – это числа, списки, строки и т.п. При этом, под данными следует понимать как непосредственно сами объекты, так и отношения между ними (об этом чуть позже). Каждый объект имеет три атрибута – это идентификатор, значение и тип. Идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга, а значение – непосредственно информация, хранящаяся в памяти, которой управляет интерпретатор.

10. Как получить список ключевых слов в Python?

Список ключевых слов можно получить непосредственно в программе, для этого нужно подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

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

Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`. Тип переменной можно определить с помощью функции `type()`.

12. Что такое изменяемые и неизменяемые типы в Python.

К неизменяемым (`immutable`) типам относятся: целые числа (`int`), числа с плавающей точкой (`float`), комплексные числа (`complex`), логические переменные (`bool`), кортежи (`tuple`), строки (`str`) и неизменяемые множества (`frozen set`). К изменяемым (`mutable`) типам относятся: списки (`list`), множества (`set`), словари (`dict`). Как уже было сказано ранее, при создании переменной,

вначале создается объект, который имеет уникальный идентификатор, тип и значение, после этого переменная может ссылаться на созданный объект.

13. Чем отличаются операции деления и целочисленного деления?

Целочисленное деление (`div`) отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Перед выполнением операции оба операнда округляются до целых значений.

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

Для создания комплексного числа можно использовать функцию `complex(a, b)`, в которую, в качестве первого аргумента, передается действительная часть, в качестве второго – мнимая. Либо записать число в виде $a + bj$.

15. Каково назначение и основные функции библиотеки (модуля) `math`? По аналогии с модулем `math` изучите самостоятельно назначение и основные функции модуля `cmath`.

В стандартную поставку Python входит библиотека `math`, в которой содержится большое количество часто используемых математических функций. Для работы с данным модулем его предварительно нужно импортировать.

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

`sep` – с помощью этого параметра вы можете указать разделитель строк. А по умолчанию в качестве разделителя используется пробел `end` – этот параметр позволяет указать, что нужно добавить после последней строки. По умолчанию добавляется управляющий символ `'\n'` (перевод строки)

17. Каково назначение метода `format()`? Какие еще существуют средства для форматирования строк в Python? Примечание: в дополнение к рассмотренным средствам изучите самостоятельно работу с f-строками в Python.

Метод `format()`, который определен у строк, позволяет форматировать строку, вставляя в нее на место плейсхолдеров определенные значения. Для вставки в строку используются специальные параметры, которые обрамляются фигурными скобками (`{}`).

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

Для ввода нужно нажать Enter после завершения набора текста. Обычно Enter добавляет символ новой строки (`\n`), но не в этом случае. Введенная строка просто будет передана приложению.

Вывод: исследовали процесс установки и базовый возможности языка Python3.