

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2

Выполнила студентка группы ИТС-б-о-
20-1 (2)

Маслова А.В. « » _____ 2021г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент
Кафедры инфокоммуникаций Воронкин
Р.А.

(подпись)

г. Ставрополь, 2021

Лабораторная работа 2.

Основы языка Python3

Цель работы: исследование процесса установки и базовых возможностей языка Python версии 3. <https://github.com/alina-j/main>

Порядок выполнения работы:

1. Установка Python

1) Загрузка приложения с официального сайта

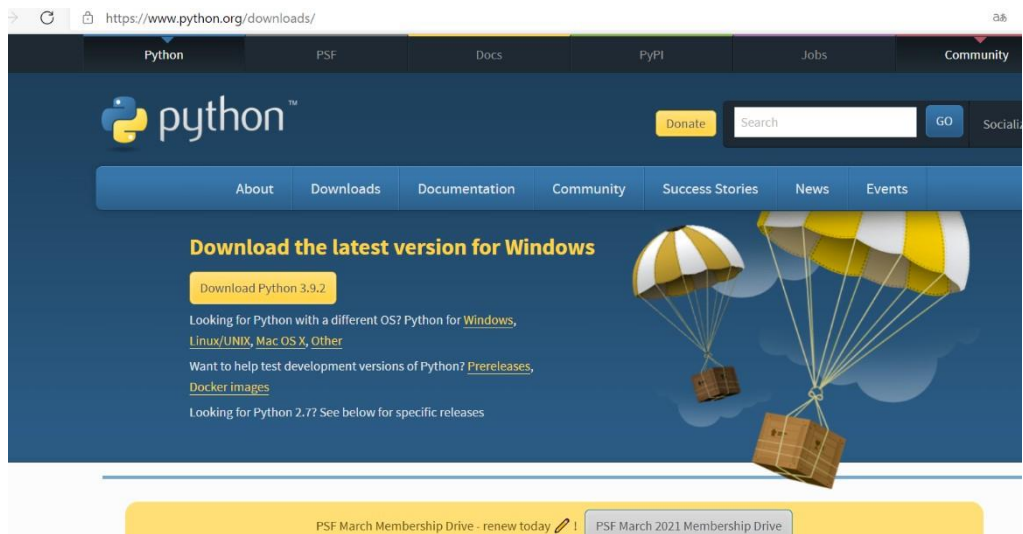


Рисунок 1 – официальный сайт Python

2) Выбираем 2-й способ установки



Рисунок 2 – способ установки

3) Отмечаем все опции установки

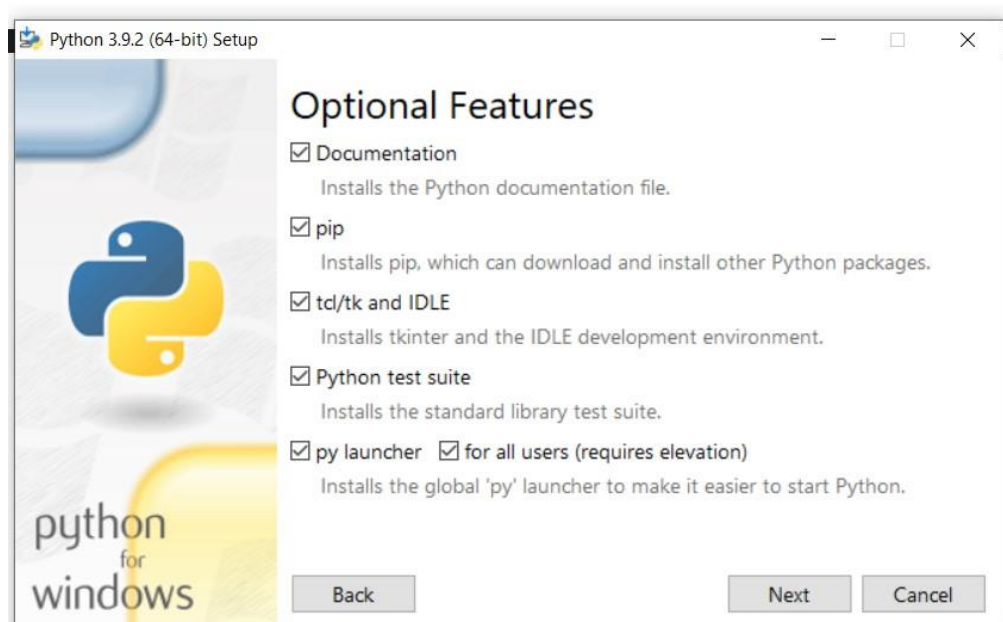


Рисунок 3 – опции установки

4) выбираем место установки: Install for all users, Associate files with Python, Create shortcuts for installed applications, Add Python to environment variables, Precompile standard library;

5) выбираем место на диске;

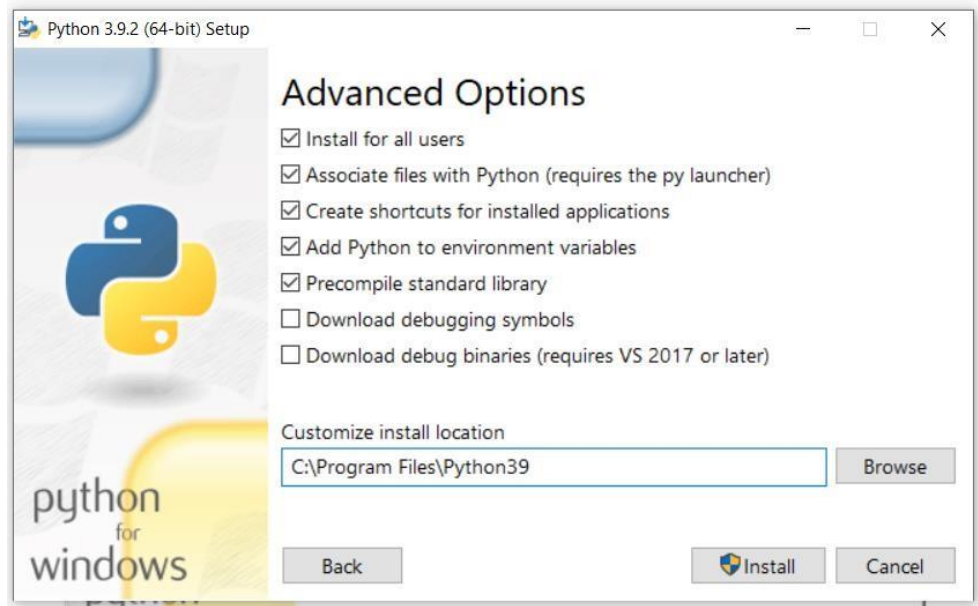


Рисунок 4 – место установки

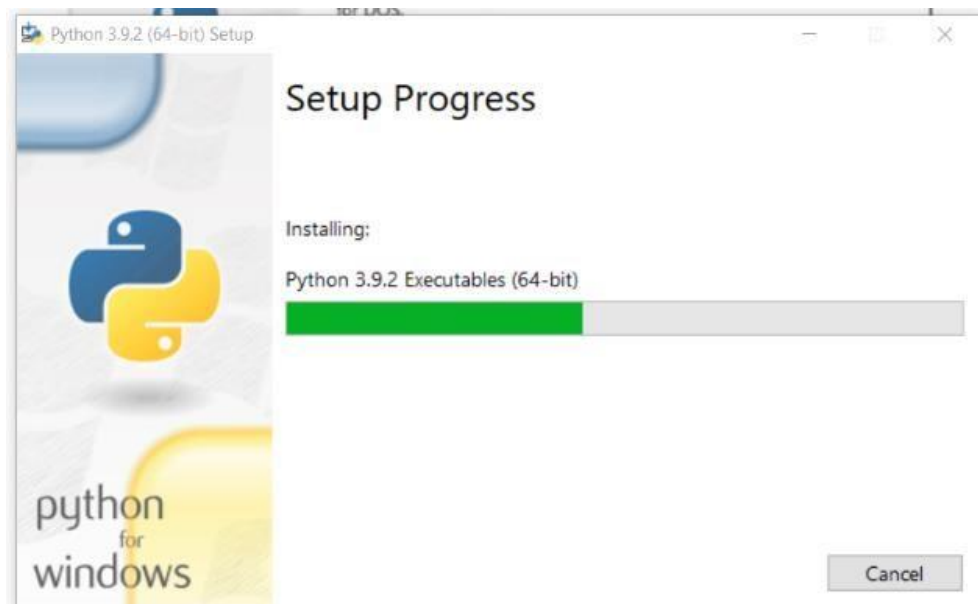


Рисунок 5 – установка

б) после установки нажимаем на “Close”;

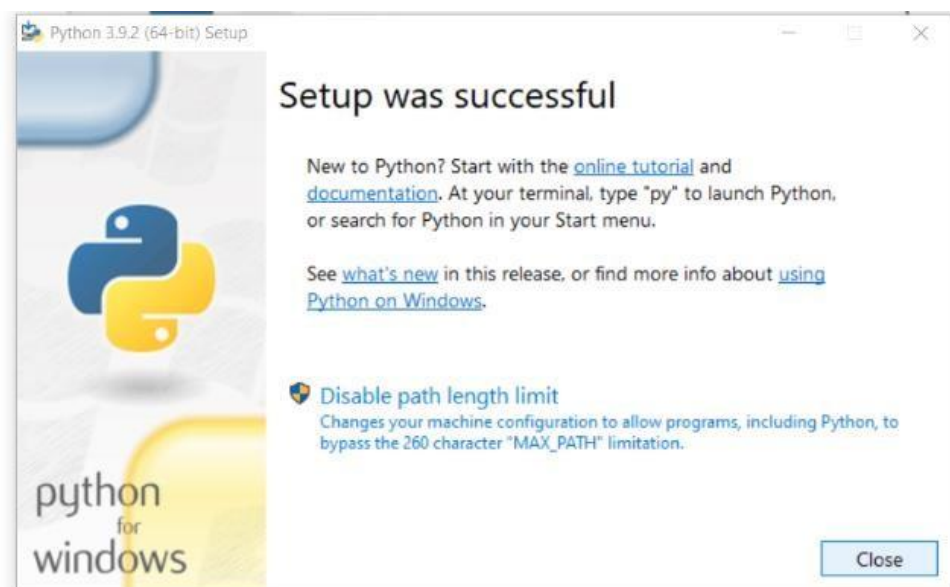


Рисунок 6 – завершение установки

2. Установка Anaconda

- 1) переходим на сайт <https://www.anaconda.com/products/individual> и нажимаем Download;



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

Рисунок 7 – официальный сайт Anaconda;

2) запускаем скачанный инсталлятор и нажимаем Next;

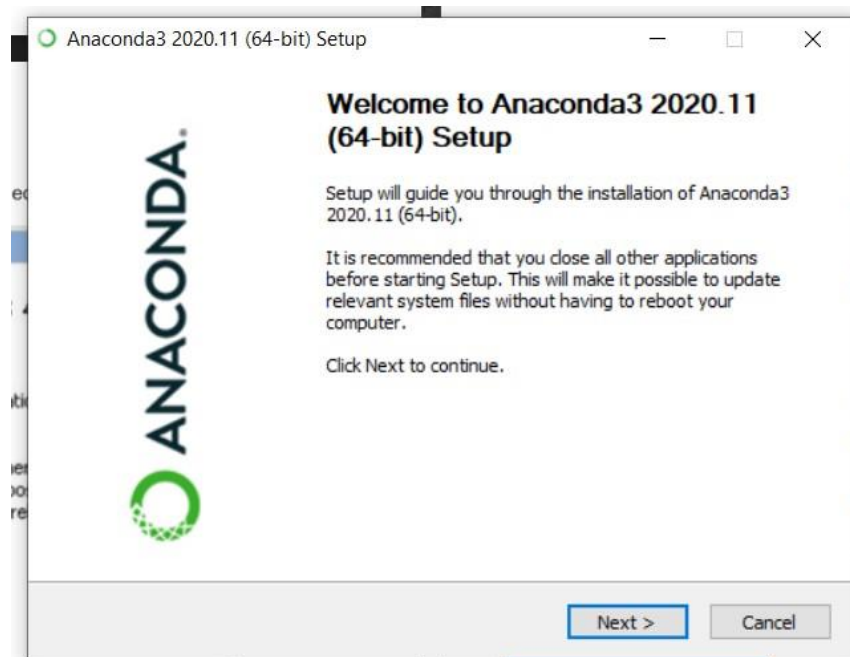


Рисунок 8 – скачанный инсталлятор

3) принимаем лицензионное соглашение «I Agree»;

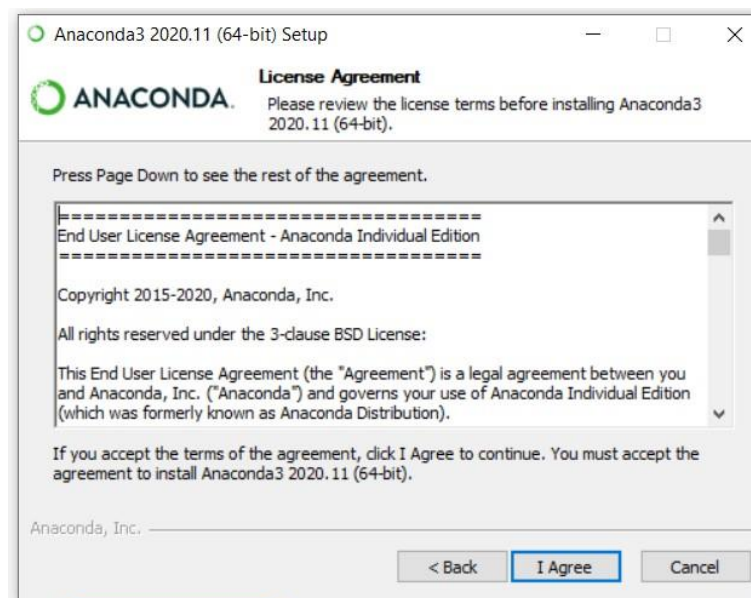


Рисунок 9 – лицензионное соглашение

- 4) выбираем одну из опции установки «All Users» и нажимаем «Next»;

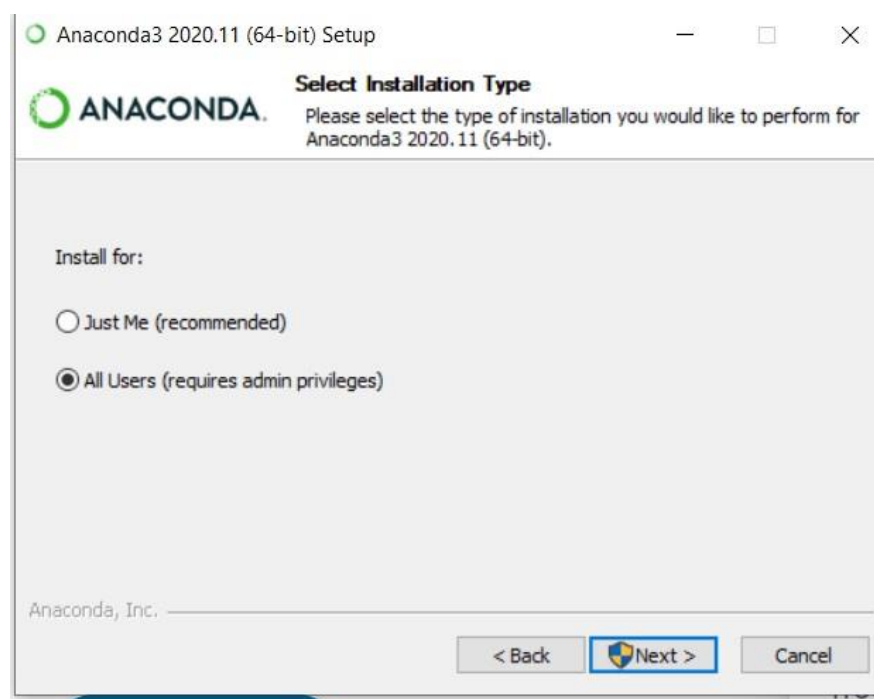


Рисунок 10.– Опция установки

- 5) указываем путь, по которому будет установлена Anaconda и нажимаем Next;

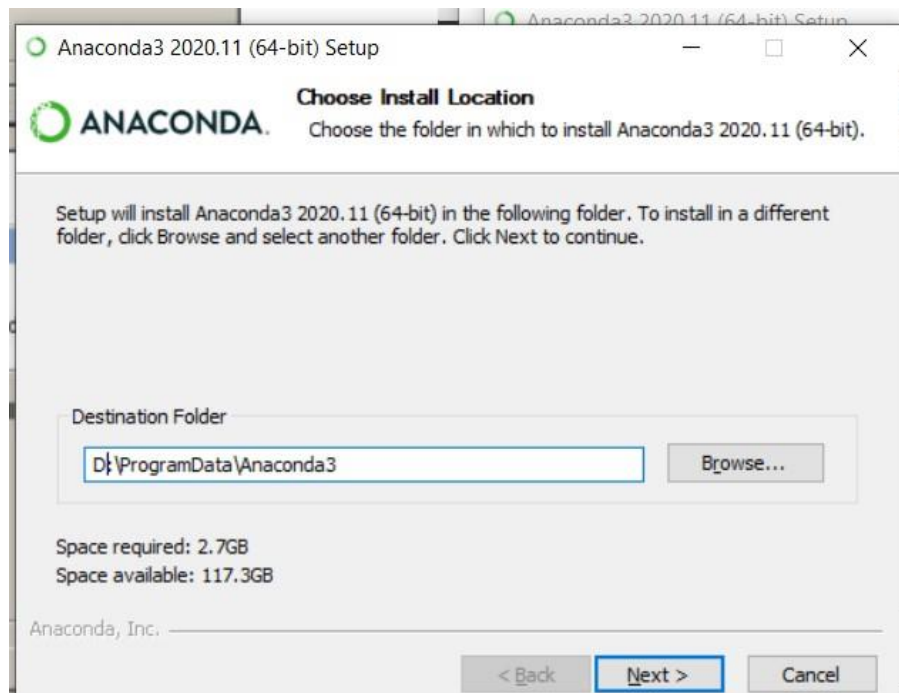


Рисунок 11 – Выбор пути для установки Anaconda

- 6) указываем дополнительные опции «Add Anaconda to the system PATH environment variable» и «Register Anaconda as the system Python 3.5 » и нажимаем «Install»;

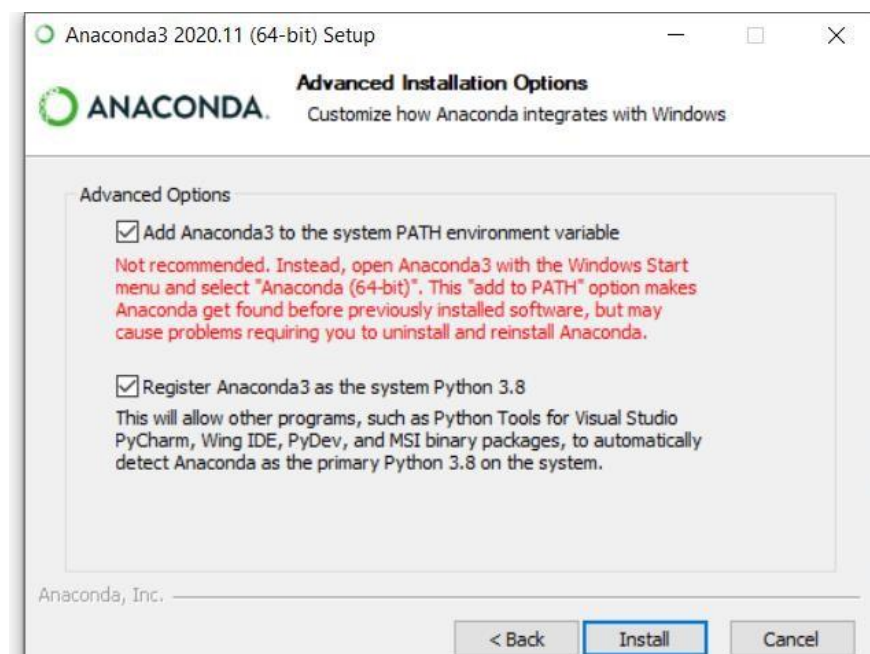


Рисунок 12 – Дополнительные опции

- 7) при завершении нажимаем на «Finish»;

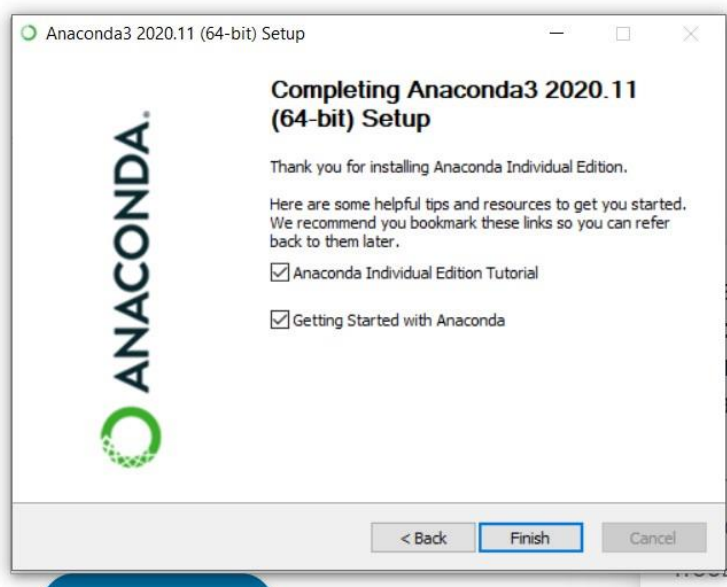


Рисунок 13 – Завершение установки

3. Установка PyCharm

1) переходим по ссылке <https://www.jetbrains.com/pycharm/download/#section=windows> ;

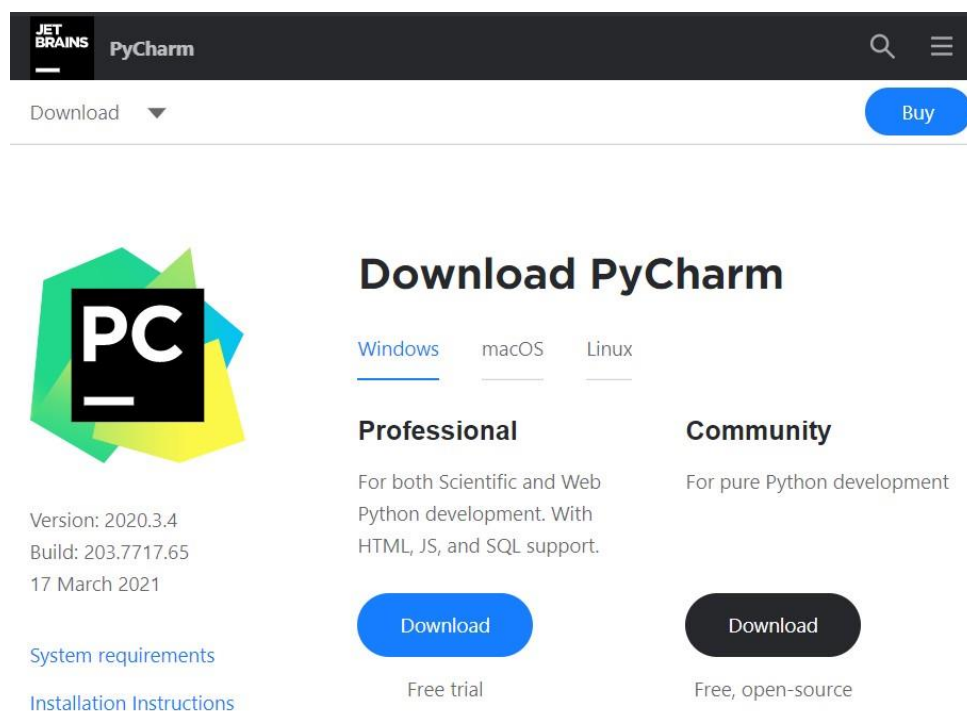


Рисунок 14 – Официальный сайт PyCharm

2) запускаем скачанный дистрибутив PyCharm;

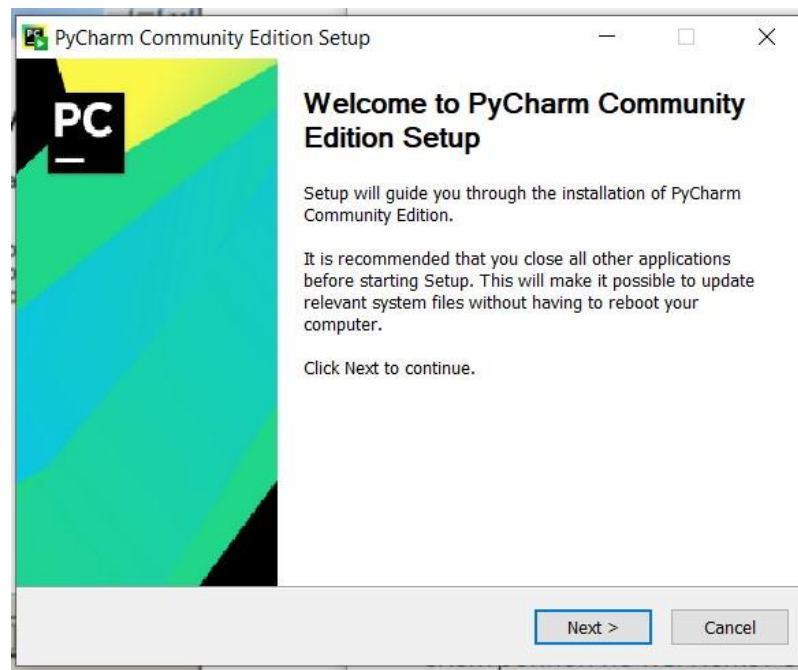


Рисунок 15 – Скачанный дистрибутив

3) выбираем путь установки программы;

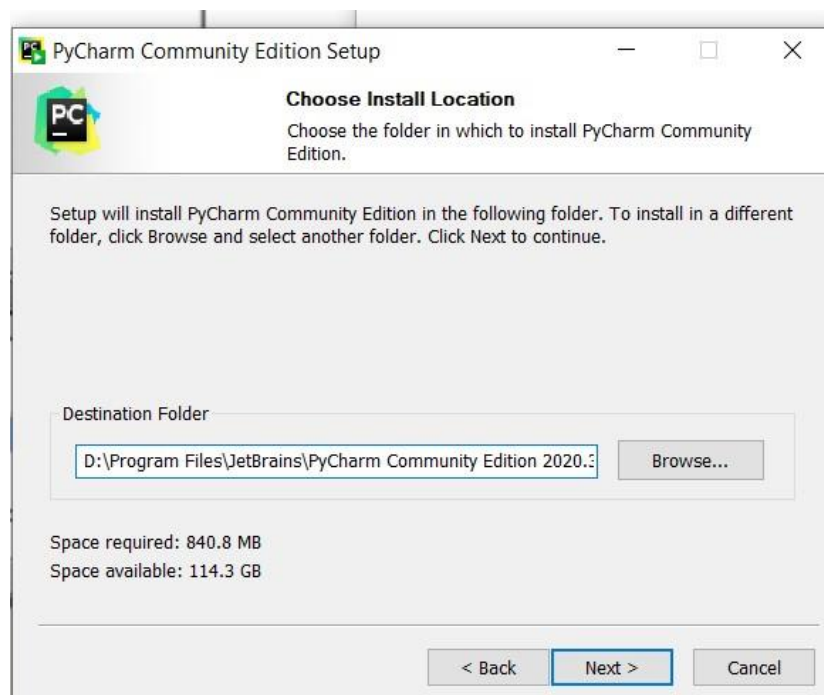


Рисунок 16 – Путь установки

4) указываем ярлыки, которые нужно создать на рабочем столе;

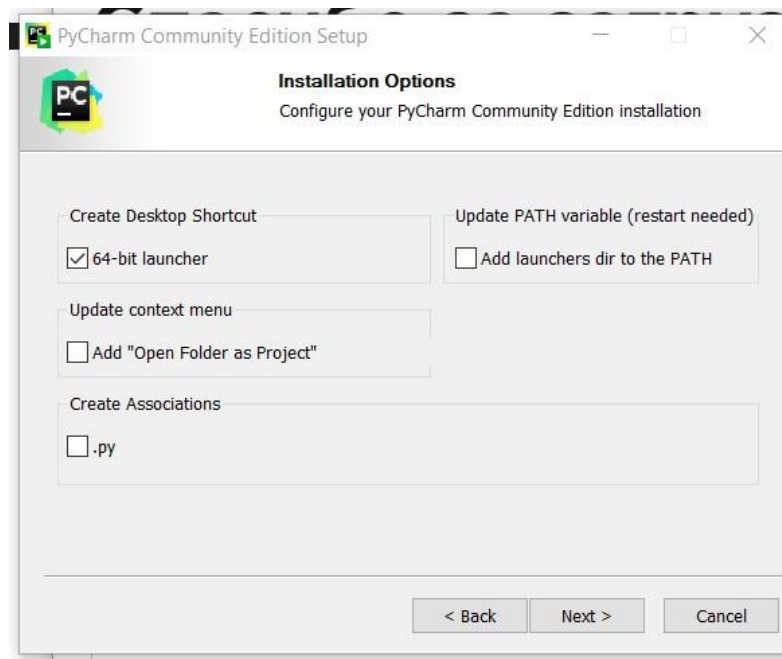


Рисунок 17 – Запуск версии PyCharm

5) выбираем имя для папки в меню Пуск;

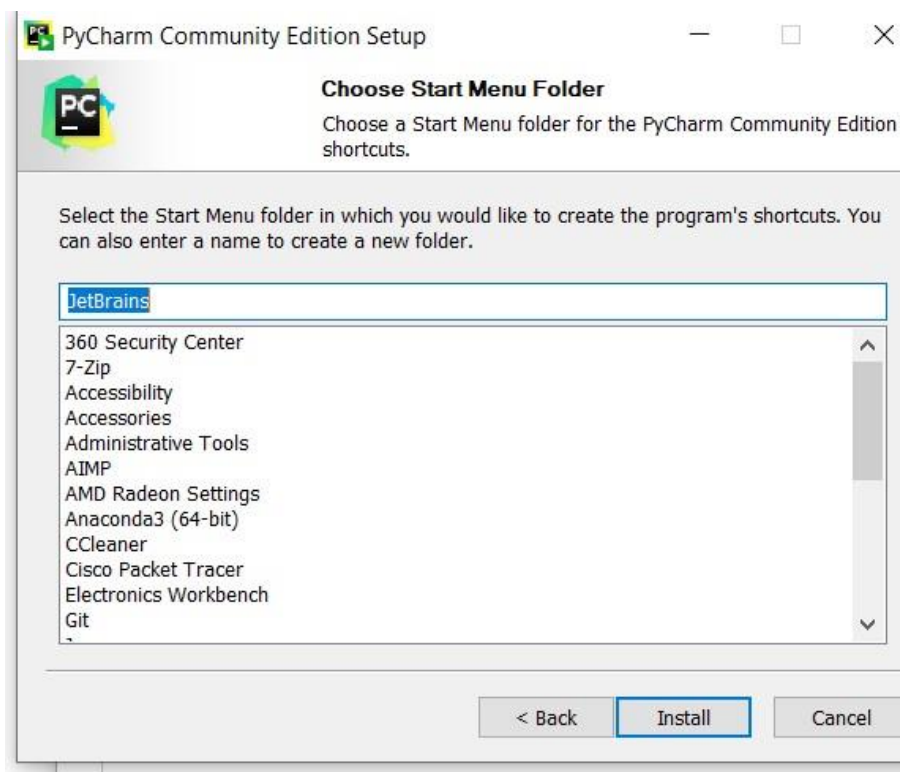


Рисунок 18 – Имя для папки

6) завершаем установку и нажимаем «Finish»;



Рисунок 19 – Завершение установки

1.

7. 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`"
```

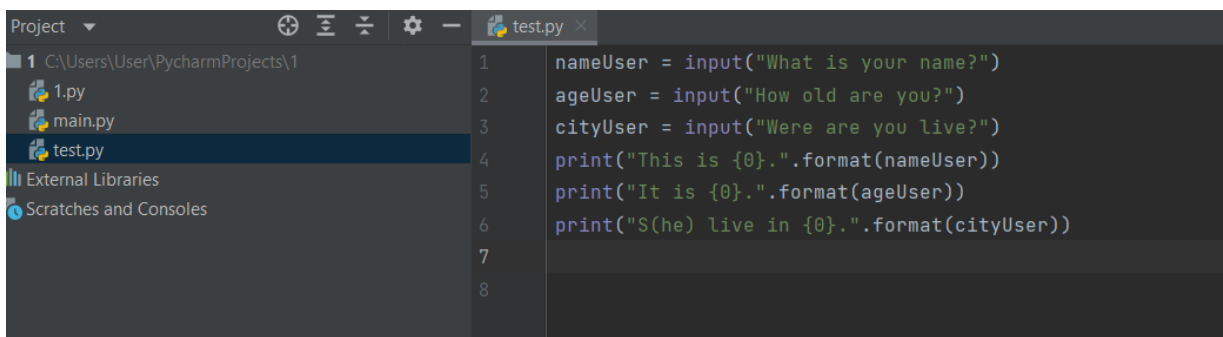
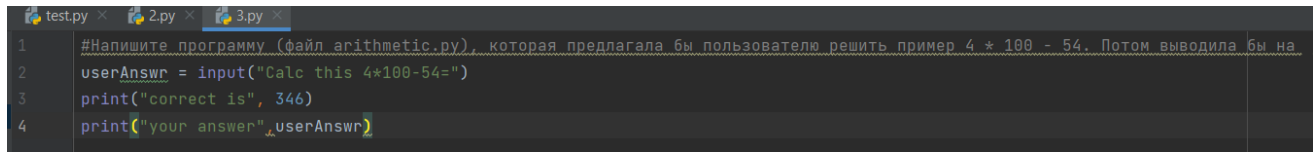


Рисунок 1.1 – программа 1

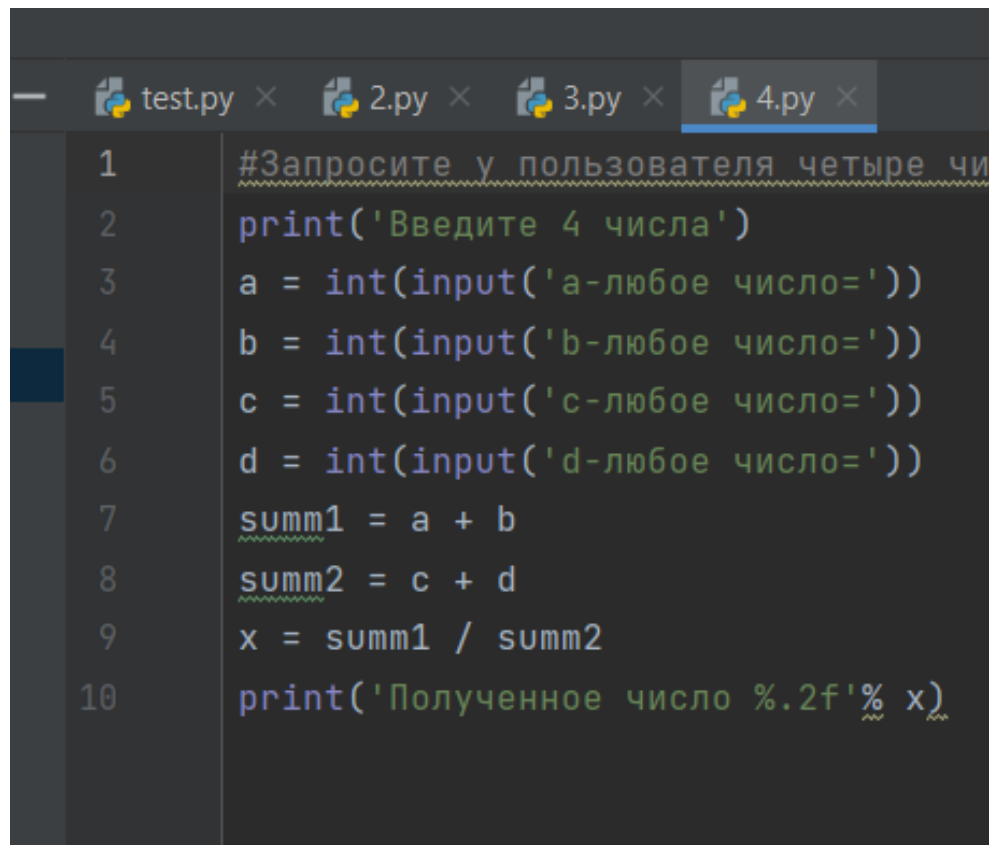
Задача 2. Напишите программу (файл arithmetic.py), которая предлагала бы пользователю решить пример $4 * 100 - 54$. Потом выводила бы на экран правильный ответ и ответ пользователя.



```
1 #Напишите программу (файл arithmetic.py), которая предлагала бы пользователю решить пример 4 * 100 - 54. Потом выводила бы на
2 userAnswr = input("Calc this 4*100-54=")
3 print("correct is", 346)
4 print("your answer", userAnswr)
```

Рисунок 1.2 – программа 2

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.

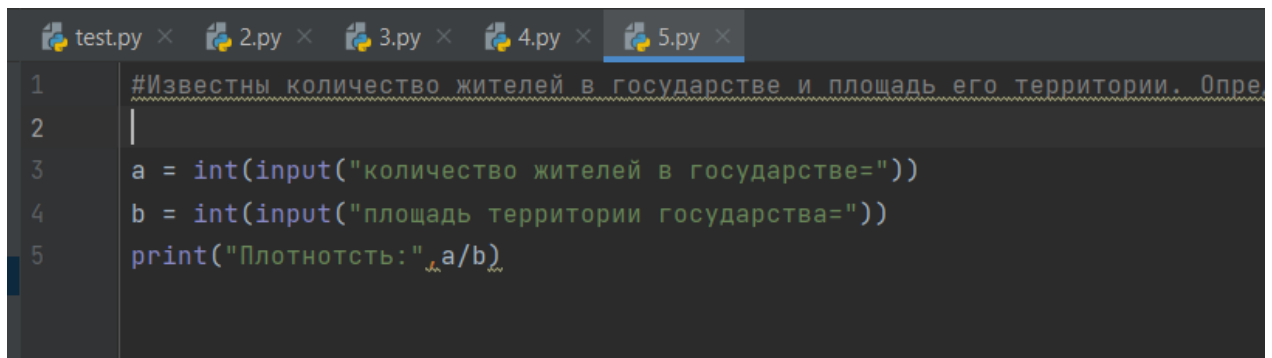


```
1 #Запросите у пользователя четыре чи
2 print('Введите 4 числа')
3 a = int(input('a-любое число='))
4 b = int(input('b-любое число='))
5 c = int(input('c-любое число='))
6 d = int(input('d-любое число='))
7 summ1 = a + b
8 summ2 = c + d
9 x = summ1 / summ2
10 print('Полученное число %.2f'% x)
```

Рисунок 1.3.— программа 3

Индивидуальное задание. Вариант 19.

Известны количество жителей в государстве и площадь его территории. Определить плотность населения в этом государстве.

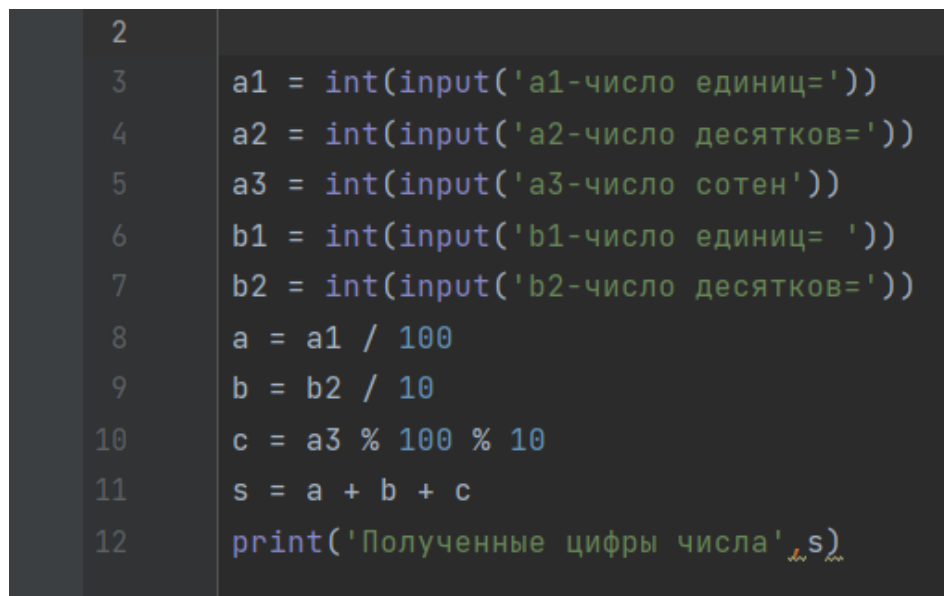
A screenshot of a Python IDE with multiple tabs. The active tab is '5.py'. The code in the editor is as follows:

```
1 #Известны количество жителей в государстве и площадь его территории. Опре
2 |
3 a = int(input("количество жителей в государстве="))
4 b = int(input("площадь территории государства="))
5 print("Плотность:" ,a/b)
```

Рисунок 1.4.– программа 4

Задача повышенной сложности. Вариант 3.

Даны цифры двух десятичных целых чисел: трехзначного и двузначного, где a_1 и b_1 – число единиц, a_2 и b_2 – число десятков, a_3 – число сотен. Получить цифры числа, равного сумме заданных чисел (известно, что это число трехзначное). Числа-слагаемые и число-результат не определять; условный оператор не использовать.

A screenshot of a Python IDE showing a program that takes input for two numbers and outputs the digits of their sum. The code is as follows:

```
2
3 a1 = int(input('a1-число единиц='))
4 a2 = int(input('a2-число десятков='))
5 a3 = int(input('a3-число сотен='))
6 b1 = int(input('b1-число единиц= '))
7 b2 = int(input('b2-число десятков='))
8 a = a1 / 100
9 b = b2 / 10
10 c = a3 % 100 % 10
11 s = a + b + c
12 print('Полученные цифры числа',s)
```

Рисунок 1.5 – программа 5

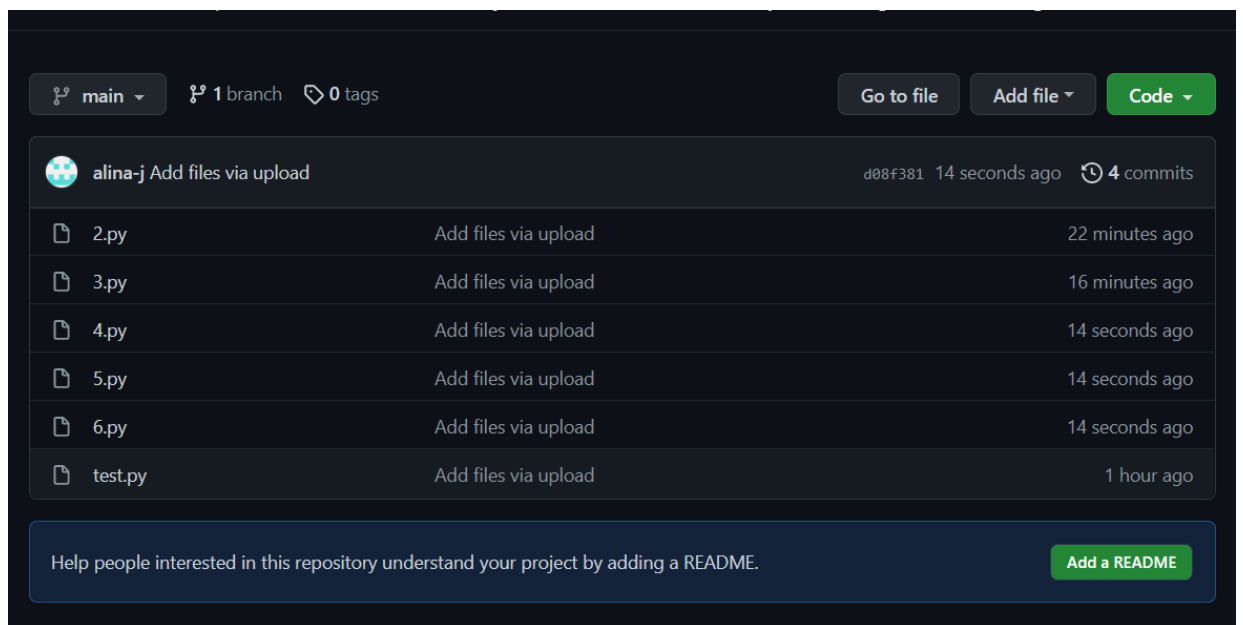


Рисунок 1.6 – GitHub с выполненными заданиями

Вывод: в ходе выполнения данной лабораторной работы были проведены исследования процессов установки Python, Anaconda, PyCharm, также изучены базовые возможности языка Python.

Ответы на контрольные вопросы:

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

Порядок установки:

- запускаем скачанный установочный файл;
- выбираем способ установки;
- указываем путь установки; – устанавливаем Python.

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

Anaconda является дистрибутивом языков программирования таких как python и R, содержит набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, а python- это язык программирования высокого уровня общего назначения.

3. Как осуществить проверку работоспособности пакета Anaconda?

- запускаем программу Anaconda Prompt;
- в командной строке вводим: «jupyter notebook»;
- в открывшемся веб-сервисе создаем новый ноутбук;
- далее вводим пробный код: «print(“Hello, World!”)»;
- если все выполнено правильно, то на экране появится надпись «Hello, World!», а это значит, что программа работает.

4. Как задать используемый интерпретатор языка Python в IDE PyCharm? – используемый интерпретатор языка Python в IDE PyCharm задается при создании нового проекта;

5. Как осуществить запуск программы с помощью IDE PyCharm?

- создаем код программы;
- нажимаем на кнопку Run;

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

В интерактивном режиме работы Python ожидает ввода команд пользователя. При вводе команды интерпретатор выполнит строку и отобразит строкой ниже результат своей работы.

В пакетном режиме работы Python будет только выполнять уже написанный код. Для этого нужно набрать в командной строке ”python «название файла».py”.

7. Почему язык программирования Python называется языком динамической типизации?

Ответ: потому что язык поддерживает несколько парадигм программирования: структурное, функциональное, объектно-ориентированное, аспектно-ориентированное. В нем переменная связывается с типом в момент присваивания значения.

8. Какие существуют основные типы в языке программирования Python?

- строки;
- логические переменные;
- числа;
- списки;
- бинарные списки;
- неопределенные значения переменной.

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

Для того чтобы создать объект, необходимо написать его имя, потом поставить знак равенства и значение, с которым объект будет создан.

10. Как получить список ключевых слов в Python?

Необходимо подключить модуль `keyword` и воспользоваться командой `keyword.kwlist`.

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

Данные функции создают идентификатор объекту и тип переменной;

12. Что такое изменяемые и неизменяемые типы в Python.

Изменяемые типы данных – типы данных, которые могут изменяться в процессе выполнения кода программы. К ним относятся списки (`list`), множества (`set`) и словари (`dict`).

Неизменяемые типы данных – типы данных, неизменяемые в процессе выполнения кода программы. К ним относятся целые числа (`int`), числа с плавающей точкой (`float`), комплексные числа (`complex`), логические переменные (`bool`), кортежи (`tuple`), строки (`str`) и неизменяемые множества (`frozen set`).

13. Чем отличаются операции деления и целочисленного деления?

Деление возвращает частное с его дробной частью (при наличии).

Целочисленное деление возвращает целую часть частного, а дробная часть отбрасывается.

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

Используются функции `complex(a, b)`, `x.real`, `x.imag`, `x.conjugate()`.

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

Функции `math`:

– `math.celi(x)`, `math.fabs(x)`, `math.factorial(x)`, `math.floor(x)`, `math.exp(x)`, `math.log2(x)`, `math.log10(x)`, `math.log(x[, base])`, `math.pow(x,y)`, `math.sqrt(x)`, `math.sin(x)`, `math.cos(x)`, `math.tan(x)`, `math.acos(x)`, `math.asin(x)`, `math.atan(x)`, `math.pi`, `math.e`— тригонометрические.

Библиотека `math` содержит в себе большое количество часто используемых математических функций.

Модуль `cmath` работает с комплексными числами

Основные функции `cmath`: `cmath.phase(x)`, `cmath.polar(x)`, `cmath.rect(x)`, `cmath.exp(x)`, `cmath.log(x[, base])`, `cmath.acosh(x)`, `cmath.asinh(x)`, `cmath.atanh(x)`.

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

Параметр `sep` указывает отличный от пробела разделитель строк.

Параметр `end` указывает, что делать после вывода строки (поставить знак, сделать перенос на несколько строк).

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()` применяется к строке и позволяет подставлять типы данных в неё (`%s`, `%d`, `%f`), указывать количество знаков после запятой для чисел (`%1f`).

F-строки позволяют форматировать строки схожим способом, как `format()`, но с некоторыми отличиями. Они позволяют подставить значение в строку, лишь указав имя переменной в фигурных скобках, использовать расширенное форматирование чисел, форматировать дату без вызова метода `strftime()`, совершать базовые арифметические операции прямо в строке, обращаться к значениям списков по индексу и элементам словаря по ключу, и т.д.

18. Каким образом осуществить ввод с консоли значения целочисленной и вещественной переменной в языке Python?— при помощи функции `input()`.