# Лабораторная работа 1

# «Среда программирования. Работа в режиме интерпретатора на языке Python»

**Цель работы:**

* Получить практические навыки установки интерпретатора Python
* Получить практические навыки написание простейших программ в режиме интерпретатора Python
* Получить практические навыки установки среды разработки PyCharm
* Изучить основные приемы работы в среде разработки PyCharm
* Познакомиться со структурой программы на языке Python.
* Научиться составлять простейшие программы на языке Python.
* Получить практические навыки создания, редактирования, тестирования и отладки консольных приложений на языке Python
* Освоить организацию ввода / вывода в консольных приложениях на Python
* Получить практические навыки форматирования строк вывода в Python
* Получить практические навыки обработки математических выражений и работы в интеллектуальном калькуляторе Python.

## Краткие теоретические сведения

В лабораторных работах курса «Программирование на языках высокого уровня» будет использоваться тип приложений операционной системы Windows – приложения с интерфейсом командной строки, так называемые **консольные приложения**.

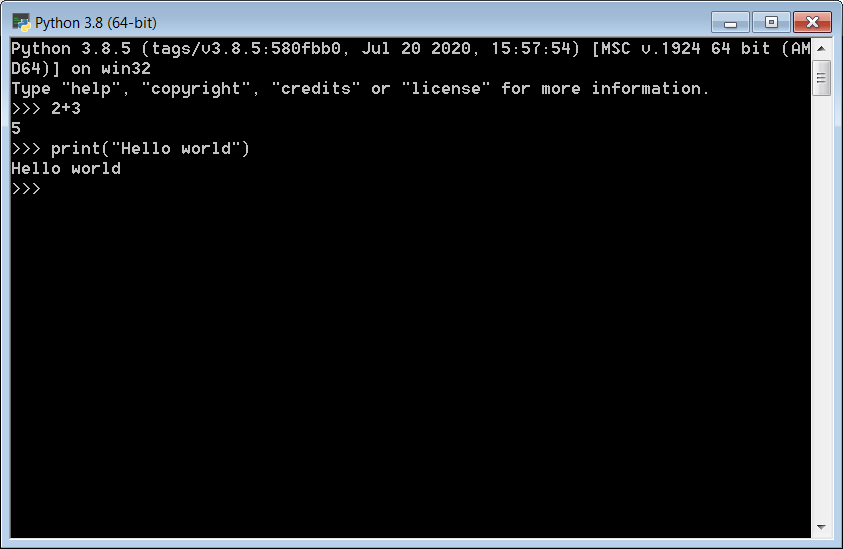
1. **Установка и запуск Python**

Для написания и выполнения программ на языке Python необходимо скачать и установить:

* интерпретатор Python - сайт «python.org» – в меню «Downloads»
* среду разработки (IDE) – PyCharm www.jetbrains.com/. Community edition – бесплатная версия

Для установки интерпретатора языка Python необходимо скачать (https://www.python.org/downloads/) с официального сайта установщик версии 3.8.5 (на данный момент самая последняя версия). В процессе установки выберите соответствующий пункт для того, чтобы путь к интерпретатору был добавлен в переменную среды PATH.

Для установки среды разработки PyCharm необходимо скачать последнюю версию с официального сайта (https://www.jetbrains.com/pycharm/download/) для своей операционной системы. Редакция Community является наиболее приемлемым вариантом

**Интерпретатор Python**

1. **Создание нового проекта в PyCharm**

Создание проекта и его запуск. Для того, чтобы создать проект необходимо запустить среду PyCharm и выбрать пункт Create New Project

|  |  |
| --- | --- |
|  | а) |

В поле Location нужно указать путь к новому проекту, в поле Interpreter выбрать установленный интерпретатор языка Python и нажать Create.

|  |  |
| --- | --- |
|  | б) |

Рис. 1. – Создание нового приложения в PyCharm

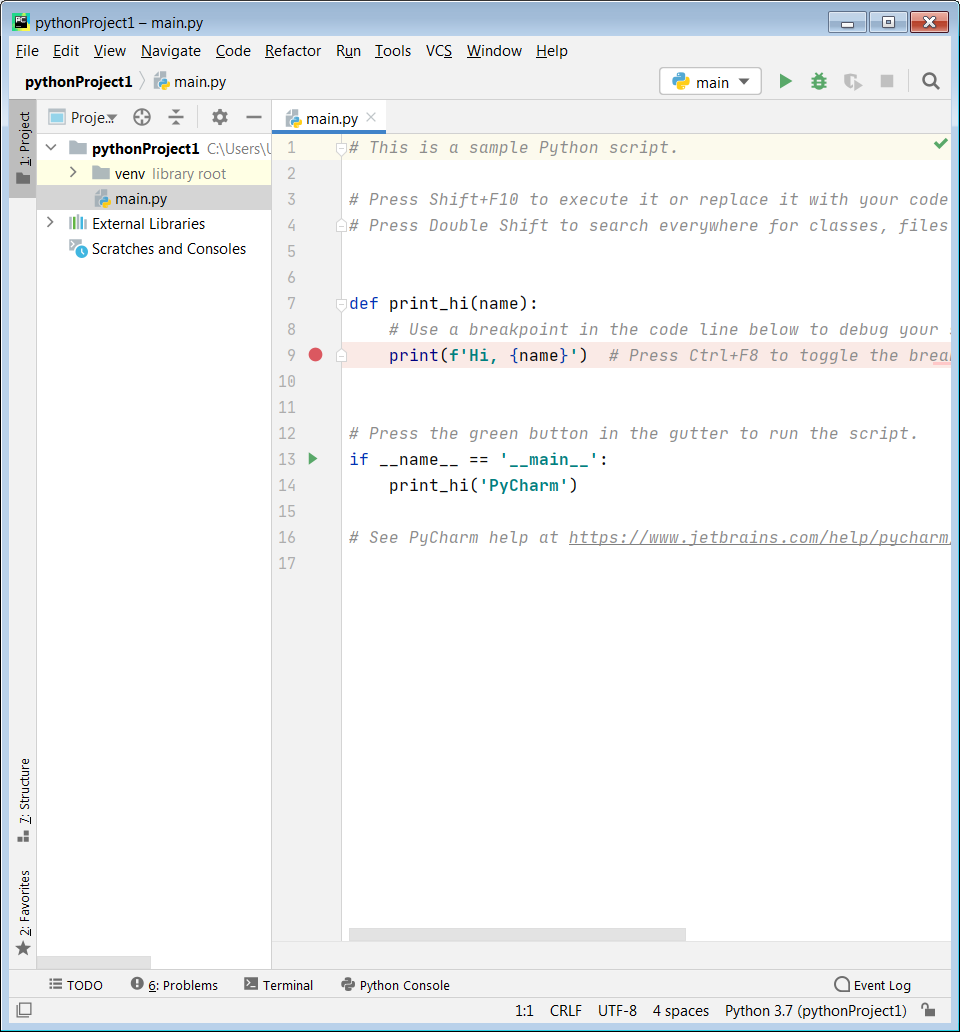


Рис.2 – Рабочее окно в IDE PyCharm

Принципы работа в среде PyCharm схожи с работой в среде IntelliJ IDEA

**Способы запуска программы**

* Команда меню Run / Run
* Shift+F10

После компилирования файлов с исходным кодом, среда отобразит результат выполнения программы в появившейся внизу вкладке «Run». Последняя строка содержит информацию о коде завершения приложения. 0 – означает, что приложение было выполнено без ошибок.

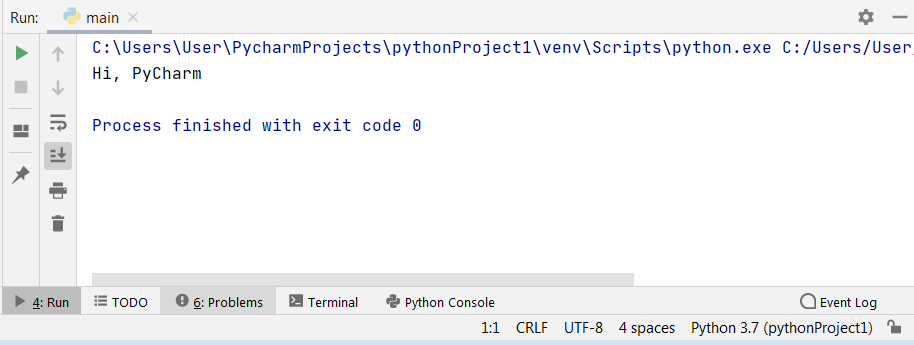


Рис. Окно вывода результатов выполнения программы

1. **Обработка математических выражений**

Большинство арифметических и математических стандартных методов описаны в модуле math. Некоторые из них приведены в презентации к лекции.

Для математических вычислений в Python имеются как встроенные, так и дополнительные функции и методы. Применить дополнительные математические функций можно после подключения модуля math:

import math

либо

from math import \*

В первом случае функции рассматриваются как методы объекта math и должны записываться так:

import math

print(math.sin(math.pi/4))

Во втором случае вызов функции может быть сделан в более

привычной для нас форме:

from math import \*

print(sin(pi/4))

print(sqrt(2)/2)

Вместе с тем, такой способ импорта может нарушить пространство имен программы, поскольку может возникнуть конфликт между именами переменных, которые использует программист и именами импортируемых функций. При импорте можно ограничиться только необходимыми функциями, например:

from math import (pi, sin, cos,

tan, log)

В этом примере демонстрируется способ импорта необходимых функций, и способ размещения инструкции на нескольких строках. Такие функции так же можно использовать в привычной для нас манере.

Набор функций и методов модуля math приводится в Таблице 1, см. Приложение 1. Больше информации о функциях модуля math можно получить из документации или в сети Интернет.

В тех случаях, когда в языке программирования нужная функция отсутствует, ее можно написать, либо вычислить, используя известные формулы. Например,

ctg(x) = 1/tan(x)= cos(x)/sin(x).

Имена переменных следует выбирать тщательнее и использовать либо принятые в математике или физике символы, либо фразы, отражающие назначение переменной.

Например, символ можно заменить на a, или alpha. Значение цвета можно хранить в переменной Color, а объема в переменной Volume или Capacity.

Обращайте внимание на цветовую раскраску переменной при наборе в редакторе IDLE. Она должна быть черной.

Для решения задания потребуется вводить и выводить данные. В нашем случае это числа целого или вещественного типа.

**Ввод данных**

Ввод данных можно выполнить с клавиатуры функцией input():

m = input([str])

При этом на экран будет выведена строка str, а переменная m получит значение строкового типа, введённое пользователем. Строковый тип может быть преобразован, например, к типу int или float, если введённое значение – число.

Для ввода нескольких значений можно воспользоваться методом split(), который позволяет разбить строку на подстроки (*split* –расщеплять).

Например, для ввода значений параметра и переменной x можно поступить так:

a, x = input('Введите данные (a, x): ').split()

a = float(a)

x = float(x)

Используемый разделитель указывается в качестве параметра метода split(). Если разделитель не указан, то им будет пробел. При вводе вещественного числа целая часть отделяется от десятичной дроби точкой. Если пользователь не ввел данные (просто нажал Enter) или вместо цифр и точки ввел недопустимые символы, например буквы, программа завершится аварийно на шаге приведения к типу – float().

Исключительная ситуация, которая при этом возникает, может быть обработана с помощью инструкции try.

В следующем примере пользователь должен ввести первое число целого типа, а второе – вещественного. Если ввод будет неправильным, то возникнет исключительная ситуация ValueError и управление программой будет передано в блок except. В этом блоке можно предусмотреть возможные ситуации и принять необходимое решение, например, заставить пользователя правильно ввести данные.

import sys, traceback  
while True:  
 a = x = **""** try:  
 a,x= (input(**'Введите [a: int, x: float]:'**).split())  
 a = int(a)  
 x = float(x)  
 break  
 except ValueError:  
 if a == **""** and x == **""**:  
 a = x = 0  
 continue  
 print(**"a - int, x - float. Пример: 3 4.5"**)  
print(a, x,)

В теле "вечного" цикла while True:, в блоке try,

инициализируются две переменные, а затем следует инструкция для ввода

данных. При ошибочном вводе числа, например, вместо целого –

вещественное, или вместо цифры – буква, возникнет исключительная

11

ситуация ValueError. Управление будет передано в модуль except, где, в условном операторе, проверяется, были ли введены данные. Если был нажат Enter, то переменные получат нулевое значение и управление будет передано инструкции, которая следует за циклом. Если ввод данных был сделан, то исключительная ситуация возникла при преобразовании типов данных. В этом случае выдается предупреждающее сообщение, и управление передается в начало цикла (ввод должен быть повторен).

*Замечание:* Если не выполнить инициализацию переменных перед инструкцией input(), то при возникновении исключительной ситуации (при вводе нажат только Enter), управление будет передано в модуль except, где возникнет новая исключительная ситуация в условном операторе if: переменная a не определена.

**Вывод данных**

Вывод данных на экран монитора может быть выполнен функцией print(). Эта функция позволяет выполнять форматированный вывод, как с использованием Си-подобного форматирования, так и с использованием форматной строки Python.

Следующие строки демонстрируют, как можно форматировать вывод.

for x in range(1,11):

print('%2d %3d %7.2f' % (x, x\*x, x\*x\*x))

print("{0:.2f} {1:.2f} {2:.4f}".format(a, x, y))

Буква в формате числа определяет тип выводимого числа. Так, d – это целый тип, f – вещественное число. Число в формате означает то число позиций, которое будет использовано для вывода числа. Для вещественного числа указывается, после точки, количество выводимых десятичных знаков.

Во второй строке использован Си-подобный формат, в котором формат числа начинается с процента "%". В этом формате аргументы отделяются от форматной части строки так же символом % – процент. В третьей строке используется форматная строка Python, в которой в форматной строке позиции для значений аргументов выделяются фигурными скобками.

Обратите внимание на то, что сами форматные строки начинаются и завершаются одиночной или двойной кавычкой. В Python допускаются оба вида кавычек для выделения строки. Важно только, что бы начало и конец были одинаковыми.

Так же следует понимать, что в промежутках между символами форматирования могут находиться и другие символы или слова:

print('x=%2d x^2=%3d x^3=%7.2f'

% (x, x\*x, x\*x\*x))

print("a={0:.2f} x={1:.2f} y={2:.4f}".format(a, x, y))

Использование форматных строк делает вывод данных более внятным. За более подробной информацией обращайтесь к учебникам или Интернет.

**Решение задания**

Вернемся к нашим примерам и запишем их, используя правила языка Python. Первое выражение примет вид:

1. y = tan(x\*\*2/2-1)\*\*2+2\*cos(x-pi/6)/(1/2+sin(a)\*\*2)

Второе выражение представим в виде двух:

2. tmp=log(3-cos(pi/4+2\*x),3+sin(x))/(1+tan(2\*x/pi)\*\*2)

y = pow(2, tmp)

**Описание алгоритма**

Для вычислений необходимо обеспечить ввод двух переменных x и a. Поскольку по условиям задачи их тип и точность представления не заданы, выберем для них вещественный тип (float). Для оптимизации записи выражения используем промежуточную переменную tmp.

1. Ввести значения a и x, преобразовать к типу float.

2. Вычислить выражение 1.

3. Вывести результат вычисления.

4. Вычислить значение переменной tmp;

5. Вычислить выражение 2.

6. Вывести результат вычисления.

**Листинг программы**

# -\*- coding: cp1251 -\*-

from math import \*

a = float(input('Введите параметр а: '))

x = float(input('Введите значение x: '))

y=tan(x\*\*2/2-1)\*\*2+(2\*cos(x-pi/6))/(1/2+sin(a)\*\*2)

print("{0:.2f} {1:.2f} {2:.4f}".format(a, x, y))

tmp=log(3-cos(pi/4+2\*x),

3+sin(x))/(1+tan(2\*x/pi)\*\*2)

y=pow(2,tmp)

print("{0:.2f} {1:.4f}".format(x, y))

**Результаты тестирования программы**



*Примечание:* Эта таблица оформлялась в текстовом редакторе вручную.\_\_

## Задание на лабораторную работу

Составить консольные приложения, реализующие приведенные задания на языке Python. В отчет включить листинги созданных программ и результаты выполненных расчетов (скриншоты выделенного окна – сочетание клавиш Alt+PrtSc).

**Задание 1**. Вычислить значения функции при указанном значении аргумента *x*. Значения *a***, b** и **c** задать в виде констант.

В отчет поместить результаты вычислений, полученные после выполнения программы.

1. Вычислить с использованием интерпретатора Python. Вывод результата должен включать информацию о значениях всех используемых переменных (с форматированием)

|  |  |  |
| --- | --- | --- |
| **Номер**  **варианта** | ***y* = *f*(*x*)** | **Исходные**  **данные** |
| 1 | image8 | image9 |
| 2 | image10 | image11 |
| 3 | image12 | image13 |
| 4 | image14 | image15 |
| 5 | image16 | image17 |
| 6 | image18 | image19 |
| 7 | image20 | image21 |
| 8 | image22 | image23 |
| 9 | image24 | image25 |
| 10 | image26 | image27 |
| 11 | image28 | image29 |
| 12 | image30 | image31 |
| 13 | image32 | image33 |
| 14 | image34 | image35 |
| 15 | image36 | image37 |
| 16 | image38 | image39 |
| 17 | image40 | image41 |
| 18 | image42 | image43 |
| 19 | image44 | image45 |
| 20 |  | image48 |  |
| 21 | image49 | image50 |
| 22 | image51 | image52 |
| 23 | image53 | image54 |
| 24 | image55 | image56 |
| 25 | image57 | image58 |

**Задание 2**. Написать программу в среде Python IDLE, в которой вводятся с клавиатуры все неизвестные вещественные числа и печатается значение выражения

|  |  |  |  |
| --- | --- | --- | --- |
| Номер  варианта | Выражение | Номер  варианта | Выражение |
| 1 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image4.png | 14 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image32.png |
| 2 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image6.png | 15 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image34.png |
| 3 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image8.png | 16 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image5.png |
| 4 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image10.png | 17 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image7.png |
| 5 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image12.png | 18 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image9.png |
| 6 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image14.png | 19 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image11.png |
| 7 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image16.png | 20 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image13.png |
| 8 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image18.png | 21 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image15.png |
| 9 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image20.png | 22 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image17.png |
| 10 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image22.png  C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image24.png | 23 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image19.png |
| 11 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image26.png | 24 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image21.png |
| 12 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image28.png | 25 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image23.png  C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image25.png |
| 13 | C:\Users\Admin\AppData\Local\Temp\FineReader12.00\media\image30.png |  | |

**Задание 3**. Написать программу в среде PyCharm, в которой вводятся с клавиатуры все неизвестные вещественные числа и печатается значение выражения

## 

## 

## 

## Контрольные вопросы

1. Подключение дополнительных модулей
2. Пример динамической типизации
3. Приведите варианты запуска программы на языке Python
4. Простейшая программа на Python
5. Правила отладки программ на языке Python
6. Организация консольного ввода/вывода