МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №1

Создание программы решающей квадратное уравнение

Выполнила: Студент группы

БВТ2207

Ковалевский Стас

ЦЕЛЬ РАБОТЫ: написать программу, которая сможет решить квадратное уравнение. Коэффициенты должны задаваться пользователем.

ХОД РАБОТЫ: данная лабораторная работа будет выполняться на языке программирования Python в одноимённой программе Python IDLE.

1. *Создание проекта*. Открываем программу для написания кода и создаём новый файл. После чего можно приступать к написанию кода.

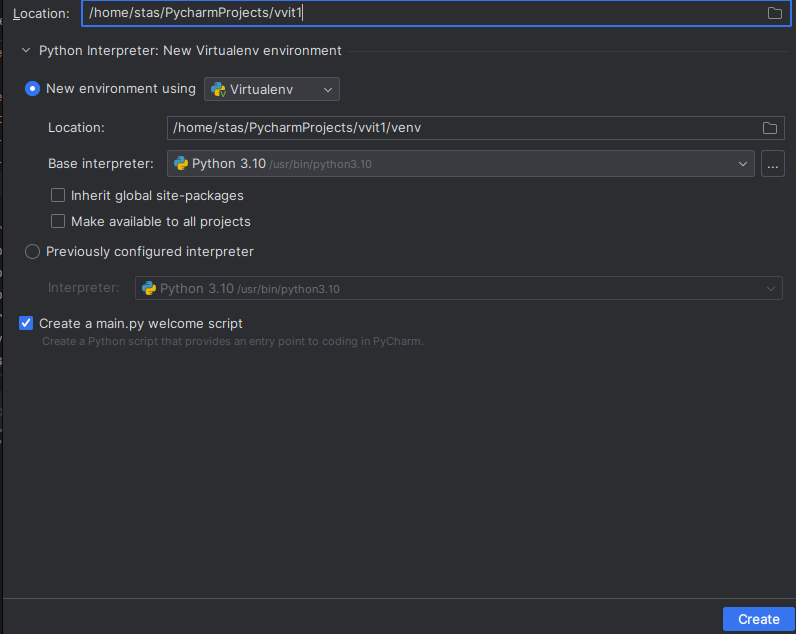


Рисунок 1 – создание нового файла

2*. Написание кода.*

*2.1.* Так как, исходя из условия, мы будем иметь дело с квадратным уравнением, то нам первым шагом необходимо подключить математическую библиотеку «**math**».

Для этого пропишем команду:***import math***

**

Рисунок 2 – подключение библиотеки «математика»

2.2 В задании сказано о том, что коэффициенты для квадратного уравнения должны задаваться пользователем. Сделаем это с помощью команды ***float(input())***и чтобы пользователь понимал, в какой момент ему нужно будет вводить коэффициенты, то для удобства и красоты пропишем обращение через ***print("Решение квадратного уравнения: ax^2 + bx + c = 0")***

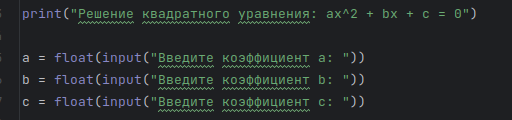
**

Рисунок 3 – ввод коэффициентов квадратного уравнения

2.3 Зададим переменную “D” – дискриминант квадратного уравнения и пропишем формулу для его нахождения.



Рисунок 4 – добавление переменной (дискриминанта) и формулы для её нахождения

2.4 Далее необходимо рассмотреть все случаи нахождения корней квадратного уравнения, а также случая когда их нет. Для этого будем использовать 3 оператора**: *if, elif, else****.*

***IF*** *=* Если (то есть: «если» выполняется какое-то условие, то…

С оператором***if***будет выполняться условие, при котором квадратное уравнения будет иметь 2 корня. В словаре Python команда нахождения корня имеет вид ***sqrt*,** а так как мы импортируем её из библиотеки, то нахождение корня примет вид ***math.sqrt***

Для нахождения корней также пропишем формулы, соответствующие правилу их нахождения при дискриминанте «больше нуля»

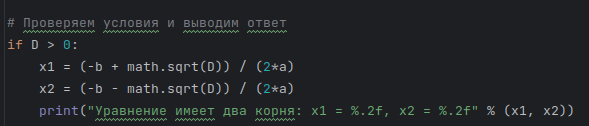
**

Рисунок 5 – оператор ***if*** и формулы для нахождения корней

Оператор ***elif***. По факту этот оператор – это тот же **if**. Однако, всё-таки его отличие заключается в том, что с помощью оператора **elif** будет выполняться проверка на уже другое условие, с учётом результата оператора if: то есть если оператор **if** даст результат, то **elif** задействован не будет. Если же наоборот, то оператор **elif** начнёт свою работу.

Нельзя написать в данном случае вместо **elif** ещё один оператор **if** по той причине, что тогда второй **if** (для второго условия) будет выполняться в любом случае вне зависимости от результата первого оператора **if**. Поэтому мы обязаны написать именно оператор **elif**.

Использовать оператор ***elif*** будем для нахождения единственного корня квадратного уравнения - когда дискриминант равен нулю.

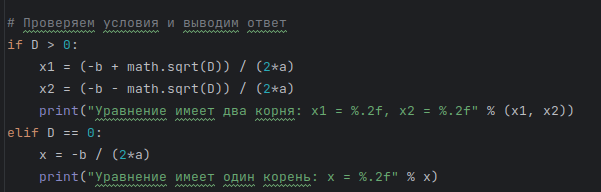


Рисунок 6 – оператор ***elif*** и нахождение единственного корня

Оператор ***else*** (***else*** = если не…): суть этого оператора предельно проста – если не выполняется ни одно из условий ранее (в нашем случае это ***if*** и ***elif***), то тогда будет выполняться обязательно третье и оно же единственное условие. Если же хоть один из операторов ранее всё-таки сработал и дал результат (в нашем случае нашёл хотя бы один корень квадратного уравнения), то тогда оператор ***else*** не будет включён в работу.

Для оператора ***else*** остаётся третье условие – дискриминант меньше нуля, корней нет. Причём этот оператор работает таким образом, что для него не обязательно прописывать это условие, так как логика нашего кода это «если так, или если так, ИНАЧЕ так». Поэтому после ***else*** можно сразу писать ***print***, а конкретнее ***print(“корней нет”)***

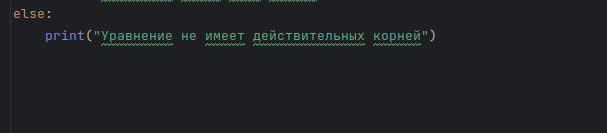


Рисунок 7 – добавление оператора ***else*** и вывод ответа пользователю

Код готов, теперь необходимо его запустить и проверить работоспособность.

Сделать это с помощью горячей клавиши f5 или же по направлению ***run*** => ***run*** ***main.py.***

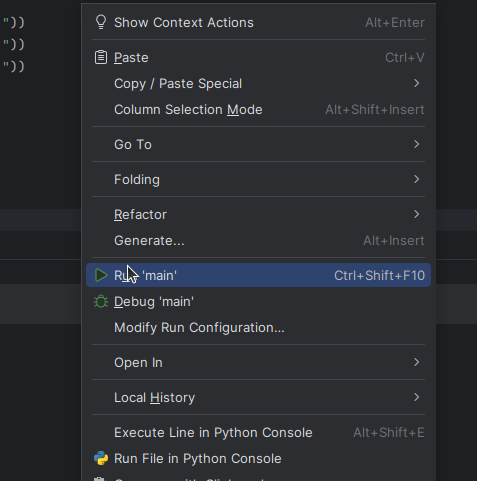
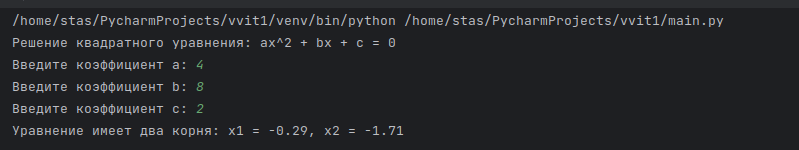


Рисунок 8 – запуск программы

3. Проверка работоспособности программы.



Как итог – видим, что программа работает исправно.

ВЫВОД: в ходе этой лабораторной работы я освоил навыки по работе с операторами (if, elif, else), смог написать программу для нахождения корней квадратного уравнения с помощью подключения соответствующей библиотеки.

•Ссылка на гит

https://github.com/nihaobrat/vvit1-8