**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Смыслов Дмитрий  Олегович |  | Гапанюк Юрий Евгеньевич |
|  |  |  |
| Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
|  |  |  |

Москва, 2021 г.

**Описание задания**

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2.
4. Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

**Текст программы**

import sys

import math

def is\_float(arg):

try:

float(arg)

return True

except ValueError:

return False

def get\_coefficient(index, prompt):

try:

coefficient\_str = sys.argv[index]

while not is\_float(coefficient\_str):

coefficient\_str = input(prompt)

except:

coefficient\_str = input(prompt)

while not is\_float(coefficient\_str):

coefficient\_str = input(prompt)

return float(coefficient\_str)

def get\_quadratic\_roots(a, b, c):

result = []

if a == 0:

if b == 0:

if c == 0:

result.append('R')

else:

if c == 0:

result.append(0.0)

else:

result.append(-c / b)

else:

D = b \* b - 4 \* a \* c

if D == 0.0:

root = -b / (2.0 \* a)

result.append(root)

elif D > 0.0:

sqD = math.sqrt(D)

root1 = (-b + sqD) / (2.0 \* a)

root2 = (-b - sqD) / (2.0 \* a)

result.append(root1)

result.append(root2)

return result

def get\_biquadratic\_roots(a, b, c):

roots = []

sub\_roots = get\_quadratic\_roots(a, b, c)

if len(sub\_roots) != 0 and sub\_roots[0] == 'R':

roots.append('R')

else:

for sub\_root in sub\_roots:

for root in get\_quadratic\_roots(1, 0, -sub\_root):

roots.append(root)

return roots

def main():

a = get\_coefficient(1, 'Введите коэффициент A: ')

b = get\_coefficient(2, 'Введите коэффициент B: ')

c = get\_coefficient(3, 'Введите коэффициент C: ')

roots = get\_biquadratic\_roots(a, b, c)

len\_roots = len(roots)

if len\_roots == 0:

print('Нет корней')

elif roots[0] == 'R':

print('Корень - любое число')

else:

print('Корни', end=': ')

for root in roots:

print(root, end='; ')

print()

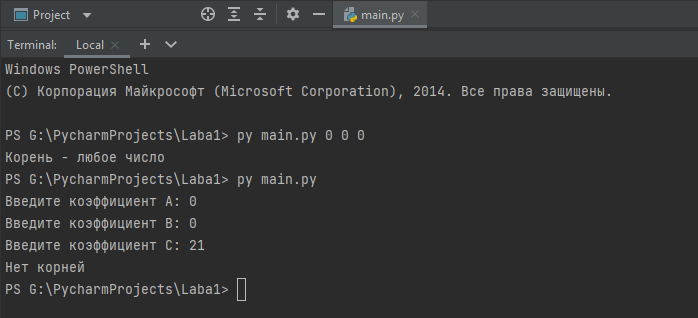
if \_\_name\_\_ == "\_\_main\_\_":

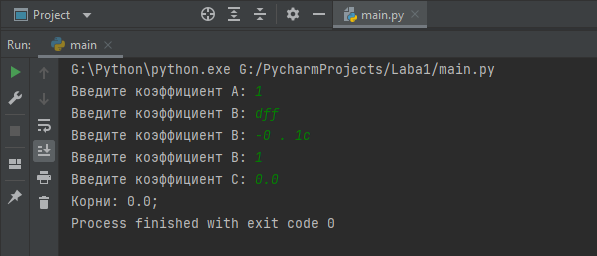
main()

**Экранные формы с примерами выполнения программы**

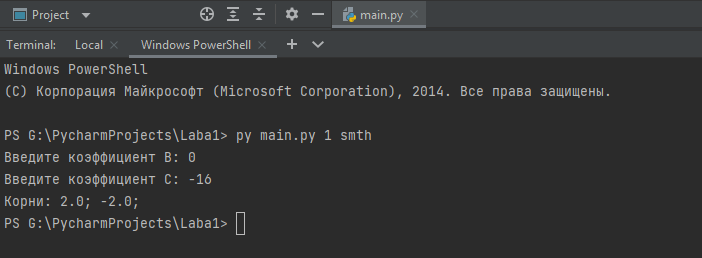
Программа может обрабатывать случаи, когда у биквадратного уравнения одно, два, три или четыре решения. В случае, когда у уравнения нет корней или их бесконечно много, выводится на консоль соответствующее сообщение.

Примеры выполнения программы с параметрами, заданными в командной строке и введёнными с клавиатуры, когда корней нет или их бесконечно много

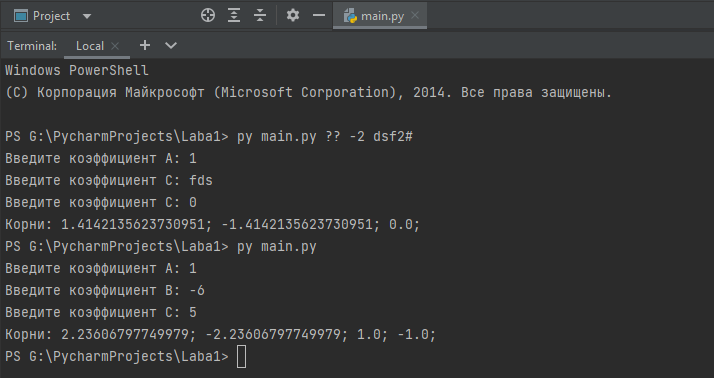


Пример выполнения программы при некорректном вводе коэффициентов с клавиатуры в случае, когда уравнение имеет единственный корень

Пример выполнения программы при некорректном задании коэффициентов в командной строке в случае, когда уравнение имеет два корня



Примеры выполнения программы в случае, когда уравнение имеет три или четыре корня



Таким образом проверили работу программы в случае разного числа корней уравнения и при различных входных данных, как некорректных, так и правильных, заданных с клавиатуры или в консоли.