|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация:  Гапанюк Ю. Е.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Отчет по лабораторной работе №3 по курсу**

**«Парадигмы и конструкции языков программирования»**

#### Тема работы: "Изучение языка C#"

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: | Агапова Анна Денисовна |
| студент группы  ИУ5Ц-52Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Гапанюк Ю.Е. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ 2024

**СОДЕРЖАНИЕ**

[Цель лабораторной работы 3](#_Toc178668626)

[Практическое задание 3](#_Toc178668627)

[Листинг программы 4](#_Toc178668628)

[Результат работы программы 6](#_Toc178668629)

[Сравнения программ на C# и Python 16](#_Toc178668630)

## Цель лабораторной работы

Получение практических навыков по C#

## Практическое задание

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

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 ( [вариант задания параметров приведен в конце файла с примером кода](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

## Листинг программы

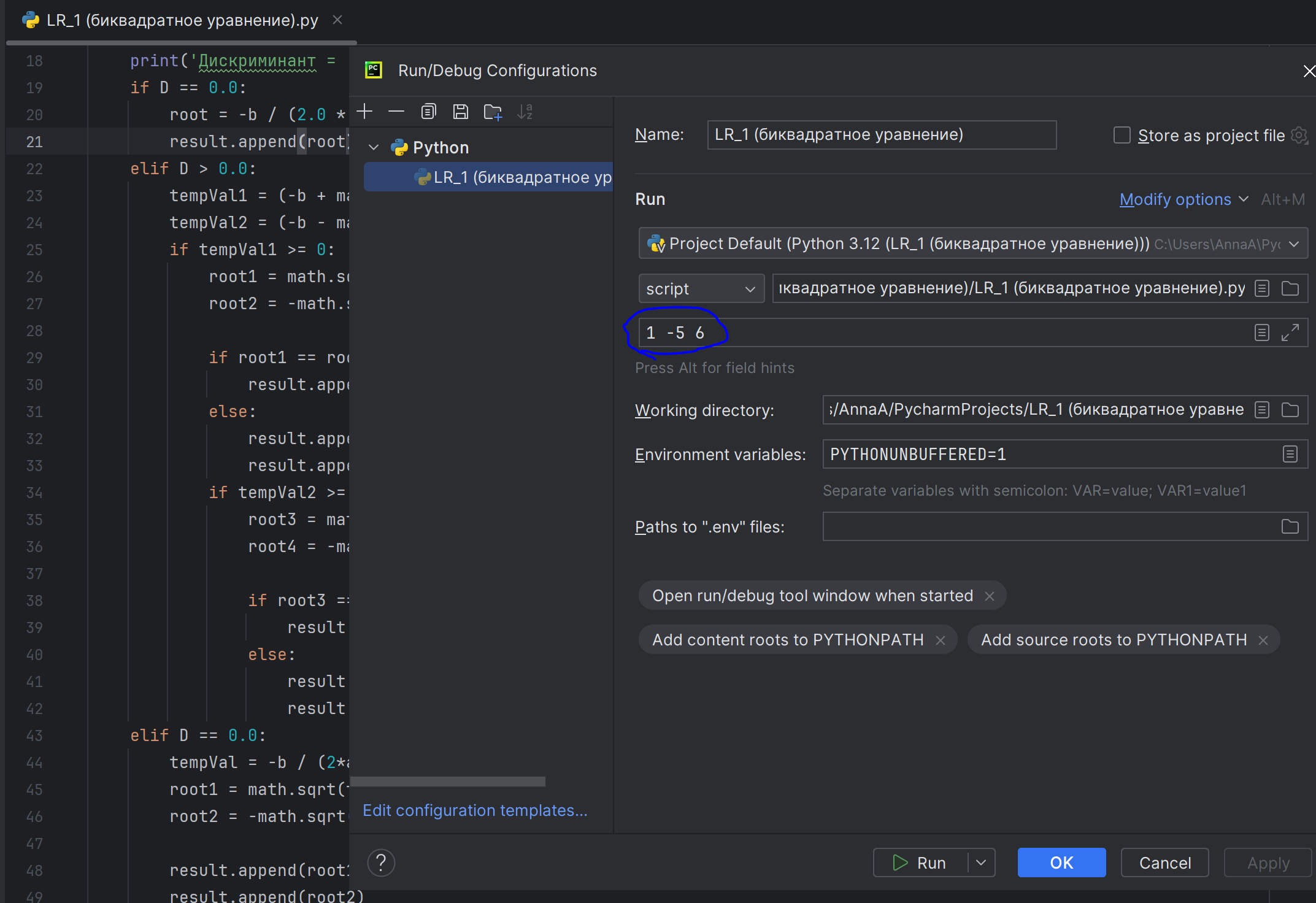
import sys  
import math  
  
  
def get\_K(index, prompt):  
 while True:  
 try:  
 coef\_str = sys.argv[index]  
 coef = float(coef\_str)  
 return coef  
 except (IndexError, ValueError):  
 print(prompt)  
 coef\_str = input()  
 try:  
 coef = float(coef\_str)  
 return coef  
 except ValueError:  
 print("Некорректное значение. Попробуйте снова.")  
  
def out\_red(text):  
 print("\033[31m{}".format(text))

def get\_roots(a, b, c):  
 result = []  
  
 D = b \* b - 4 \* a \* c  
 print('Дискриминант = ', D, '\n')

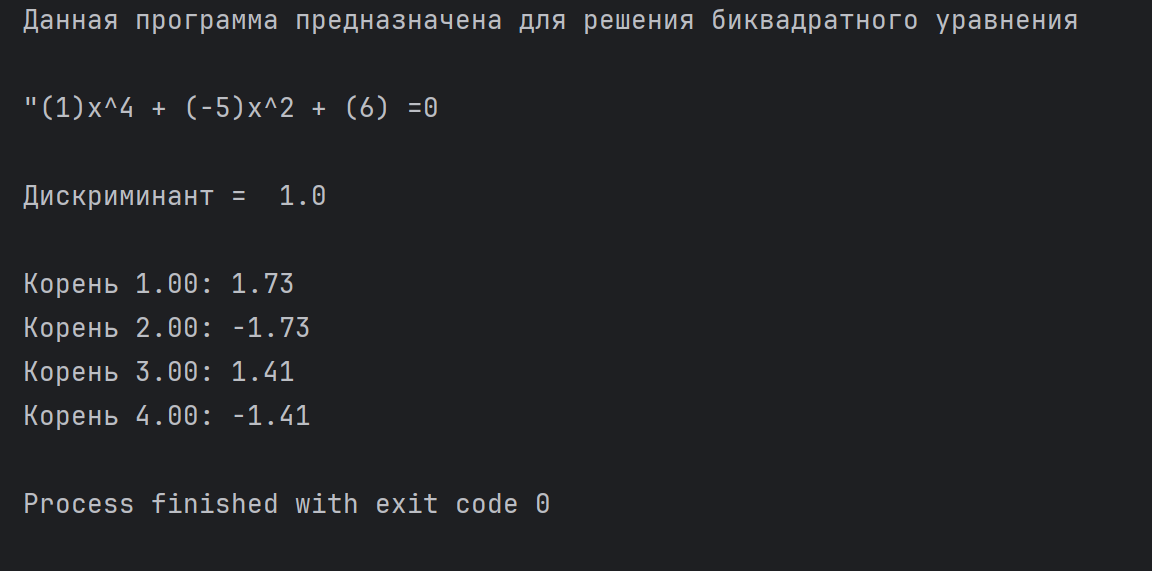
if D < 0:  
 out\_red('Корней нет')  
 elif D > 0.0:  
 tempVal1 = (-b + math.sqrt(D)) / (2 \* a)  
 tempVal2 = (-b - math.sqrt(D)) / (2 \* a)  
  
 if tempVal1 >= 0:  
 root1 = math.sqrt(tempVal1)  
 root2 = -math.sqrt(tempVal1)  
 if root1 == -0:  
 result.append(-root1)  
 result.append(root1)  
 result.append(root2)  
  
 if tempVal2 >= 0:  
 root3 = math.sqrt(tempVal2)  
 root4 = -math.sqrt(tempVal2)  
 result.append(root3)  
 result.append(root4)  
  
 elif D == 0.0:  
 tempVal = -b / (2 \* a)  
 if tempVal >= 0:  
 root1 = math.sqrt(tempVal)  
 root2 = -math.sqrt(tempVal)  
 result.append(root1)  
 result.append(root2)  
 result = set(result)  
 return result  
  
  
def main():  
 *'''  
 Основная функция  
 '''* print("Данная программа предназначена для решения биквадратного уравнения\n")  
 a = get\_K(1, 'Введите коэффициент A:')  
 b = get\_K(2, 'Введите коэффициент B:')  
 c = get\_K(3, 'Введите коэффициент C:')  
 print('"({:.0f})x^4 + ({:.0f})x^2 + ({:.0f}) = 0\n'.format(a, b, c))  
  
 # Вычисление корней  
 roots = get\_roots(a, b, c)  
  
 # Вывод корней  
 for k, root in enumerate(roots, start=1):  
 print('Корень {:.0f}: {:.2f}'.format(k, root))  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

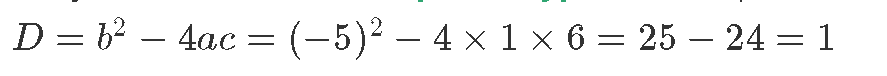
## Результат работы программы

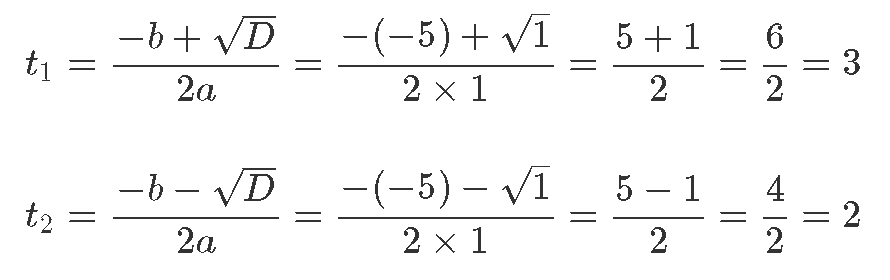
При задании аргументов командной сторки



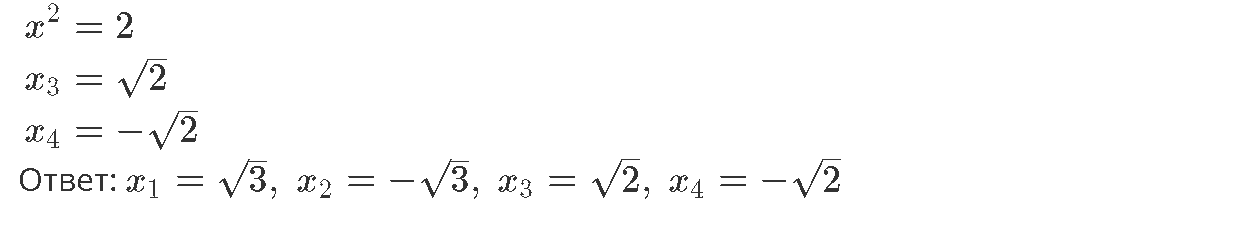




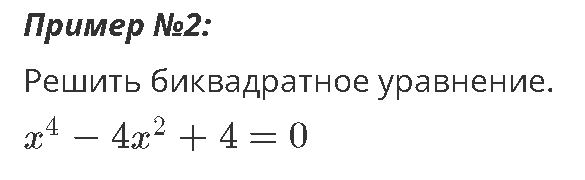


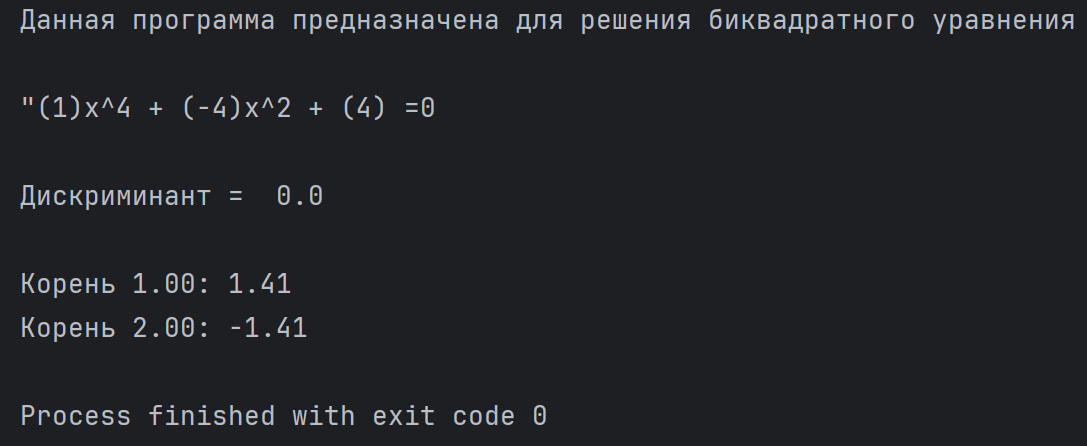


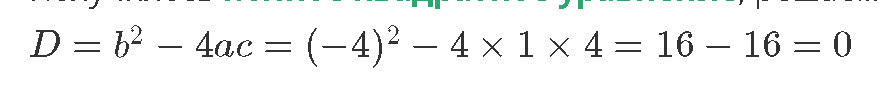
Вернемся к старой переменной:



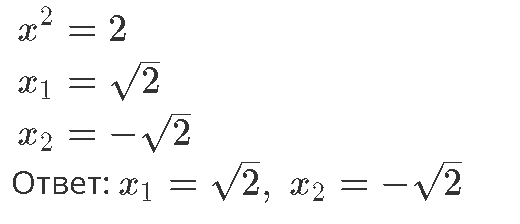
 



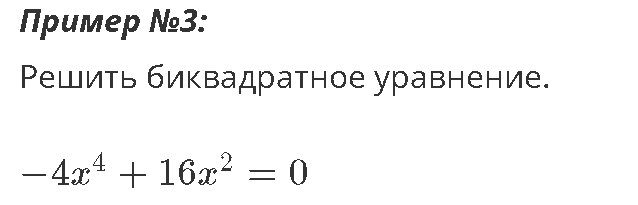


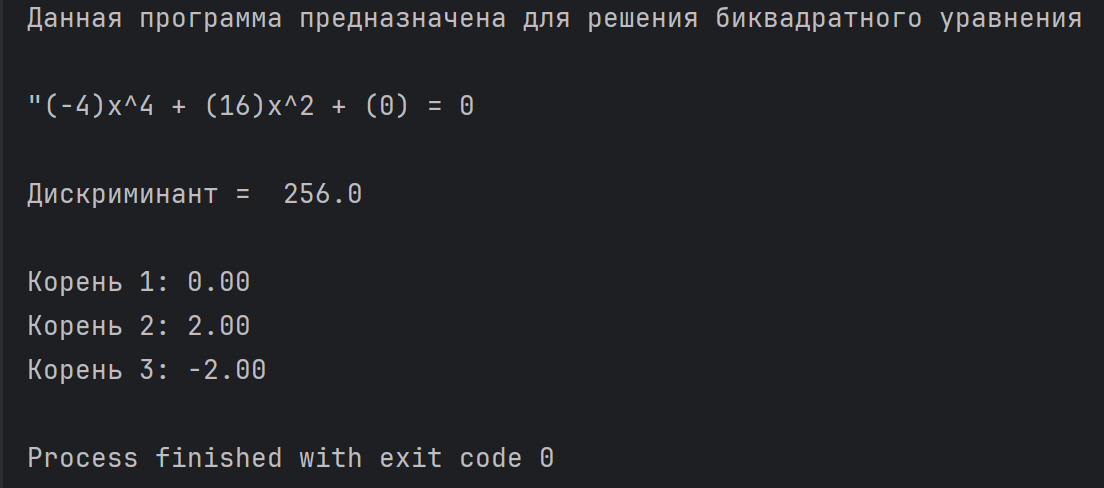


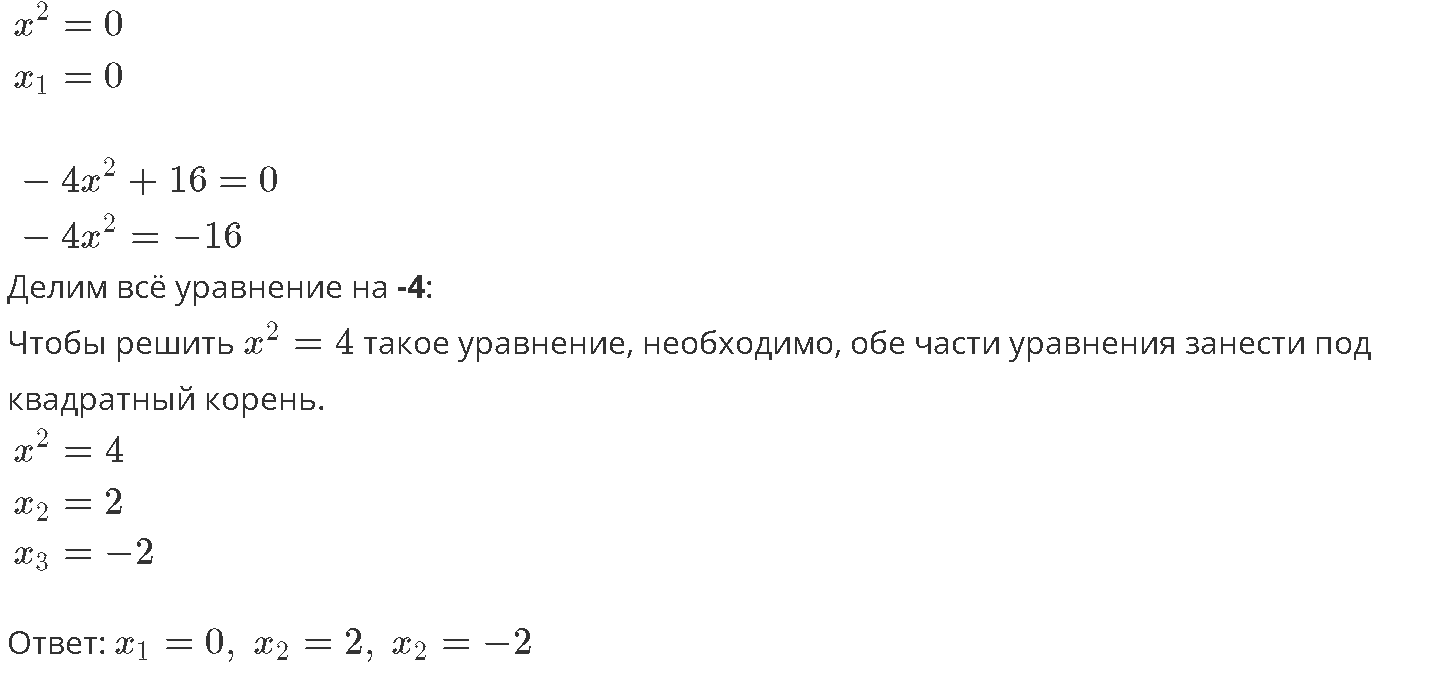
Вернемся к старой переменной:

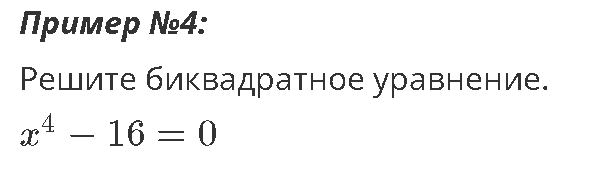


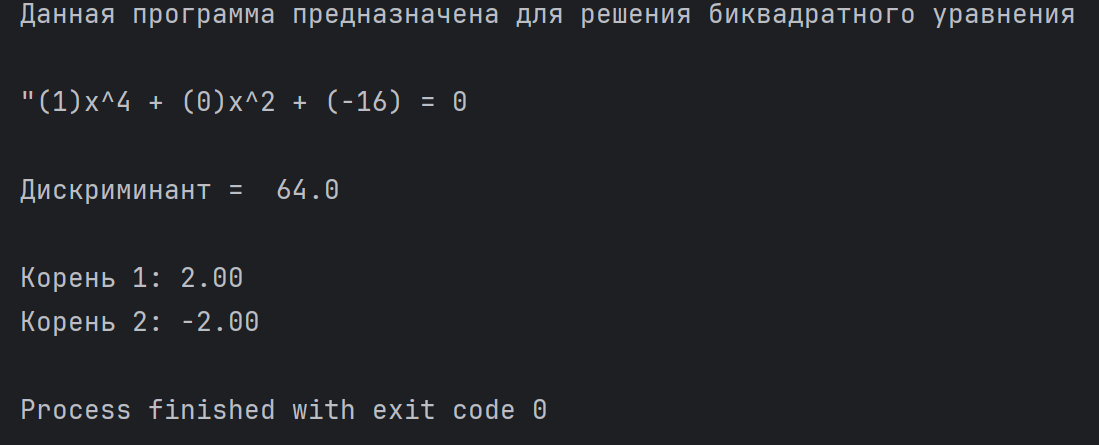


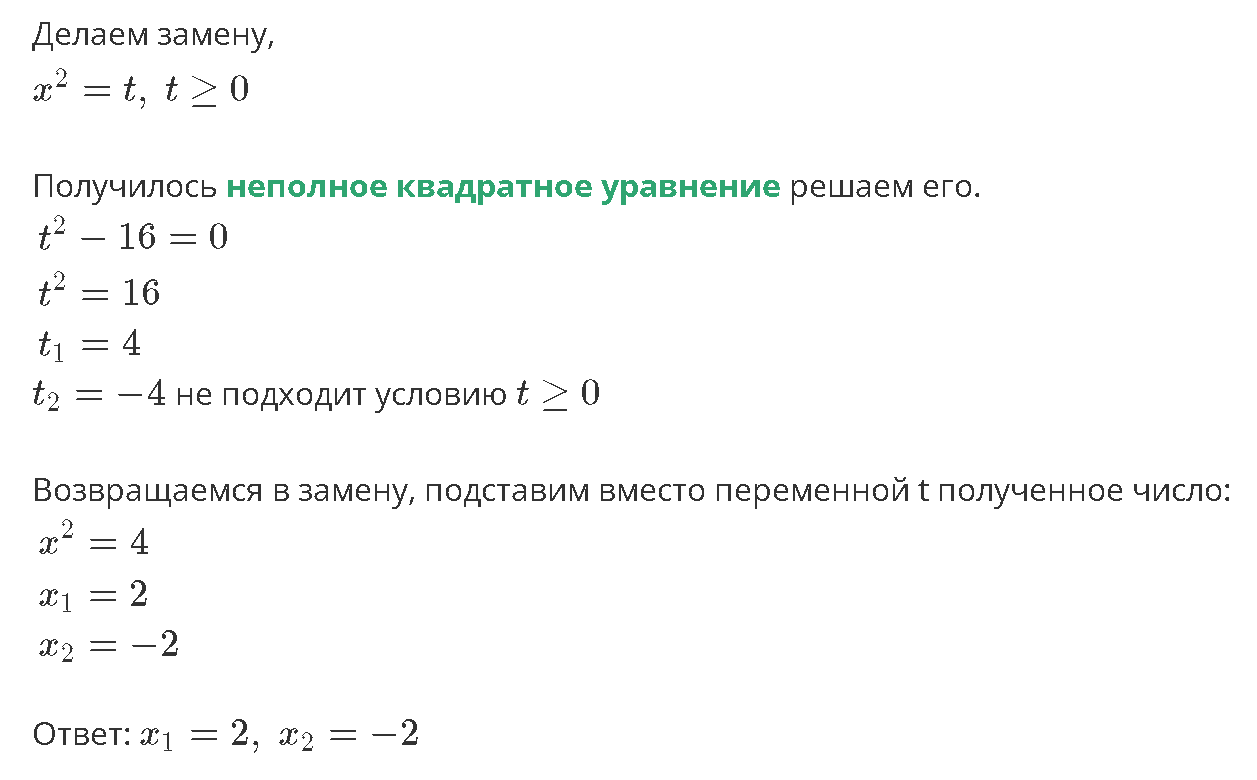


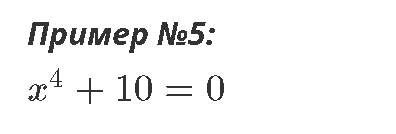


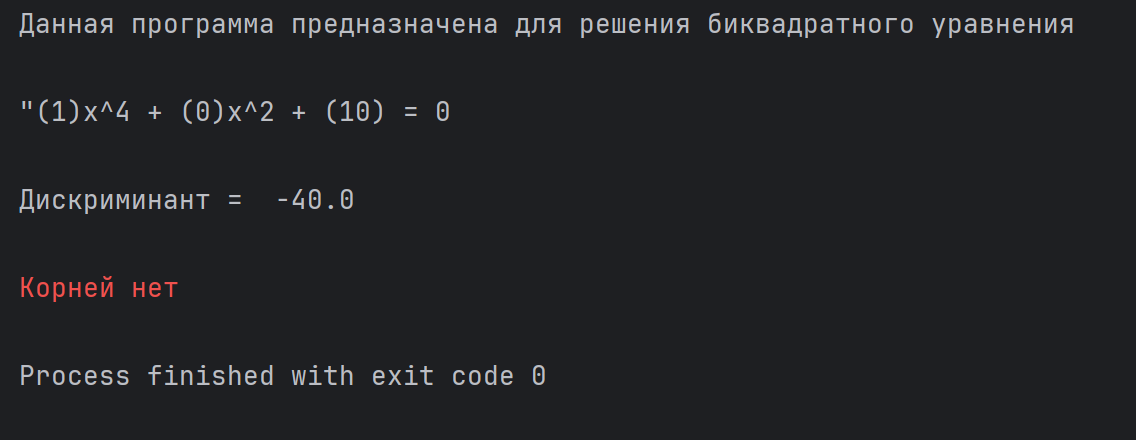


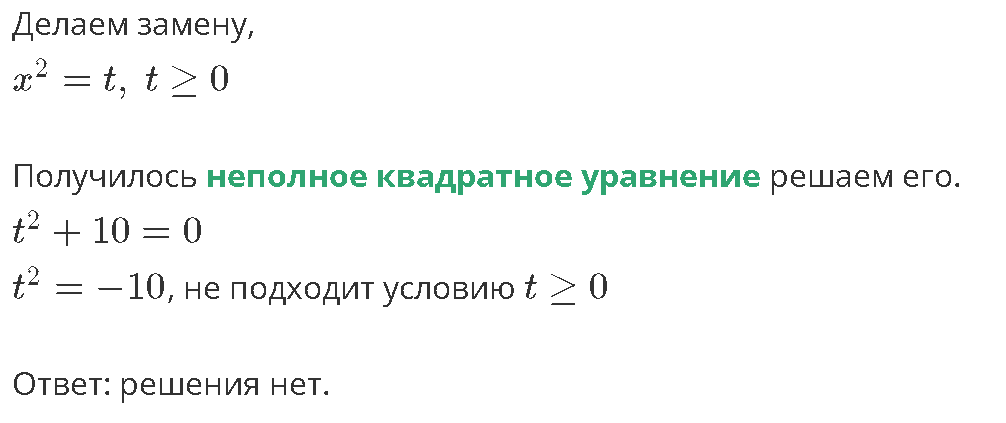






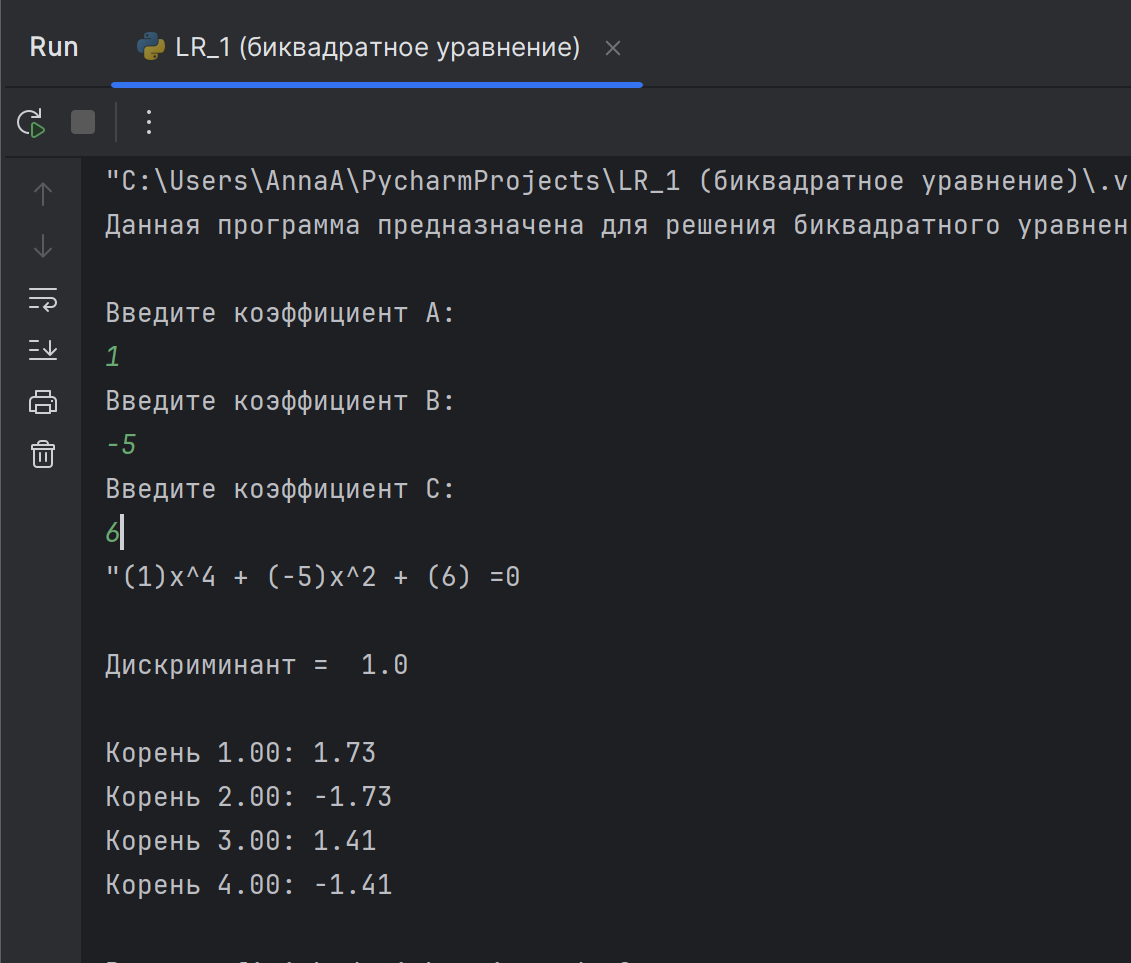






Проверка пункта (6):

Если они (параметры командной строки) не указаны, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.



**Дополнительное задание 1** (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

**с применением объектно-ориентированной парадигмы**

import sys  
import math  
  
class BQ\_Equation:  
 def \_\_init\_\_(self, a, b, c):  
 self.a = a  
 self.b = b  
 self.c = c  
 def get\_roots(self):  
 result = []  
 D = self.b \*\* 2 - 4 \* self.a \* self.c  
 print('Дискриминант = ', D, '\n')  
  
 if D > 0.0:  
 tempVal1 = (-self.b + math.sqrt(D)) / (2 \* self.a)  
 tempVal2 = (-self.b - math.sqrt(D)) / (2 \* self.a)  
  
 if tempVal1 >= 0:  
 result.append(math.sqrt(tempVal1))  
 result.append(-math.sqrt(tempVal1))  
  
 if tempVal2 >= 0:  
 result.append(math.sqrt(tempVal2))  
 result.append(-math.sqrt(tempVal2))  
  
 elif D == 0.0:  
 tempVal = -self.b / (2 \* self.a)  
 if tempVal >= 0:  
 result.append(math.sqrt(tempVal))  
 result.append(-math.sqrt(tempVal))  
  
 return result  
  
 @staticmethod  
 def get\_K(index, prompt):  
 while True:  
 try:  
 coef\_str = sys.argv[index]  
 coef = float(coef\_str)  
 return coef  
 except (IndexError, ValueError):  
 print(prompt)  
 coef\_str = input()  
 try:  
 coef = float(coef\_str)  
 return coef  
 except ValueError:  
 print("Некорректное значение. Попробуйте снова.")  
  
def main():  
 print("Данная программа предназначена для решения биквадратного уравнения\n")  
 a = BQ\_Equation.get\_K(1, 'Введите коэффициент A:')  
 b = BQ\_Equation.get\_K(2, 'Введите коэффициент B:')  
 c = BQ\_Equation.get\_K(3, 'Введите коэффициент C:')  
  
 equation = BQ\_Equation(a, b, c)  
 print('({:.0f})x^4 + ({:.0f})x^2 + ({:.0f}) = 0\n'.format(equation.a, equation.b, equation.c))  
  
 # Вычисление корней  
 roots = equation.get\_roots()  
  
 # Вывод корней  
 for k, root in enumerate(roots, start=1):  
 print('Корень {:.0f}: {:.2f}'.format(k, root))  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Дополнительное задание 2 (\*).** Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Аналогичная программа на C#**

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LR\_1

{

class Program

{

static void Main(string[] args)

{

//создаем объект

Stopwatch stopwatch = new Stopwatch();

//засекаем время начала операции

stopwatch.Start();

Console.WriteLine("Данная программа предназначена для решения биквадратного уравнения\n");

// Предполагаем, что коэффициенты A, B, C инициализируются нулевыми значениями

double A = 0, B = 0, C = 0;

// Пытаемся получить коэффициенты из параметров командной строки

if (args.Length == 3)

{

//присваивем коэффициентам A, B, C значения из аргументов командной строки

double.TryParse(args[0], out A);

double.TryParse(args[1], out B);

double.TryParse(args[2], out C);

Console.WriteLine("А В С получены из командной строки\n");

Console.WriteLine("(" + args[0] + ")x^4 + (" + args[1] + ")x^2 + (" + args[2] + ") =0\n");

//если таковые заданы (проверим и если нет - попросим пользователя ввести их вручную)

if (!double.TryParse(args[0], out A) || !double.TryParse(args[1], out B) || !double.TryParse(args[2], out C))

{

Console.WriteLine("Необходимо ввести вещественное число. Вводите коэффициенты заново.\n");

}

}

else //если аргументы не заданы, их вводит пользователь

{

A = ReadDouble("Введите коэффициент A: ");

B = ReadDouble("Введите коэффициент B: ");

C = ReadDouble("Введите коэффициент C: ");

}

double discriminant = (B \* B) - (4 \* A \* C); //вычислим дискриминант

Console.WriteLine("Дискриминант = " + discriminant + "\n");

// Рассмотрим возможные значения дискриминанта

if (discriminant > 0)

{

double TempVal1 = (-B + Math.Sqrt(discriminant)) / (2 \* A);

double TempVal2 = (-B - Math.Sqrt(discriminant)) / (2 \* A);

Console.ForegroundColor = ConsoleColor.Green;

if (TempVal1 >= 0)

{

double root1 = Math.Sqrt(TempVal1);

double root2 = -Math.Sqrt(TempVal1);

if (root1 == 0)

{

Console.WriteLine("Корень 1: {0:F2}", -root1);

}

else if (root1 == root2)

{

Console.WriteLine("Корень 1: {0:F2}", root1);

}

else

{

Console.WriteLine("Корень 1: {0:F2}", root1);

Console.WriteLine("Корень 2: {0:F2}", root2);

}

}

if (TempVal2 >= 0)

{

double root3 = Math.Sqrt(TempVal2);

double root4 = -Math.Sqrt(TempVal2);

if (root3 == -0)

{

Console.WriteLine("Корень 4: {0:F2}", -root3);

}

else if (root3 == root4)

{

Console.WriteLine("Корень 3: {0:F2}", root3);

}

else

{

Console.WriteLine("Корень 3: {0:F2}", root3);

Console.WriteLine("Корень 4: {0:F2}", root4);

}

}

Console.ResetColor();

}

else if (discriminant == 0)

{

double TempVal = (-B) / (2 \* A);

ProcessSingleRoot(TempVal);

}

else

{

// Нет корней => выводим сообщение красным цветом

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Корней нет\n");

Console.ResetColor();

}

//останавливаем счётчик

stopwatch.Stop();

//смотрим сколько миллисекунд было затрачено на выполнение

Console.WriteLine("Время выполнения программы (мс) = {0:F2}", stopwatch.ElapsedMilliseconds);

}

/// <summary>

/// Ввод вещественного числа с проверкой корректности ввода

/// </summary>

/// <param name="message">Подсказка при вводе</param>

static double ReadDouble(string message)

{

string resultString;

double resultDouble;

bool flag;

do

{

Console.Write(message);

resultString = Console.ReadLine();

flag = double.TryParse(resultString, out resultDouble);

if (!flag)

{

Console.WriteLine("Необходимо ввести вещественное число");

}

}

while (!flag);

return resultDouble;

}

static void ProcessSingleRoot(double TempVal)

{

Console.ForegroundColor = ConsoleColor.Green;

if (TempVal >= 0)

{

double root1 = Math.Sqrt(TempVal);

double root2 = -Math.Sqrt(TempVal);

Console.WriteLine("Корень 1: {0:F2}", root1);

Console.WriteLine("Корень 2: {0:F2}\n", root2);

}

Console.ResetColor();

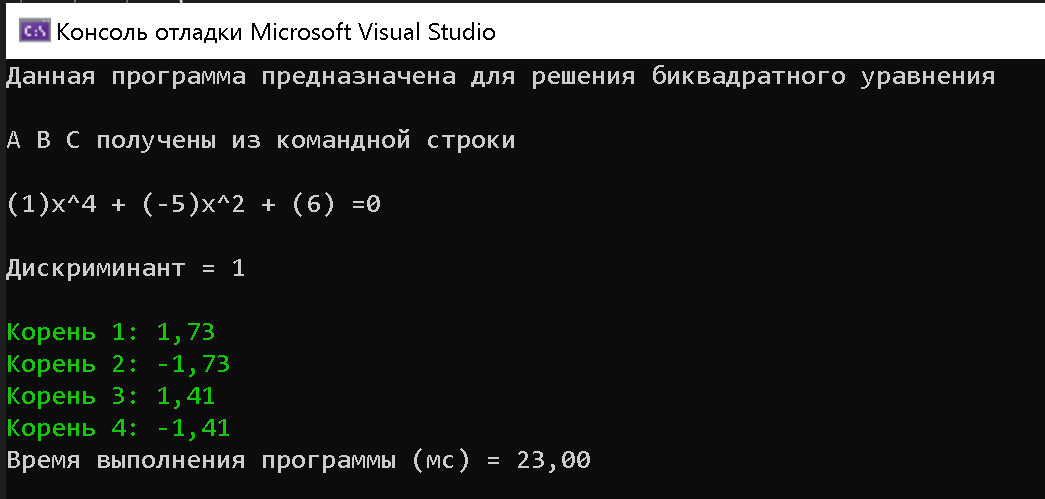
}

}

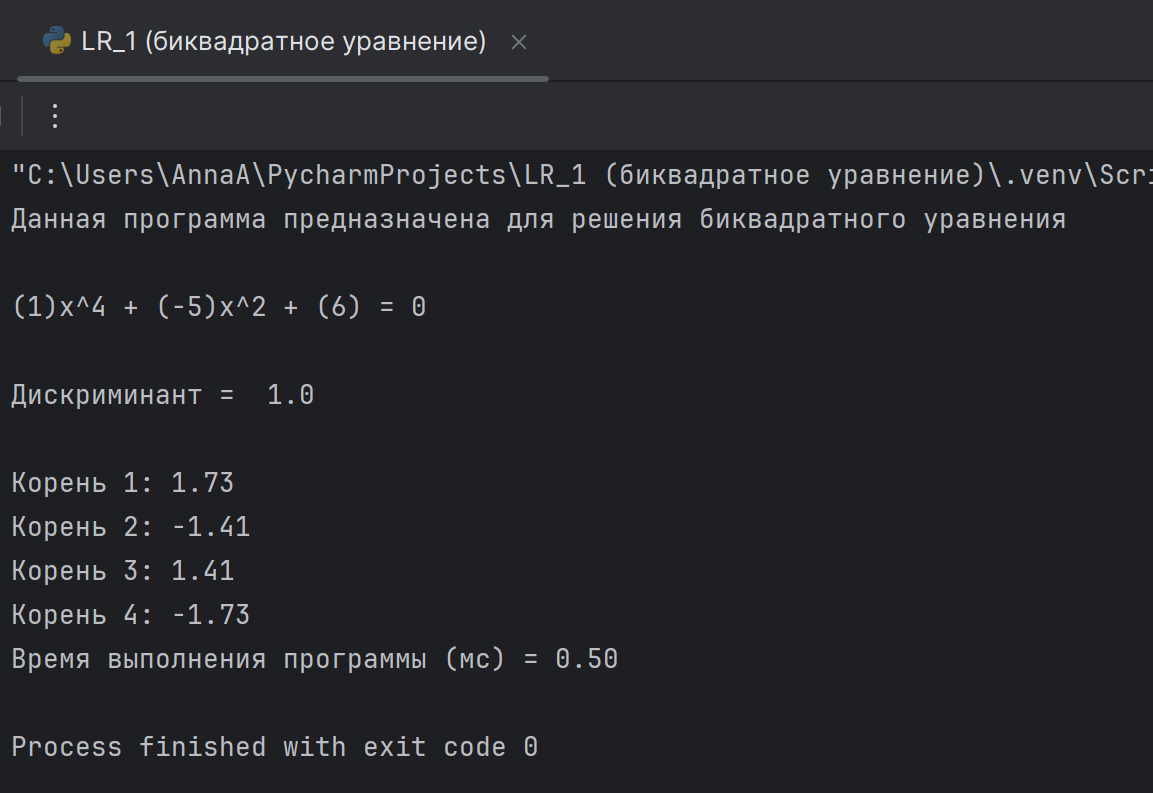
}

## Сравнения программ на C# и Python

Код на C#:



Код на **Python**:



Вывод: данную задачу лучше реализовывать на Питоне