**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Лабораторная работа №1

Выполнил:

Студент ИУ5-34Б

Флоринский В. А.

Подпись и дата:

Проверил:

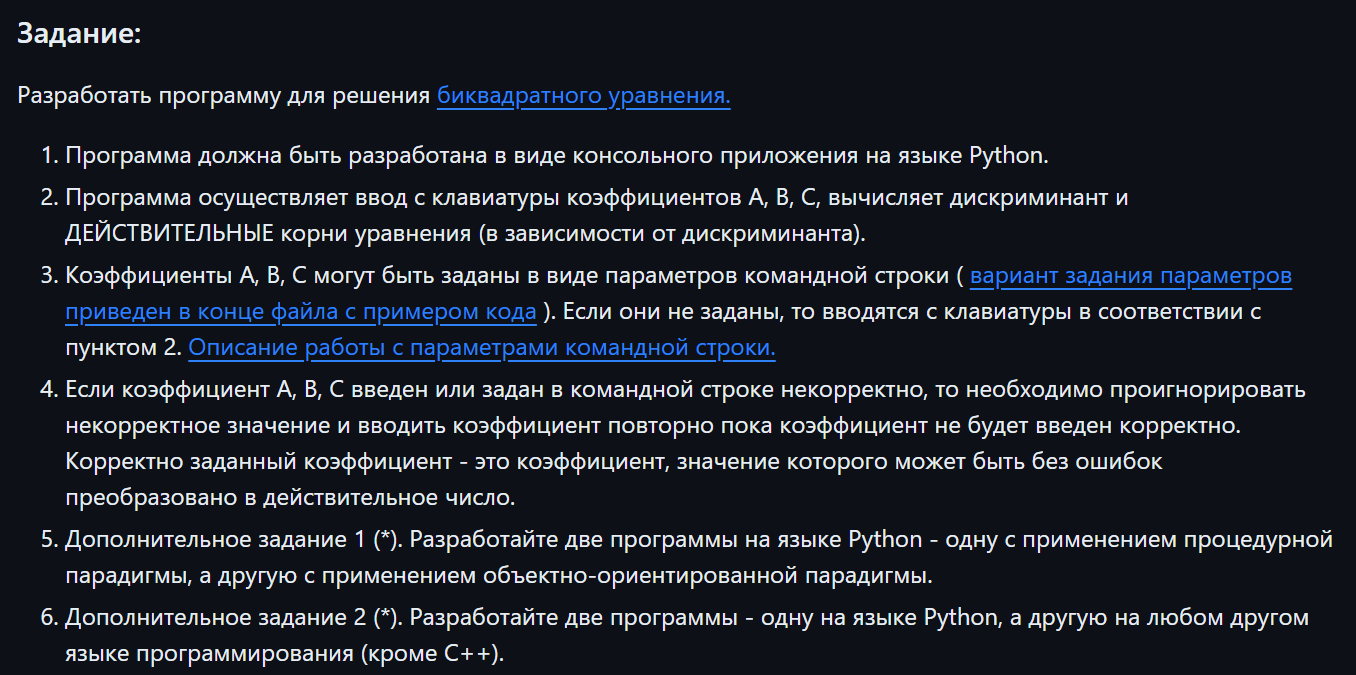
Преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

**Задание.**

****

**Текст программ.**

Данная ЛР разработана двумя путями: с помощью ООП и с помощью процедурной парадигмы. LR1OOP.py – решение задачи с помощью ООП, LR1P.py – с помощью процедурной парадигмы.

**LR1OOP.py:**

import sys

class Equation:

    def \_\_init\_\_(self, a = 1, b = 1, c = 1, arr = []):

        self.a = a

        self.b = b

        self.c = c

        self.arr = arr

    def get\_coef(self):

        ans = 0

        try:

            arr = [float(i) for i in sys.argv[1:]]

            if (len(arr) == 3 and arr[0] != 0):

                print("Взяты параметры из cmd")

                #return arr

                self.a = arr[0]

                self.b = arr[1]

                self.c = arr[2]

                ans = 1

            else:

                print("Ошибка при взятии аргументов из cmd")

        except:

            print("Ошибка при взятии аргументов из cmd")

        while ans == 0:

            try:

                print ("Введите 3 действительных числа через пробел (A, B, C)")

                arr = list(map(float, input().split())) #[float(i) for i in input().split()]

                if (len(arr) == 3 and arr[0] != 0):

                    self.a = arr[0]

                    self.b = arr[1]

                    self.c = arr[2]

                    ans = 1

                else:

                    print("Error")

                    ans = 0

            except:

                print("Error")

                ans = 0

    def solve(self):

        solutions = []

        d = float(self.b\*\*2 - 4 \* self.a \* self.c)

        if (d == 0):

            x = -self.b / (2 \* self.a)

            solutions.append(x)

        elif (d > 0):

            x1 = (-self.b + d\*\*0.5) / (2 \* self.a)

            x2 = (-self.b - d\*\*0.5) / (2 \* self.a)

            solutions.append(x1)

            solutions.append(x2)

        self.arr = solutions

    def see\_eq(self):

        print(f"{self.a}x^2", end="")

        if (self.b > 0): print(f"+{self.b}x", end="")

        elif (self.b < 0): print(f"{self.b}x", end="")

        if (self.c > 0): print(f"+{self.c}")

        elif (self.c < 0): print(f"{self.c}")

        print()

        if(len(self.arr) == 0): print("Нет корней")

        elif(len(self.arr) == 1): print(f"x = {self.arr[0]}")

        else: print(f"X1 = {self.arr[0]}, X2 = {self.arr[1]}")

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

    obj = Equation()

    obj.get\_coef()

    obj.solve()

    obj.see\_eq()

**LR1P.py:**

import sys

def get\_param():

    try:

        arr = [float(i) for i in sys.argv[1:]]

        if (len(arr) == 3 and arr[0] != 0):

            print("Взяты параметры из cmd")

            return arr

        else:

            print("Ошибка при взятии аргументов из cmd")

    except:

        print("Ошибка при взятии аргументов из cmd")

    ans = 0

    while ans == 0:

        try:

            print ("Введите 3 действительных числа через пробел (A, B, C)")

            arr = list(map(float, input().split())) #[float(i) for i in input().split()]

            if (len(arr) == 3 and arr[0] != 0):

                return arr

            else:

                print("Error")

                ans = 0

        except:

            print("Error")

            ans = 0

#arr[0, 1, 2] - a, b, c

def solve(arr):

    solutions = []

    d = float(arr[1]\*\*2 - 4 \* arr[0] \* arr[2])

    if (d == 0):

        x = -arr[1] / (2 \* arr[0])

        solutions.append(x)

    elif (d > 0):

        x1 = (-arr[1] + d\*\*0.5) / (2 \* arr[0])

        x2 = (-arr[1] - d\*\*0.5) / (2 \* arr[0])

        solutions.append(x1)

        solutions.append(x2)

    return solutions

def main():

    arr = get\_param()

    solutions = solve(arr)

    if (len(solutions) == 2):

        print(f"X1 = {solutions[0]}; X2 = {solutions[1]}")

    elif (len(solutions) == 1):

        print(f"X = {solutions[0]}")

    else:

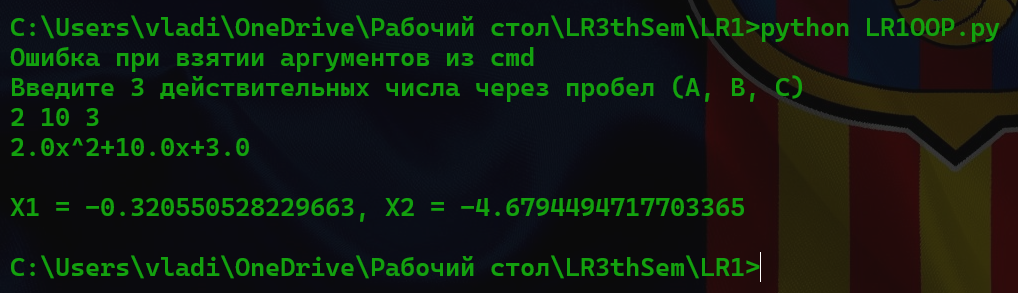
        print("Корней нет")

if '\_\_main\_\_' == \_\_name\_\_:

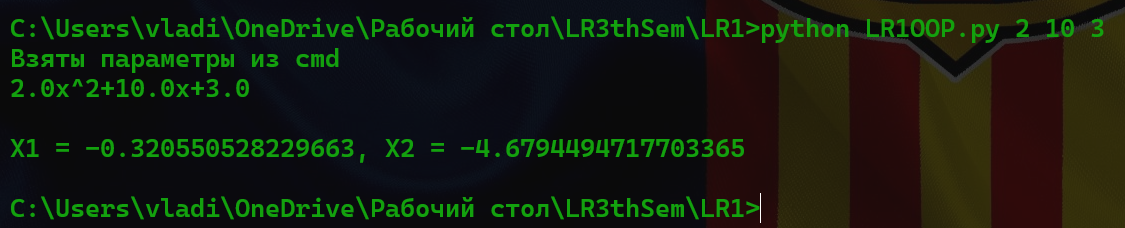
    main()

**Примеры работы.**

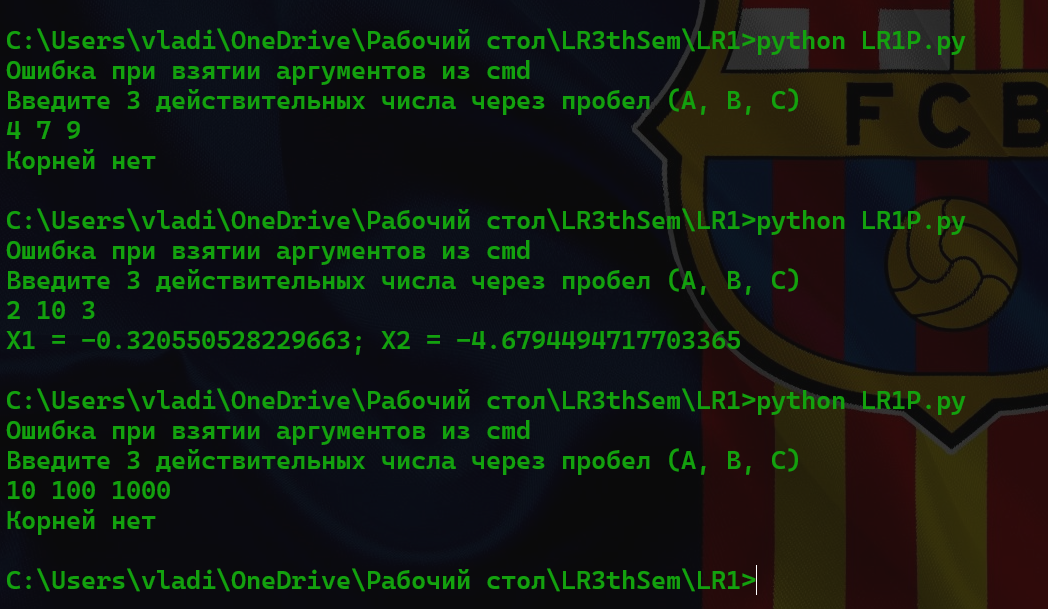
Запускаем LR1OOP.py через командную строку, но параметры на вход не даем:



Запускаем LR1OOP.py через командную строку c параметрами на вход:



Запускаем LR1P.py через командную строку, но параметры на вход не даем:



Запускаем LR1P.py через командную строку c параметрами на вход:

