**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «ПиКЯП (Парадигмы и конструкции языков программирования)»

Отчет по лабораторной работе №1

«**Основные конструкции языка Python**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студент группы ИУ5-31Б |  | Преподаватель каф. ИУ5 |
| Нагдасёв Д.М. |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2024 г

**Цель лабораторной работы:** изучение основных конструкций языка Python.

**Код программы:**

import sys

import math

class SquareRoots:

    def \_\_init\_\_(self):

        self.coef\_A = 0.0

        self.coef\_B = 0.0

        self.coef\_C = 0.0

        self.num\_roots = 0

        self.roots\_list = []

    def get\_coef(self, index, prompt):

        while True:

            try:

                coef\_str = sys.argv[index]

            except:

                print(prompt)

                coef\_str = input()

            try:

                return float(coef\_str)

            except ValueError:

                print("Некорректная переменная")

    def get\_coefs(self):

        self.coef\_A = self.get\_coef(1, 'Введите коэффициент А:')

        self.coef\_B = self.get\_coef(2, 'Введите коэффициент B:')

        self.coef\_C = self.get\_coef(3, 'Введите коэффициент C:')

    def calculate\_roots(self):

        a = self.coef\_A

        b = self.coef\_B

        c = self.coef\_C

        D = b\*b - 4\*a\*c

        if D == 0.0:

            root = -b / (2.0\*a)

            if root == 0.0:

                self.num\_roots = 1

                self.roots\_list.append(root)

            elif root > 0.0:

                self.num\_roots = 2

                self.roots\_list.append(math.sqrt(root))

                self.roots\_list.append(-math.sqrt(root))

        elif D > 0.0:

            sqD = math.sqrt(D)

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

            if root1 == 0.0:

                self.num\_roots += 1

                self.roots\_list.append(root1)

            elif root1 > 0.0:

                self.num\_roots += 2

                self.roots\_list.append(math.sqrt(root1))

                self.roots\_list.append(-math.sqrt(root1))

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

            if root2 == 0.0:

                self.num\_roots += 1

                self.roots\_list.append(root2)

            elif root2 > 0.0:

                self.num\_roots += 2

                self.roots\_list.append(math.sqrt(root2))

                self.roots\_list.append(-math.sqrt(root2))

    def print\_roots(self):

        if self.num\_roots != len(self.roots\_list):

            print(('Ошибка. Уравнение содержит {} действительных корней, ' +\

                'но было вычислено {} корней.').format(self.num\_roots, len(self.roots\_list)))

        else:

            if self.num\_roots == 0:

                print('Нет корней')

            elif self.num\_roots == 1:

                print('Один корень: {}'.format(self.roots\_list[0]))

            elif self.num\_roots == 2:

                print('Два корня: {} и {}'.format(self.roots\_list[0], \

                    self.roots\_list[1]))

            elif self.num\_roots == 3:

                print('Три корня: {}, {} и {}'.format(self.roots\_list[0], \

                    self.roots\_list[1], self.roots\_list[2]))

            elif self.num\_roots == 4:

                print('Четыре корня: {}, {}, {} и {}'.format(self.roots\_list[0], \

                    self.roots\_list[1], self.roots\_list[2], self.roots\_list[3]))

def main():

    r = SquareRoots()

    r.get\_coefs()

    r.calculate\_roots()

    r.print\_roots()

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

    main()

**Результаты выполнения программы:**







