**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»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Сайфутдинов Р.И. |  | 1. Гапанюк Ю.Е. |
|  |  |  |

**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.
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++).

**Текст программы**

**Python**

Файл lab1\_oop.py

import sys

import math

class SquareRoots:

    def \_\_init\_\_(self):

        '''

        Конструктор класса

        '''

        # Объявление коэффициентов

        self.coef\_A = 0.0

        self.coef\_B = 0.0

        self.coef\_C = 0.0

        # Количество корней

        self.num\_roots = 0

        # Список корней

        self.roots\_list = []

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

        try:

            # Пробуем прочитать коэффициент из командной строки

            coef\_str = sys.argv[index]

        except:

            # Вводим с клавиатуры

            print(prompt)

            coef\_str = input()

        # Переводим строку в действительное число

        coef = float(coef\_str)

        return coef

    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):

        '''

        Вычисление корней квадратного уравнения

        '''

        a = self.coef\_A

        b = self.coef\_B

        c = self.coef\_C

        # Вычисление дискриминанта и корней

        D = b\*b - 4\*a\*c

        if D == 0.0 and -b / (2.0\*a) > 0:

            root1 = math.sqrt(-b / (2.0\*a))

            root2 = -root1

            self.num\_roots = 2

            self.roots\_list.append(root1)

            self.roots\_list.append(root2)

        elif D > 0.0:

            sqD = math.sqrt(D)

            temp1 = (-b + sqD) / (2.0\*a)

            temp2 = (-b - sqD) / (2.0\*a)

            if temp1 > 0:

                self.num\_roots += 2

                self.roots\_list.append(math.sqrt(temp1))

                self.roots\_list.append(-math.sqrt(temp1))

            if temp2 > 0:

                self.num\_roots += 2

                self.roots\_list.append(math.sqrt(temp2))

                self.roots\_list.append(-math.sqrt(temp2))

    def print\_roots(self):

        # Проверка отсутствия ошибок при вычислении корней

        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 == 2:

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

                    self.roots\_list[1]))

            elif self.num\_roots == 4:

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

                    self.roots\_list[1], \

                    self.roots\_list[2], \

                    self.roots\_list[3]))

def main():

    '''

    Основная функция

    '''

    # Создание объекта класса

    r = SquareRoots()

    # Последовательный вызов необходимых методов

    r.get\_coefs()

    r.calculate\_roots()

    r.print\_roots()

# Если сценарий запущен из командной строки

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

    main()

# Пример запуска

# roots\_oop.py 1 0 -4

Файл lab1\_proc.py

import sys

import math

def get\_coef(index, prompt):

    try:

        # Пробуем прочитать коэффициент из командной строки

        coef\_str = sys.argv[index]

    except:

        # Вводим с клавиатуры

        print(prompt)

        coef\_str = input()

    # Переводим строку в действительное число

    coef = float(coef\_str)

    return coef

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

    result = []

    D = b\*b - 4\*a\*c

    if D == 0.0 and -b / (2.0\*a) > 0:

        root1 = math.sqrt(-b / (2.0\*a))

        root2 = - root1

        result.append(root1)

        result.append(root2)

    elif D > 0.0:

        sqD = math.sqrt(D)

        temp1 = (-b + sqD) / (2.0\*a)

        temp2 = (-b - sqD) / (2.0\*a)

        if temp1 > 0:

            result.append(math.sqrt(temp1))

            result.append(-math.sqrt(temp1))

        if temp2 > 0:

            result.append(math.sqrt(temp2))

            result.append(-math.sqrt(temp2))

    return result

def main():

    '''

    Основная функция

    '''

    a = get\_coef(1, 'Введите коэффициент А:')

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

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

    # Вычисление корней

    roots = get\_roots(a,b,c)

    # Вывод корней

    len\_roots = len(roots)

    if len\_roots == 0:

        print('Нет корней')

    elif len\_roots == 2:

        print('Два корня: {} и {}'.format(roots[0], roots[1]))

    elif len\_roots == 4:

        print('Четыре корня: {} и {} и {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))

# Если сценарий запущен из командной строки

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

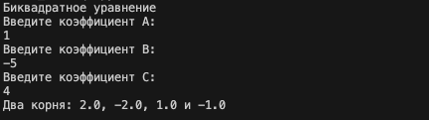
    main()

# Пример запуска

# roots\_proc.py 1 0 -4

**Примеры выполнения программы**

1. **Python**

****