

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по лабораторной работе №1
«Основные конструкции языка C#»**

Выполнил:
студент группы РТ5-31Б:
Кузнецов С. А.

Подпись и дата:

Проверил:
преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г.

Постановка задачи

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. Если коэффициент А, В, С введен некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно.
4. Корни уравнения выводятся зелёным цветом. Если корней нет, то сообщение выводится красным цветом.
5. 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. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.
6. Решить с помощью ооп и процедурной парадигмы

Текст программы

Файл «Polinom.py»(ООП)

```
import math
import sys

class Polinom:
    def __init__(self):
        self.rootsArr = None
        self.a = 0.0
        self.b = 0.0
        self.c = 0.0

    def getCoef(self, index, prompt):
        """
        Читаем коэффициент из командной строки или вводим с клавиатуры
        Args:
            index (int): Номер параметра в командной строке
            prompt (str): Приглашение для ввода коэффициента
        Returns:
            float: Коэффициент квадратного уравнения
        """
        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 enterCoefs(self):
    while True:
        self.a = self.getCoef(1, "A = ")
        self.b = self.getCoef(2, "B = ")
        self.c = self.getCoef(3, "C = ")
        if self.a == 0:
            print("invalid data")
            continue
        else:
            break

def calculate(self):

    self.enterCoefs()
    d = self.b ** 2 - 4 * self.a * self.c
    if d < 0.0:
        self.rootsArr = []
        return []
    elif d == 0.0:
        x = (-self.b) / (2 * self.a)
        self.rootsArr = self.processX(x)
    else:
        x1 = (-self.b + math.sqrt(d)) / (2 * self.a)
        x2 = (-self.b - math.sqrt(d)) / (2 * self.a)

        self.rootsArr = self.processX(x1) + self.processX(x2)
    return self.rootsArr

    @staticmethod
    def processX(x):
        if x < 0.0:
            return []
        elif x == 0.0:
            return [0.0]
        else:
            return [math.sqrt(x), -math.sqrt(x)]

def main():
    pol = Polinom()
    root = pol.calculate()

    print(root)

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

```

файл «Process.py»(процедурная)

```

import math
import sys

def getCoef(index, prompt):
    """
    Читаем коэффициент из командной строки или вводим с клавиатуры
    Args:
        index (int): Номер параметра в командной строке
    """

```

```

    prompt (str): Приглашение для ввода коэффициента
Returns:
    float: Коэффициент квадратного уравнения
'''
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 is_valid(a):
    if a == 0:
        return False
    else:
        return True

def process_roots(x):
    if x > 0:
        return [math.sqrt(x), -math.sqrt(x)]
    elif x == 0:
        return [0]
    else:
        return []

def get_roots(a, b, d):
    if d < 0:
        return []
    elif d == 0:
        x = -b / (2 * a)
        return process_roots(x)
    elif d > 0:
        x1 = (-b + math.sqrt(d)) / (2 * a)
        x2 = (-b - math.sqrt(d)) / (2 * a)
        return process_roots(x1) + process_roots(x2)

def calculate_roots():
    a = 0.0
    b = 0.0
    c = 0.0
    while not is_valid(a):
        a = getCoef(1, "A = ")
        b = getCoef(1, "B = ")
        c = getCoef(1, "C = ")
    d = (b ** 2) - 4 * a * c
    return get_roots(a, b, d)

def main():
    print(calculate_roots())

if __name__ == '__main__':
    main()

```

Экранные формы с примерами выполнения программы

```
A =
```

```
1
```

```
B =
```

```
2
```

```
C =
```

```
3
```

```
[]
```

```
A =
```

```
0
```

```
B =
```

```
1
```

```
C =
```

```
2
```

```
invalid data
```

A =

1

B =

2

C =

-10

[1.5220462510565833, -1.5220462510565833]

A =

1

B =

-10

C =

2

[3.129829312169071, -3.129829312169071, 0.4518500599615772, -0.4518500599615772]