

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

Факультет «Радиотехнический»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1  
«Основные конструкции языка Python»

Выполнил:  
Студент группы ИУ5-33Б  
Серик И. Н.  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.  
Подпись и дата:

Москва, 2023 г.

## Задание

Разработать программу для решения биквадратного уравнения.

- Программа должна быть разработана в виде консольного приложения на языке Python.
- Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
- Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( вариант задания параметров приведен в конце файла с примером кода ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
- 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.

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

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

### Текст программы с применением процедурной парадигмы

```
import sys
import math

def get_coef(index, prompt):
    """Получение коэффициентов"""
    try:
        """Чтение из командной строки"""
        coef_input = sys.argv[index]
    except:
        """Ввод с клавиатуры"""
        try:
            coef_input = float(input(prompt))
        except ValueError:
            print("Not a valid value")
            return get_coef(index, prompt)
    return float(coef_input)

def get_sqRoots(root, result):
    """Вычисление корней квадратного уравнения вида  $x^2 = n$ """
```

```

    if root >= 0:
        root = math.sqrt(root)
        if root > 0:
            result.append(-root)
        result.append(root)

def get_roots(a, b, c):
    """Вычисление корней уравнения"""
    result = []
    d = b * b - 4 * a * c
    print("Discriminant =", d)
    if d == 0.0:
        root = -b / (2.0 * a)
        get_sqRoots(root, result)
    elif d > 0.0:
        sqd = math.sqrt(d)
        root1 = (-b + sqd) / (2.0 * a)
        root2 = (-b - sqd) / (2.0 * a)
        get_sqRoots(root1, result)
        get_sqRoots(root2, result)
    return result

def main():
    a = get_coef(1, 'Enter real A:')
    b = get_coef(2, 'Enter real B:')
    c = get_coef(3, 'Enter real C:')
    print('You entered: {}x^4 + {}x^2 + {} = 0'.format(a, b, c))
    roots = get_roots(a, b, c)
    len_roots = len(roots)
    if len_roots == 0:
        print('There are no real roots')
    elif len_roots == 1:
        print('There is one real root:\n', *roots)
    else:
        print('There are {} real roots:'.format(len_roots))
        list(map(print, roots))

if __name__ == "__main__":
    main()

```

## Текст программы с применением объектно-ориентированной парадигмы

```

import sys
import math

class QuadraticEquationSolver:
    def __init__(self):
        self.a = self.get_coef(1, 'Enter real A:')
        self.b = self.get_coef(2, 'Enter real B:')
        self.c = self.get_coef(3, 'Enter real C:')
        self.roots = []

    def get_coef(self, index, prompt):
        try:
            coef_input = sys.argv[index]
        except IndexError:
            try:
                coef_input = float(input(prompt))
            except ValueError:
                print("Not a valid value")
                return self.get_coef(index, prompt)

```

```

        return float(coef_input)

def get_sqRoots(self, root):
    if root >= 0:
        root = math.sqrt(root)
        if root > 0:
            self.roots.append(-root)
        self.roots.append(root)

def get_roots(self):
    d = self.b * self.b - 4 * self.a * self.c
    print("Discriminant =", d)
    if d == 0.0:
        root = -self.b / (2.0 * self.a)
        self.get_sqRoots(root)
    elif d > 0.0:
        sqd = math.sqrt(d)
        root1 = (-self.b + sqd) / (2.0 * self.a)
        root2 = (-self.b - sqd) / (2.0 * self.a)
        self.get_sqRoots(root1)
        self.get_sqRoots(root2)

def show_equation(self):
    print('You entered: {}x^4 + {}x^2 + {} = 0'.format(self.a, self.b, self.c))

def show_solution(self):
    len_roots = len(self.roots)
    if len_roots == 0:
        print('There are no real roots')
    elif len_roots == 1:
        print('There is one real root:\n', *self.roots)
    else:
        print('There are {} real roots:'.format(len_roots))
        list(map(print, self.roots))

def main():
    equation_solver = QuadraticEquationSolver()
    equation_solver.show_equation()
    equation_solver.get_roots()
    equation_solver.show_solution()

if __name__ == "__main__":
    main()

```

## Текст программы на C#

```

using System;

class Program
{
    static double GetCoef(int index, string prompt)
    {
        double coef;
        Console.Write(prompt);
        try
        {
            coef = double.Parse(Environment.GetCommandLineArgs()[index]);
        }
        catch
        {
            while (!double.TryParse(Console.ReadLine(), out coef))
            {
                Console.WriteLine("Not a valid value");
            }
        }
    }
}

```

```

        Console.WriteLine(prompt);
    }
}
return coef;
}

static void GetSquareRoots(double root, ref double[] result)
{
    if (root >= 0)
    {
        root = Math.Sqrt(root);
        if (root > 0)
        {
            Array.Resize(ref result, result.Length + 1);
            result[result.Length - 1] = -root;
        }
        Array.Resize(ref result, result.Length + 1);
        result[result.Length - 1] = root;
    }
}

static double[] GetRoots(double a, double b, double c)
{
    double[] result = new double[0];
    double d = b * b - 4 * a * c;
    Console.WriteLine("Discriminant = " + d);
    if (d == 0.0)
    {
        double root = -b / (2.0 * a);
        GetSquareRoots(root, ref result);
    }
    else if (d > 0.0)
    {
        double sqd = Math.Sqrt(d);
        double root1 = (-b + sqd) / (2.0 * a);
        double root2 = (-b - sqd) / (2.0 * a);
        GetSquareRoots(root1, ref result);
        GetSquareRoots(root2, ref result);
    }
    return result;
}

static void Main()
{
    double a = GetCoef(1, "Enter real A:");
    double b = GetCoef(2, "Enter real B:");
    double c = GetCoef(3, "Enter real C:");
    Console.WriteLine($"You entered: {a}x^4 + {b}x^2 + {c} = 0");
    double[] roots = GetRoots(a, b, c);
    int lenRoots = roots.Length;
    if (lenRoots == 0)
    {
        Console.WriteLine("There are no real roots");
    }
    else if (lenRoots == 1)
    {
        Console.WriteLine($"There is one real root:\n{roots[0]}");
    }
    else
    {
        Console.WriteLine($"There are {lenRoots} real roots:");
        Array.ForEach(roots, root => Console.WriteLine(root));
    }
}
}

```

## Результат

Enter real A:Not a valid value

Enter real A:Not a valid value

Enter real A:Enter real B:Not a valid value

Enter real B:Enter real C:Not a valid value

Enter real C:You entered:  $2.0x^4 + 3.0x^2 + -4.0 = 0$

Discriminant = 41.0

There are 2 real roots:

-0.9223779373761127

0.9223779373761127

Press any key to continue . . .