

**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 лабораторным работам №1-3
«Основные конструкции языка Python»
«Модульное тестирование на языке Python»

Выполнил:
студент группы ИУ5-32Б
Носков Алексей

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий
Евгеньевич

Москва, 2024 г.

Постановка задачи

Задание.

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов A , B , C , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта). Коэффициенты A , B , C 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 A , B , C 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

lab1.py

```
import sys
import math

def get_coeff(coeff_name):
    while True:
        s = input(f'Введите коэффициент {coeff_name}: ')
        try:
            return float(s)
        except:
            print("Введён неверный коэффициент!")

def no_roots(): print("Действительных корней нет.")

def print_roots(number, roots_set = set()):
    if number == 0:
        no_roots()
        return
    print(f'Количество корней: {number}')
    print("Корни:")
    for x in roots_set: print(x)

def solve_equation(a, b, c):
    roots = set()
    if a == 0:
        if b == 0:
            if c == 0:
                print("R")
```

```

        return
    print_roots(0)
    return
    roots.add(-c / b)
    print_roots(1, roots)
    return
d = b * b - 4 * a * c
if d == 0:
    new_root = (-b) / (2 * a)
    if new_root > 0:
        roots.add(math.sqrt(new_root))
        roots.add(-math.sqrt(new_root))
    elif new_root == 0:
        roots.add(0)
elif d > 0:
    for i in range(2):
        new_root = (-b + (-1)**i * math.sqrt(d)) / (2 * a)
        if new_root > 0:
            roots.add(math.sqrt(new_root))
            roots.add(-math.sqrt(new_root))
        elif new_root == 0:
            roots.add(0)
else:
    print_roots(0)
    return

print_roots(len(roots), roots)

A = get_coeff('a')
B = get_coeff('b')
C = get_coeff('c')

solve_equation(A, B, C)

```

roots_number.py

```

import math

def roots_number(a, b, c):
    roots = 0
    if a == 0:
        if b == 0:
            if c == 0:
                return False
            return 0
        return 2

    d = b * b - 4 * a * c
    if d < 0: return 0
    if d == 0:
        if -b / a > 0: return 2
        elif -b / a == 0: return 1
    return 0

```

```

for i in range(2):
    roots += 2 * (((-b) + (-1)**i * math.sqrt(d)) / (2 * a) > 0)
return roots

```

tests.py

```

import unittest
from roots_number import *

class TestRootsNumber(unittest.TestCase):

    def test(self):
        self.assertEqual(roots_number(1, 1, 1), 0)
        self.assertEqual(roots_number(1, 4, 4), 0)
        self.assertEqual(roots_number(1, -4, 4), 2)
        self.assertEqual(roots_number(0, 1, -1), 2)
        self.assertEqual(roots_number(1, -5, 6), 4)
        self.assertEqual(roots_number(1, 0, -1), 2)

if __name__ == "__main__":
    unittest.main()

```

Program.cs

```

using System.Drawing;

void setColor(ConsoleColor color) {
    Console.ForegroundColor = color;
}

double getCoeff(ref double x, char coeffName) {
    string? input;
    while (true) {
        Console.WriteLine($"Введите коэффициент {coeffName}: ");
        input = Console.ReadLine();
        if (double.TryParse(input, out double numericValue)) {
            x = numericValue;
            break;
        } else {
            setColor(ConsoleColor.Red);
            Console.WriteLine("Введён неверный коэффициент!");
            Console.ResetColor();
        }
    }
    return x;
}

void solveEquation(in double a, in double b, in double c) {
    setColor(ConsoleColor.Green);
    if (a == 0) {
        if (b == 0) {
            if (c == 0) {
                Console.WriteLine("R");
                return;
            }
        }
    }
}

```

```

        setColor(ConsoleColor.Red);
        Console.WriteLine("Действительных корней нет.");
        return;
    }
    Console.WriteLine(-c / b);
    return;
}
double d = b * b - 4 * a * c;
List<double> roots = [];
double newRoot;
if (d == 0) {
    newRoot = -b / (2 * a);
    if (newRoot > 0) {
        roots.Add(Math.Sqrt(newRoot));
        roots.Add(-Math.Sqrt(newRoot));
    } else if (newRoot == 0) {
        roots.Add(0);
    }
} else if (d > 0) {
    for (int i = 0; i < 2; i++) {
        newRoot = (-b + Math.Pow(-1, i) * Math.Sqrt(d)) / (2 * a);
        if (newRoot > 0) {
            roots.Add(Math.Sqrt(newRoot));
            roots.Add(-Math.Sqrt(newRoot));
        } else if (newRoot == 0) {
            roots.Add(0);
        }
    }
} else {
    setColor(ConsoleColor.Red);
    Console.WriteLine("Действительных корней нет.");
    return;
}
if (roots.Count > 0) {
    Console.WriteLine($"Количество корней: {roots.Count}");
    Console.WriteLine("Корни:");
    foreach (double root in roots) {
        Console.WriteLine($"{root}");
    }
} else {
    setColor(ConsoleColor.Red);
    Console.WriteLine("Действительных корней нет.");
}
}

double a = 0, b = 0, c = 0;
getCoeff(ref a, 'a');
getCoeff(ref b, 'b');
getCoeff(ref c, 'c');

solveEquation(a, b, c);

```

Вывод программы на языке Python

Введите коэффициент a: 1
Введите коэффициент b: 2
Введите коэффициент c: 1
Действительных корней нет.

Введите коэффициент a: 1
Введите коэффициент b: -2
Введите коэффициент c: 1
Количество корней: 2
Корни:
1.0
-1.0

Вывод тестов на языке Python

.

Ran 1 test in 0.000s

ОК

Вывод программы на языке C#

Введите коэффициент a: 1
Введите коэффициент b: -2
Введите коэффициент c: 1
Количество корней: 2
Корни:
1
-1

Введите коэффициент a: 1
Введите коэффициент b: 3
Введите коэффициент c: 4
Действительных корней нет.