

**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:
студент группы ИУ5-32Б
Гайнуллин А. М.

Проверил:
преподаватель каф. ИУ5
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 A , B , C , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A , B , C могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2.
4. 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.
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 C++).

Текст программы

Python

Файл qr.py

```
import sys
import math

def get_coef(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 get_roots(a, b, c):
    """
    Вычисление корней квадратного уравнения

    Args:
        a (float): коэффициент A
        b (float): коэффициент B
        c (float): коэффициент C
```

```
    Returns:
        list[float]: Список корней
    """
```

```
    result = []
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        roota = math.sqrt(root)
        rootb = math.sqrt(root) * (-1)
        result.append(roota)
        result.append(rootb)
    elif D > 0.0:
        sqD = math.sqrt(D)
        root1 = (-b + sqD) / (2.0*a)
        root2 = (-b - sqD) / (2.0*a)
        root1a = math.sqrt(root1)
        root1b = math.sqrt(root1) * (-1)
        root2a = math.sqrt(root2)
        root2b = math.sqrt(root2) * (-1)
        result.append(root1a)
        result.append(root1b)
        result.append(root2a)
```

```

        result.append(root2b)
    return result

def main():
    """
    Основная функция
    """
    print("Биквадратное уравнение ")
    a = get_coef(1, 'Введите коэффициент A:')
    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('Два корня: {}, {}, {} и {}'.format(roots[0], roots[1],roots[2], roots[3]))

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

# Пример запуска
# qr.py 1 0 -4

```

JavaScript

Файл lab1.js

```

const readline = require('readline');

const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});

```

```
function getValidNumber(prompt) {
  return new Promise((resolve) => {
    rl.question(prompt, (input) => {
      let number = parseFloat(input);
      if (!isNaN(number)) {
        resolve(number);
      } else {
        console.log('Введите корректное число.');
        resolve(getValidNumber(prompt));
      }
    });
  });
}
```

```
function calculateDiscriminant(a, b, c) {
  return b * b - 4 * a * c;
}
```

```
function getRoots(a, b, c) {
  const discriminant = calculateDiscriminant(a, b, c);

  if (discriminant > 0) {
    const root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
    const root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
    return [root1, root2];
  } else if (discriminant === 0) {
    const root = -b / (2 * a);
    return [root];
  } else {
    return [];
  }
}
```

```
function getRoots1(a,b,c){
  let result = []
  let D = b*b - 4*a*c
  if (D == 0.0){
    let root = -b / (2.0*a)
    let roota = Math.sqrt(root)
    let rootb = Math.sqrt(root) * (-1)
    result.push(roota)
```

```

    result.push(rootb)
    return [roota, rootb]
}
else if (D > 0.0){
    let sqD = Math.sqrt(D)
    let root1 = (-b + sqD) / (2.0*a)
    let root2 = (-b - sqD) / (2.0*a)
    let root1a = Math.sqrt(root1)
    let root1b = Math.sqrt(root1) * (-1)
    let root2a = Math.sqrt(root2)
    let root2b = Math.sqrt(root2) * (-1)
    result.push(root1a)
    result.push(root1b)
    result.push(root2a)
    result.push(root2b)
    return [root1a, root1b, root2a, root2b]
}
else{
    return []
}
}

async function main() {
    let a, b, c;

    if (process.argv.length === 5) {
        // Если коэффициенты заданы через командную строку
        a = parseFloat(process.argv[2]);
        b = parseFloat(process.argv[3]);
        c = parseFloat(process.argv[4]);
    } else {
        // Ввод коэффициентов с клавиатуры
        a = await getValidNumber('Введите коэффициент A: ');
        b = await getValidNumber('Введите коэффициент B: ');
        c = await getValidNumber('Введите коэффициент C: ');
    }

    const roots = getRoots1(a, b, c);

```

```

if (roots.length === 0) {
    console.log("Уравнение не имеет действительных корней.");
} else {
    console.log("Действительные корни уравнения: ", roots);
}

rl.close();
}

main();

```

Примеры выполнения программы

1. Python

```

2a00:1370:8186:68b7:80d9:3c67:f1b0:959e 192.168.1.69 greg@Admin ~/D/Y/3/l/c/lab1_code> python3
/lab1_code/qr.py
Биквадратное уравнение
Введите коэффициент A:
1
Введите коэффициент B:
-5
Введите коэффициент C:
4
Два корня: 2.0, -2.0, 1.0 и -1.0

```

2. JavaScript

```

2a00:1370:8186:68b7:80d9:3c67:f1b0:959e 192.168.1.69 greg@Admin ~/D/Y/3/l/c/lab1_code> node "/Users/greg/Desktop/Учеба
code/lab1.js"
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 4
Действительные корни уравнения: [ 2, -2, 1, -1 ]

```