

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

Факультет Радиотехнический  
Кафедра РТ5

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №6  
«Основные конструкции языка Rust.»

Выполнил:

студент группы РТ5-31Б:  
Салищев И.Д.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023

## Описание задания

Разработать программу для решения [биквадратного уравнения](#).

1. Программа должна быть разработана в виде консольного приложения на языке Rust.
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 ( [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](#).
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.

## Текст программы

```
use std::io;

// Функция для безопасного чтения вещественных чисел
fn read_float(prompt: &str) -> f64 {
    loop {
        println!("{}", prompt);
        let mut input = String::new();
        io::stdin().read_line(&mut input).expect("Ошибка чтения строки");
        match input.trim().parse::<f64>() {
            Ok(num) => return num,
            Err(_) => println!("Пожалуйста, введите действительное число."),
        }
    }
}

// Функция для решения биквадратного уравнения
fn solve_biquadratic(a: f64, b: f64, c: f64) -> Vec<f64> {
    if a.abs() < f64::EPSILON {
        vec![]
    } else {
        let d = b * b - 4.0 * a * c;
        if d < 0.0 {
            vec![]
        } else {
            let sqrt_d = d.sqrt();
```

```

        let two_a = 2.0 * a;
        let roots = [(-b + sqrt_d) / two_a, (-b - sqrt_d) / two_a];
        roots.iter().filter_map(|&x| {
            if x >= 0.0 {
                Some(x.sqrt())
            } else {
                None
            }
        }).flat_map(|x| vec![x, -x]).collect()
    }
}

fn main() {
    let a = read_float("Введите коэффициент a:");
    let b = read_float("Введите коэффициент b:");
    let c = read_float("Введите коэффициент c:");

    let roots = solve_biquadratic(a, b, c);

    if roots.is_empty() {
        println!("Нет действительных корней.");
    } else {
        println!("Корни уравнения: {:?}", roots);
    }
}

```

## Экранные формы с примерами выполнения программы

```

Введите коэффициент a:
1
Введите коэффициент b:
-7
Введите коэффициент c:
6
Корни уравнения: [2.449489742783178, -2.449489742783178, 1.0, -1.0]

```