**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет Радиотехнический

Кафедра РТ5

Курс «Программирование на основе классов и шаблонов»

Отчет по лабораторной работе №4

## «Основные конструкции языка Rust»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б:  Салищев И.Д. |  | преподаватель каф. ИУ5  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023

**Описание задания**

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

1. Программа должна быть разработана в виде консольного приложения на языке Rust.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( [вариант задания параметров приведен в конце файла с примером кода](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
4. 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.

**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.");

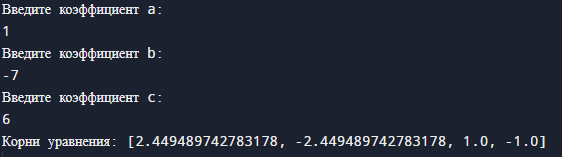
    } else {

        println!("Корни уравнения: {:?}", roots);

    }

}

**Экранные формы с примерами выполнения программы**

****