**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
«Решение биквадратного уравнения на Rust»

Выполнил:
Студент группы ИУ5-31Б
Баженов Никита

Проверил:
Нардид А. Н.

2025 г.

Задание

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.

Листинг программы

main.rs

```rust
use std::io;
use std::env;
use std::cmp::Ordering;


fn parse_number_from_string(s: &String) -> Option<f32> {
    match s.trim().parse::<f32>() {
        Ok(result) => {
            if result.is_nan() || result.is_infinite() {
                return None;
            }

            return Some(result);
        },
        Err(_) => return None,
    }
}

fn read_coefs(msg: &'static str, i: usize) -> f32 {
    let mut input_str = String::new();

    if let Some(value) = env::args().nth(i) {
        match parse_number_from_string(&value) {
            Some(num) => return num,
            None => println!("Argument {} cannot be parsed!", i),
        };
    }

    loop {
        input_str.clear();
        println!("{}", msg);
```

```rust

        match io::stdin().read_line(&mut input_str) {
            Ok(_) => {},
            Err(_) => {
                println!("Failed to read line!");
                continue;
            }
        };

        match parse_number_from_string(&input_str) {
            Some(num) => return num,
            None => println!("Please, enter correct number!"),
        };
    }
}

fn print_roots_for(a: f32, b: f32, d_sqrt: f32, msg: &'static str) {
    println!("{}", msg);

    let value = (-b + d_sqrt) / (2.0 * a);

    match value.total_cmp(&0.0) {
        Ordering::Less => println!("No roots"),
        Ordering::Equal => println!("Root: 0.0"),
        Ordering::Greater => {
            let value = value.sqrt();
            println!("Roots: {} {}", value, -value);
        }
    }
}

fn main() {
    let a = read_coefs("Enter A coef:", 1);
    let b = read_coefs("Enter B coef:", 2);
    let c = read_coefs("Enter C coef:", 3);

    if a == 0.0 {
        let value = -c / b;

        match value.total_cmp(&0.0) {
            Ordering::Less => println!("No roots"),
            Ordering::Equal => println!("Root: 0.0"),
            Ordering::Greater => {
                let value = value.sqrt();
                println!("Roots: {} {}", value, -value);
            }
        }

        return;
    }

    let d = b * b - 4.0 * a * c;

    if d < 0.0 {
        println!("(D < 0) => no roots");
        return;
    }

    let d_sqrt = d.sqrt();

    if d_sqrt == 0.0 {
        print_roots_for(a, b, 0.0, "sqrt(D) == 0.0:");
        return;
```

```
95     }
96
97     print_roots_for(a, b, d_sqrt, "First case:");
98     print_roots_for(a, b, -d_sqrt, "Second case:");
99 }
```

Примеры выполнения

Пример 1

```
$ cargo run
Enter A coef:
1
Enter B coef:
5
Enter C coef:
0
First case:
Root: 0.0
Second case:
No roots
```

Пример 2

```
$ cargo run 1 1 1
(D < 0) => no roots
```

Пример 3

```
$ cargo run inf 2 3
Argument 1 cannot be parsed!
Enter A coef:
-1
First case:
No roots
Second case:
Roots: 1.7320508 -1.7320508
```

Пример 4

```
$ cargo run 2 hello 5
Argument 2 cannot be parsed!
Enter B coef:
10
First case:
No roots
Second case:
No roots
```

Пример 5

```
$ cargo run 2 -7
Enter C coef:
1
First case:
Roots: 1.8305138 -1.8305138
Second case:
Roots: 0.38628864 -0.38628864
```