**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 №5

«Разработка на языке программирования Rust»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Никитенко Михаил |  | Гапанюк Юрий |
|  |  |  |
|  |  |  |

Москва, 2024 г.

Постановка задачи

Реализуйте любое из заданий курса на языке программирования Rust.

Разработайте хотя бы один макрос.

**Текст программы**

B2\_Solve.rs

use std::env;

use std::io;

fn main() {

    let args: Vec<String> = env::args().collect();

    let (a, b, c) = if args.len() == 4 {

        (

            args[1].parse::<f64>().unwrap\_or\_else(|\_| {

                println!("Некорректный аргумент для A. Введите число:");

                get\_number\_from\_user()

            }),

            args[2].parse::<f64>().unwrap\_or\_else(|\_| {

                println!("Некорректный аргумент для B. Введите число:");

                get\_number\_from\_user()

            }),

            args[3].parse::<f64>().unwrap\_or\_else(|\_| {

                println!("Некорректный аргумент для C. Введите число:");

                get\_number\_from\_user()

            }),

        )

    } else {

        println!("Введите коэффициент A:");

        let a = get\_number\_from\_user();

        println!("Введите коэффициент B:");

        let b = get\_number\_from\_user();

        println!("Введите коэффициент C:");

        let c = get\_number\_from\_user();

        (a, b, c)

    };

    let discriminant = b \* b - 4.0 \* a \* c;

    if discriminant > 0.0 {

        let x1 = (-b - discriminant.sqrt()) / (2.0 \* a);

        let x2 = (-b + discriminant.sqrt()) / (2.0 \* a);

        println!("Два различных действительных корня: x1 = {}, x2 = {}", x1, x2);

    } else if discriminant == 0.0 {

        let x = -b / (2.0 \* a);

        println!("Один действительный корень: x = {}", x);

    } else {

        println!("Нет действительных корней");

    }

}

fn get\_number\_from\_user() -> f64 {

    loop {

        let mut input = String::new();

        io::stdin()

            .read\_line(&mut input)

            .expect("Ошибка при чтении строки");

        match input.trim().parse() {

            Ok(num) => return num,

            Err(\_) => println!("Некорректный ввод. Введите число:"),

        }

    }

}

Marcos.rs

#[derive(Debug)]

struct Point {

    x: i32,

    y: i32,

}

fn main() {

    let point = Point { x: 10, y: 20 };

    println!("{:?}", point);

}

**Обработка результатов**





