



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования**

**«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика, искусственный интеллект и
системы управления»**

**Кафедра «Системы обработки информации и
управления»**

**Лабораторная работа №6 по курсу
«Базовые компоненты интернет-технологий»**

**Выполнил:
студент группы ИУ5-33Б
Иванченко Максим**

**Проверил:
Доцент кафедры ИУ5
Гапанюк Юрий Евгеньевич**

2022 г.

Постановка задачи

1. Реализуйте любое из заданий курса на языке программирования Rust (выбрано задание из лабораторной работы №1).
2. Разработайте хотя бы один макрос.
3. Разработайте модульные тесты (не менее 3 тестов).

Текст программы

```
// Макрос: сумма нескольких целых чисел
macro_rules! summa_of_int32 {
    $( $x:expr ),* => {
        {
            let mut sum: i32 = 0;
            $(
                sum += $x;
            )*
            sum
        }
    };
}

// Через терминал
fn get_terminal_coefs_into(coefs: &mut Vec<f64>) {
    let mut args: Vec<String> = std::env::args().collect(); // считываем все
    параметры
    // Параметры начинаются с индекса 1, потому что 0 - имя программы
    // Введем массив из имен коэффициентов для удобства
    let coef_names = [String::from("A"), String::from("B"), String::from("C")];
    for i in 1..4 {
        if i < args.len() {
            loop {
                match args[i].trim().parse::<f64>() {
                    Ok(value) => {
                        break coefs.push(value);
                    }
                    Err(_) => {
                        println!("Введите коэффициент {} заново!", coef_names[i-
1]);
                        std::io::stdin().read_line(&mut args[i]).expect("Ошибка
ввода!");
                        continue
                    }
                }
            }
        }
        else {
            let mut input: String = String::new(); // Строка для ввода
            loop {
                println!("Введите коэффициент {}: ", coef_names[i-1]);
                input = String::new();
                std::io::stdin().read_line(&mut input).expect("Ошибка ввода!");
                match input.trim().parse::<f64>() {
                    Ok(value) => {
                        break coefs.push(value);
                    }
                    Err(_) => {
                        print!("Неверно! ");
                        continue
                    }
                }
            }
        }
    }
}

// Ввод
fn get_coef(message: &String) -> f64 {
    let mut input = String::new();
    loop {
        println!("{}", message);
    }
}

// Через консоль
fn get_coef(message: &String) -> f64 {
    let mut input = String::new();
    loop {
        println!("{}", message);
    }
}
```

```

std::io::stdin().read_line(&mut input).expect("Ошибка ввода");
match input.trim().parse::<f64>() {
    Ok(value) => {
        break value;
    }
    Err(_) => {
        println!("Неверный формат ввода!");
        continue;
    }
}
}

fn solution(a: f64, b: f64, c: f64) -> Vec<f64>{
    let mut result: Vec<f64> = Vec::new();
    let disk: f64 = b.powf(2.0) - 4.0*a*c;
    if disk == 0f64 {
        result.push(-b/(2.0*a));
    }
    else if disk > 0f64 {
        result.push((-b-disk.sqrt())/(2.0*a));
        result.push((-b+disk.sqrt())/(2.0*a));
    }
    return result;
}

fn main() {
    // Ввод коэффициентов из консоли
    /* let a: f64 = get_coef(&String::from("Введите коэффициент A:"));
    let b: f64 = get_coef(&String::from("Введите коэффициент B:"));
    let c: f64 = get_coef(&String::from("Введите коэффициент C:"));
    // Нахождение решения
    let solution = solution(a, b, c); */

    // Ввод коэффициентов из терминала
    let mut coeffs: Vec<f64> = Vec::new();
    get_terminal_coefs_into(&mut coeffs);
    // Нахождение решения
    let solution = solution(coeffs[0], coeffs[1], coeffs[2]);

    // Вывод результата
    match solution.len() {
        0 => println!("Нет решений!"),
        1 => println!("Одно решение уравнения: {:.5}", solution[0]),
        2 => println!("Два решения уравнения: {:.5} и {:.5}", solution[0],
solution[1]),
        _ => println!("Что-то пошло не так!")
    }

    // Проверка макроса
    println!("Сумма чисел 1, 3, 5, 7 равна:");
    println!("{}", summa_of_int32!(1, 3, 5, 7));
}

#[cfg(test)]
mod tests {
    use super::*;

    #[test]
    fn test_summa_of_int32() {
        assert_eq!(summa_of_int32!(1, 3, 5, 7), 16);
    }
    #[test]
    fn test_solution_2_ans() {

```

```
        assert_eq!(solution(1.0, 0.0, -4.0), vec![-2.0, 2.0]);
    }
    #[test]
    fn test_solution_1_ans() {
        assert_eq!(solution(1.0, -2.0, 1.0), vec![-1.0]); // намеренно ставлю -1
    }
}
```

Результаты

```
PS C:\Users\maxim\PycharmProjects\BKIT_2022\Lab6> cargo run -- 1 0
warning: value assigned to `input` is never read
--> src\main.rs:35:21
|
|
35 |         let mut input: String = String::new(); // Строка для ввода
|             ^^^^^^
|
|
= note: `[warn(unused_assignments)]` on by default
= help: maybe it is overwritten before being read?

warning: crate `Lab6` should have a snake case name
|
= note: `[warn(non_snake_case)]` on by default
= help: convert the identifier to snake case: `lab6`

warning: `Lab6` (bin "Lab6") generated 2 warnings
    Finished dev [unoptimized + debuginfo] target(s) in 0.01s
    Running `target\debug\Lab6.exe 1 0`
Введите коэффициент C:
dfggh
Неверно! Введите коэффициент C:
-4
Два решения уравнения: -2.00000 и 2.00000
Сумма чисел 1, 3, 5, 7 равна:
16
PS C:\Users\maxim\PycharmProjects\BKIT_2022\Lab6>
```

```
PS C:\Users\maxim\PycharmProjects\BKIT_2022\Lab6> cargo test
  Compiling Lab6 v0.1.0 (C:\Users\maxim\PycharmProjects\BKIT_2022\Lab6)
warning: value assigned to `input` is never read
--> src\main.rs:35:21
|
35 |         let mut input: String = String::new(); // Строка для ввода
|               ^^^^^^
|
= note: `[warn(unused_assignments)]` on by default
= help: maybe it is overwritten before being read?

warning: crate `Lab6` should have a snake case name
|
= note: `[warn(non_snake_case)]` on by default
= help: convert the identifier to snake case: `lab6`

warning: `Lab6` (bin "Lab6" test) generated 2 warnings
  Finished test [unoptimized + debuginfo] target(s) in 0.44s
  Running unittests src\main.rs (target\debug\deps\Lab6-3b6316361c0f5a09.exe)

running 3 tests
test tests::test_solution_2_ans ... ok
test tests::test_solution_1_ans ... FAILED
test tests::test_summa_of_int32 ... ok
```