

**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 №3  
«Основные конструкции языка Rust»

Выполнил:  
студент группы ИУ5-33Б  
Малышева Екатерина  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.  
Подпись и дата:

Москва, 2024 г.

# Постановка задачи

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

1. Программа должна быть разработана в виде консольного приложения на языке Python.
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](#) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки](#).
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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

## Текст программы на языке Rust

### Main.rs

```
use std::env;
use rand::Rng;

fn discriminate(a: i32, b: i32, c: i32) -> f64 {
    (b * b) as f64 - 4.0 * (a * c) as f64
}

fn calc_solutions(discr: f64, a: i32, b: i32) -> Vec<f64> {
    let mut roots = Vec::new();
    if discr == 0.0 {
        roots.push((-b as f64 / (2.0 * a as f64)));
    }
    if discr > 0.0 {
        roots.push((-b as f64 + discr.sqrt()) / (2.0 * a as f64));
        roots.push((-b as f64 - discr.sqrt()) / (2.0 * a as f64));
    }
}
```

```

    roots
}

fn get_abc(index: usize, text: &str) -> f64 {
    env::args().nth(index).map_or_else(
        || {
            println!("{}", text);
            let mut input = String::new();
            std::io::stdin().read_line(&mut input).unwrap();
            input.trim().parse::<f64>().unwrap()
        },
        |arg| arg.parse::<f64>().unwrap(),
    )
}

fn main() {
    println!("Do you want to set the coefficients yourself?");
    println!("1=yes, 2=no");
    let mut koef = String::new();
    std::io::stdin().read_line(&mut koef).unwrap();
    let koef: i32 = koef.trim().parse().unwrap();

    let (a, b, c);
    if koef == 1 {
        a = get_abc(1, "A: ") as i32;
        b = get_abc(2, "B: ") as i32;
        c = get_abc(3, "C: ") as i32;
    } else {
        let mut rng = rand::thread_rng();
        a = rng.gen_range(1..=100);
        b = rng.gen_range(1..=100);
        c = rng.gen_range(1..=100);
        println!("A = {}\nB = {}\nC = {}", a, b, c);
    }

    let discr = discriminate(a, b, c);
    let solutions = calc_solutions(discr, a, b);

    match solutions.len() {
        0 => println!("No roots. Discriminate < 0"),
        1 => {
            println!("1 Solution:");
            println!("{}", solutions);
        }
        _ => {
            println!("2 Solutions:");
            println!("{}", solutions);
        }
    }
}

```

```
}
```

## Cargo.toml

```
[package]
name = "roller"
version = "0.1.0"
authors = ["User"]
edition = "2018"
```

```
[[bin]]
name = "main"
path = "src/main.rs"
# See more keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html
```

```
[dependencies]
rand="0.8.3"
```

## Полученный результат

```
PS C:\Users\UserPC\Desktop\пикап\раст> cargo run
  Compiling roller v0.1.0 (C:\Users\UserPC\Desktop\пикап\раст)
  Finished `dev` profile [unoptimized + debuginfo] target(s) in 2.11s
  Running `target\debug\main.exe`
Do you want to set the coefficients yourself?
1=yes, 2=no
1
A:
3
B:
7
C:
2
2 Solutions:
[-0.3333333333333333, -2.0]
PS C:\Users\UserPC\Desktop\пикап\раст> █
```