

Московский Государственный Технический Университет им. Баумана

Отчет по лабораторной работе №6
«Разработка на языке программирования Rust»
По дисциплине «Базовые компоненты интернет технологий»

Выполнила:
Студентка группы ИУ5-32Б
Алцыбеева Маргарита
15.12.2022

Проверил:
Преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Москва, 2022 г.

Задание:

Реализуйте любое из заданий курса на языке программирования Rust.
Разработайте хотя бы один макрос.
Разработайте модульные тесты (не менее 3 тестов).

Текст программы:

```
use std::io;

#[derive(Debug, Copy, Clone)]
struct Equation{
    a: f64,
    b: f64,
    c: f64,
    disk: f64,
    count: usize,
    result: [f64; 4],
}

impl Equation {
    fn calculate(&mut self) -> ([f64;4]){
        self.disk = self.b.powi(2) - 4.0* self.a * self.c;
        if self.disk == 0.0 {
            let root = -1.0*self.b/(2.0*self.a);
            if root >= 0.0 {
                let r1 = root.sqrt();
                let r2 = -1.0*r1;
                if r1 == r2{
                    self.result[self.count] = r1;
                    self.count+=1;
                }
            }
            else{
                self.result[self.count] = r1;
                self.count+=1;
                self.result[self.count] = r2;
                self.count+=1;
            }
        }
        else if self.disk > 0.0 {
            let root1 = (-self.b + self.disk.sqrt())/(2.0*self.a);
            let root2 = (-self.b - self.disk.sqrt())/(2.0*self.a);
            if root1 > 0.0 {
                let rt11 = root1.sqrt();
                let rt12 = -1.0*rt11;
                self.result[self.count] = rt11;
                self.count += 1;
                self.result[self.count] = rt12;
                self.count += 1;
            }
            else if root1 == 0.0{
                self.result[self.count] = root1;
                self.count += 1;
            }
            if root2 > 0.0 {
                let rt21 = root2.sqrt();
                let rt22 = -1.0*rt21;
                self.result[self.count] = rt21;
                self.count += 1;
                self.result[self.count] = rt22;
                self.count += 1;
            }
            else if root2 == 0.0{
                self.result[self.count] = root2;
                self.count += 1;
            }
        }
    }
}
```

```

        return self.result;
    }

    fn get_coef(message: &str) -> f64{
        loop{
            let mut input = String::new();
            println!("{}", message);
            io::stdin().read_line(&mut input).expect("Error");
            match input.trim().parse() {
                Ok(val) => {
                    return val;
                }
                Err(_) => {
                    continue;
                }
            }
        }
    };
}

fn get_coefs(&mut self) -> () {
    self.a = Equation::get_coef("Enter the coef A:");
    self.b = Equation::get_coef("Enter the coef B:");
    self.c = Equation::get_coef("Enter the coef C:");
}

}

macro_rules! print_result {
}

fn main(){
    let mut eq = Equation{
        a: 0.0,
        b: 0.0,
        c: 0.0,
        disk_r: 0.0,
        result: [0.0, 0.0, 0.0, 0.0],
        count: 0,
    };
    eq.get_coefs();
    eq.calculate();
    for i in 0..eq.count{
        println!("{}", eq.result[i]);
    }
    //print_result!();
}

#[cfg(test)]
mod tests{
    use crate::Equation;

    #[test]
    fn test_calc(){
        let mut eq1 = Equation{a:1.0, b: -10.0, c: 9.0, disk_r: 0.0, count: 0,
result: [0.0, 0.0, 0.0, 0.0] };
        assert_eq!(eq1.calculate(), [3.0, -3.0, 1.0,-1.0] );
    }
}

```

Примеры работы программы:

```
Enter the coef A:
fgnj
Enter the coef A:
1
Enter the coef B:
-10
Enter the coef C:
9
3
-3
1
-1

Process finished with exit code 0
```

```
PS C:\Users\marrg\PycharmProjects\lab6> cargo test
   Compiling untitled1 v0.1.0 (C:\Users\marrg\PycharmProjects\lab6)
   Finished test [unoptimized + debuginfo] target(s) in 0.64s
   Running unittests src\main.rs (target\debug\deps\untitled1-a1a167b8d92924ff.exe)

running 1 test
test tests::test_calc ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```