Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе № 6 по курсу Базовые компоненты интернет-технологий

"Программирование на ЯП Rust."

Исполнитель:

Крайников Иван Александрович, РТ5-31Б

Проверил:

Гапанюк Юрий Евгеньевич

Москва, 2022

# Описание задания

Цель лабораторной работы: изучение возможностей языка Rust.

Задание:

Написать любую программу, используя макросы.

Написать тесты к выполненной работе

Я выбрал написать сервер, обрабатывающий GET запросы с использованием библиотеки Rocket.

# Текст программы

src/main.rs:

#[macro\_use] extern crate rocket;

macro\_rules! calculator {

    ($a:expr, $b:expr, $op:expr) => {

        match $op {

            "+" => $a + $b,

            "-" => $a - $b,

            "\*" => $a \* $b,

            "/" => $a / $b,

            \_ => 0

        }

    };

}

macro\_rules! factorial {

    ($n:expr) => {

        {

            let mut result = 1;

            for i in 1..=$n {

                result \*= i;

            }

            result

        }

    };

}

#[get("/hello/<name>/<age>")]

fn hello(name: &str, age: u8) -> String {

    format!("Hello, {} year old named {}!", age, name)

}

#[get("/?<a>&<b>&<op>")]

fn calc(a: i32, b: i32, op: &str) -> String {

    format!("{} {} {} = {}", a, op, b, calculator!(a, b, op))

}

#[get("/fact/<n>")]

fn fact(n: u32) -> String {

    format!("{}! = {}", n, factorial!(n))

}

#[launch]

fn rocket() -> \_ {

    rocket::build().mount("/", routes![hello, calc, fact])

}

#[cfg(test)]

mod tests {

    use super::\*;

    use rocket::local::blocking::Client;

    #[test]

    fn test\_hello() {

        let client = Client::tracked(rocket()).expect("valid rocket instance");

        let response = client.get("/hello/John/20").dispatch();

        assert\_eq!(response.status(), rocket::http::Status::Ok);

        assert\_eq!(response.into\_string(), Some("Hello, 20 year old named John!".into()));

    }

    #[test]

    fn test\_calc() {

        let client = Client::tracked(rocket()).expect("valid rocket instance");

        let response = client.get("/?a=10&b=5&op=\*").dispatch();

        assert\_eq!(response.status(), rocket::http::Status::Ok);

        assert\_eq!(response.into\_string(), Some("10 \* 5 = 50".into()));

    }

    #[test]

    fn test\_fact() {

        let client = Client::tracked(rocket()).expect("valid rocket instance");

        let response = client.get("/fact/5").dispatch();

        assert\_eq!(response.status(), rocket::http::Status::Ok);

        assert\_eq!(response.into\_string(), Some("5! = 120".into()));

    }

}

# Экранные формы с примерами выполнения программы



