**Лабораторна робота №2 Калькулятор**

**Виконавець: Гончаренко Микита ТВ-12**

[honcharenkko/culcrust (github.com)](https://github.com/honcharenkko/culcrust)

**Для створення базової програми калькулятор нам треба спочатку створити новий проект**

**Пишемо команду в терміналі**

cargo new calculator

**Потім треба написати саму програму**

use std::io::{self, Write}; // Для обробки вводу/виводу

use std::str::FromStr; // Для перетворення рядків у числа

// Перелік доступних операцій

enum Operation {

Add,

Subtract,

Multiply,

Divide,

}

// Функція для вибору операції

fn choose\_operation(op: &str) -> Result<Operation, &'static str> {

match op {

"+" => Ok(Operation::Add),

"-" => Ok(Operation::Subtract),

"\*" => Ok(Operation::Multiply),

"/" => Ok(Operation::Divide),

\_ => Err("Невідома операція"),

}

}

// Основна функція для виконання обчислень

fn calculate(op: Operation, num1: f64, num2: f64) -> Result<f64, &'static str> {

match op {

Operation::Add => Ok(num1 + num2),

Operation::Subtract => Ok(num1 - num2),

Operation::Multiply => Ok(num1 \* num2),

Operation::Divide => {

if num2 != 0.0 {

Ok(num1 / num2)

} else {

Err("Помилка: ділення на нуль")

}

}

}

}

// Функція для зчитування числа

fn read\_number(prompt: &str) -> f64 {

loop {

print!("{}", prompt);

io::stdout().flush().unwrap(); // Очищення буфера

let mut input = String::new();

io::stdin().read\_line(&mut input).unwrap();

match f64::from\_str(input.trim()) {

Ok(num) => return num,

Err(\_) => println!("Помилка: введіть коректне число."),

}

}

}

fn main() {

let mut memory = 0.0; // Для зберігання результату

loop {

println!("\nПоточний результат: {}", memory);

// Зчитування першого числа

let num1 = read\_number("Введіть перше число (або 'm' для використання результату): ");

// Зчитування операції

let mut operation = String::new();

print!("Введіть операцію (+, -, \*, /): ");

io::stdout().flush().unwrap();

io::stdin().read\_line(&mut operation).unwrap();

let operation = match choose\_operation(operation.trim()) {

Ok(op) => op,

Err(err) => {

println!("{}", err);

continue;

}

};

// Зчитування другого числа

let num2 = read\_number("Введіть друге число: ");

// Виконання обчислення

match calculate(operation, num1, num2) {

Ok(result) => {

println!("Результат: {}", result);

memory = result; // Збереження результату в пам'ять

}

Err(err) => println!("{}", err),

}

// Пропозиція завершити або продовжити

println!("Продовжити? (y/n): ");

let mut choice = String::new();

io::stdin().read\_line(&mut choice).unwrap();

if choice.trim() != "y" {

break;

}

}

}

**Після цього треба запустити проект такою функцією в терміналі**

cargo run

Приклад роботи

