

Разработка на языке программирования Rust.

Выполнил: Беликов Константин

Группа: ИУ5-36Б

Дата: 20.11.24г.

Описание задания:

1. По заданию, программа должна реализовывать макрос, заменяющий перегрузку одной из операций $+ - * /$ для структуры `Point`
2. Во второй части нужно было реализовать макрос, который выполняет ту же задачу, что и первый, но способен при этом перегружать сразу любое количество операций $(+ - * /)$

Код программы:

```
#[derive(Debug, Copy, Clone)]
struct Point {
    x: i32,
    y: i32,
}

//Реализация макроса для перегрузки операторов +-*/
macro_rules! override_op {
    ($tr:ident, $func:ident) => (
        impl std::ops::$tr for Point {
            type Output = Self;
            fn $func(self, _rhs: Self) -> Self {
                let _x = std::ops::$tr::$func(self.x, _rhs.x);
                let _y = std::ops::$tr::$func(self.y, _rhs.y);
                Self {x: _x, y: _y}
            }
        }
    )
}

//Реализация макроса, который применяет override_op для произвольного количества
аргументов
macro_rules! override_ops {
    ($tr:ident, $func:ident) => (
        override_op!($tr, $func);
    );

    ($tr:ident, $func:ident, $($coms:ident),+) => (
        override_op!($tr, $func);
        override_ops!($($coms),+);
    )
}

override_ops!(Add, add, Sub, sub, Mul, mul, Div, div);

fn main() {
    let p1 = Point{x: 10, y: 15};
```

```
let p2 = Point{x: 5, y: 3};
let points: [Point; 4] = [p1 + p2, p1 - p2, p1 * p2, p1 / p2];

//Выводим результат применения операций +-* / над точками p1, p2
for p in points.iter() {
    println!("{}", p);
}
}
```

Снимки экрана:

```
PS C:\Users\Konstantin\Documents\VS_Code\Rust\labs\lab3> rustc lab3.rs ; ./lab3.exe
Point { x: 15, y: 18 }
Point { x: 5, y: 12 }
Point { x: 50, y: 45 }
Point { x: 2, y: 5 }
PS C:\Users\Konstantin\Documents\VS_Code\Rust\labs\lab3> █
```