**Разработка на языке программирования Rust.**

Выполнил: Беликов Константин  
Группа: ИУ5-36Б  
Дата: 16.12.24г.

Описание задания:

Задание состояло в том, чтобы продемонстрировать функциональные возможности языка rust:

1. Чётные числа: метод filter для выбора только чётных чисел из списка. Итераторы позволяют обработать данные функционально без явных циклов.
2. Квадраты чисел: С помощью map каждый элемент списка преобразуется в его квадрат.
3. Каррирование: Реализуется замыкание, которое принимает один аргумент (factor) и возвращает новую функцию, умножающую числа на этот фактор. Это демонстрирует идею частичного применения функций.
4. Суммирование: Метод reduce используется для аккумулирования значений. Функция суммирует все элементы списка.

Код программы:

fn main() {

    let numbers = vec![1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

    //Используем итератор и фильтр, чтобы выбрать только чётные числа

    let even\_numbers: Vec<i32> = numbers

        .iter()

        .filter(|&x| x % 2 == 0)

        .cloned()

        .collect();

    println!("Чётные числа: {:?}", even\_numbers);

    //Используем map для преобразования чисел: возводим каждое в квадрат

    let squared\_numbers: Vec<i32> = numbers

        .iter()

        .map(|x| x \* x)

        .collect();

    println!("Квадраты чисел: {:?}", squared\_numbers);

    //Реализация функции с каррированием

    let multiply\_by = |factor: i32| move |x: i32| x \* factor;

    //Создаём функции, которые умножают на 2 и на 3

    let multiply\_by\_2 = multiply\_by(2);

    let multiply\_by\_3 = multiply\_by(3);

    let doubled\_numbers: Vec<i32> = numbers.iter().map(|&x| multiply\_by\_2(x)).collect();

    let tripled\_numbers: Vec<i32> = numbers.iter().map(|&x| multiply\_by\_3(x)).collect();

    println!("Числа, умноженные на 2: {:?}", doubled\_numbers);

    println!("Числа, умноженные на 3: {:?}", tripled\_numbers);

    //Используем reduce для суммирования всех чисел

    let sum: i32 = numbers.iter().copied().reduce(|acc, x| acc + x).unwrap\_or(0);

    println!("Сумма чисел: {}", sum);

}

Снимки экрана:  
