



ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Системы обработки информации и управления» (ИУ-5)

## ЛАБОРАТОРНАЯ РАБОТА

№ 6

Разработка на языке программирования Rust

Группа ИУ5-35Б

Студент А. А. Торопыгин 16.12.2024 /А. А. Торопыгин/

(Подпись, дата)

(И.О.Фамилия)

Преподаватель \_\_\_\_\_ /Ю. Е. Гапанюк/

(Подпись, дата)

(И.О.Фамилия)

## Задание:

1. Реализовать задачу 7 из ЛП №3-4 на языке Rust:
  - В файле [data\\_light.json](#) содержится фрагмент списка вакансий.
  - Структура данных представляет собой список словарей с множеством полей: название работы, место, уровень зарплаты и т.д.
  - Необходимо реализовать 4 функции - f1, f2, f3, f4. Каждая функция вызывается, принимая на вход результат работы предыдущей.
  - Функция f1 должна вернуть отсортированный список профессий без повторений (строки в разном регистре считать равными). Сортировка должна игнорировать регистр.
  - Функция f2 должна фильтровать входной массив и возвращать только те элементы, которые начинаются со слова "программист".
  - Функция f3 должна модифицировать каждый элемент массива, добавив строку "с опытом Python".
  - Функция f4 должна сгенерировать для каждой специальности зарплату от 100 000 до 200 000 рублей и присоединить её к названию специальности.
2. Разработать хотя бы один макрос.
3. Разработать модульные тесты (не менее 3 тестов).

## Текст программы:

```
extern crate rand;

use serde_json::{self, Value};
use std::{fs, ptr::null};
use rand::Rng;

macro_rules! print_vec {
    ($($arg:tt)*) => {
        for i in $($arg)*.into_iter() { println!("{}", i); }
    };
    () => {
        println("It's not a vector!");
    }
}

fn f1(data: Value) -> Vec<String>{
    let mut vec: Vec<String> = Vec::new();
    let mut val: String;

    for i in 0..data.as_array().unwrap().len() {
        if data[i]["job-name"].is_null() {continue;}
        val = data[i]["job-name"].to_string().to_lowercase();
        val.retain(|c| c != '\\');
        vec.push(val);
    }
    vec.sort();
    vec.dedup();
    return vec;
}

fn f2(vec: Vec<String>) -> Vec<String> {
    let prog_jobs = vec
        .into_iter()
        .filter(|job| job.clone().starts_with("программист"))
        .collect::<Vec<String>>();
    return prog_jobs;
}

fn f3(vec: Vec<String>) -> Vec<String> {
    let python_exp = vec
```

```

        .into_iter()
        .map(|job| format!("{}", с опытом Python", job))
        .collect::<Vec<String>>();
    return python_exp;
}

fn f4(vec: Vec<String>) -> Vec<String> {
    let mut salary = rand::thread_rng();
    let rand_salaries = vec
        .into_iter()
        .map(|job| format!("{}", зарплата {} руб.",
            job, salary.gen_range(100000..200000)))
        .collect::<Vec<String>>();
    return rand_salaries;
}

fn main() {
    let str_data = fs::read_to_string("data_light.json")
        .unwrap_or_else(|_error| {panic!("Can't read file")});
    let data: Value = serde_json::from_str(&str_data).expect("Can't parse
json");
    print_vec!(f4(f3(f2(f1(data)))));
}

#[cfg(test)]
mod tests {

    use serde_json::json;

    use super::*;

    #[test]
    fn test_f1() {
        let t = json!([
            {"salary": 100, "job-name": "Я"},
            {"salary": 100, "job-name": "А"},
            {"job-name": "П"},
            {"salary": 100}
        ]);
        assert_eq!(f1(t), vec![
            "a".to_string(),

```

```
        "п".to_string(),
        "я".to_string()
    });
}

#[test]
fn test_f2() {
    assert_eq!(f2(vec!["программист".to_string()], "не
программист".to_string())),
               vec!["программист".to_string()]);
}

#[test]
fn test_f3() {
    assert_eq!(f3(vec!["qwerty".to_string()], "".to_string()),
               vec!["qwerty с опытом Python".to_string()], "с опытом
Python".to_string()))
}
}
```

## Вывод:

```
• andrey@andrey-HP-EliteBook-840-G5:~/BKIT/lab_6$ cargo run
  Compiling lab2_rust v0.1.0 (/home/andrey/BKIT/lab_6)
  Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.36s
  Running `target/debug/lab2_rust`
программист с опытом Python, зарплата 157422 руб.
программист / senior developer с опытом Python, зарплата 174885 руб.
программист 1c с опытом Python, зарплата 141844 руб.
программист c# с опытом Python, зарплата 127887 руб.
программист c++ с опытом Python, зарплата 168809 руб.
программист c++/c#/java с опытом Python, зарплата 196068 руб.
программист/ junior developer с опытом Python, зарплата 176741 руб.
программист/ технический специалист с опытом Python, зарплата 170806 руб.
программистр-разработчик информационных систем с опытом Python, зарплата 171455 руб.
```

```
• andrey@andrey-HP-EliteBook-840-G5:~/BKIT/lab_6$ cargo test
  Compiling lab2_rust v0.1.0 (/home/andrey/BKIT/lab_6)
  Finished `test` profile [unoptimized + debuginfo] target(s) in 0.35s
  Running unittests src/main.rs (target/debug/deps/lab2_rust-59dc4952cc0b324a)

running 3 tests
test tests::test_f2 ... ok
test tests::test_f3 ... ok
test tests::test_f1 ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```