

**Московский Государственный Технический  
Университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчёт по лабораторной работе №4  
«Функциональные возможности языка Python (исполнение задачи 7 на Rust)»

Выполнил:  
студент группы ИУ5-256  
Бикматов Д. А.

Проверил:  
преподаватель  
Гапанюк Ю. В.

Москва, 2024 г.

## Постановка задачи:

Выполнить следующее задание на ЯП 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. За счет декоратора `@print_result` печатается результат, а контекстный менеджер `cm_timer_1` выводит время работы цепочки функций.
- Предполагается, что функции `f1`, `f2`, `f3` будут реализованы в одну строку. В реализации функции `f4` может быть до 3 строк.
- Функция `f1` должна вывести отсортированный список профессий без повторений (строки в разном регистре считать равными). Сортировка должна игнорировать регистр. Используйте наработки из предыдущих задач.
- Функция `f2` должна фильтровать входной массив и возвращать только те элементы, которые начинаются со слова “программист”. Для фильтрации используйте функцию `filter`.
- Функция `f3` должна модифицировать каждый элемент массива, добавив строку “с опытом Python” (все программисты должны быть знакомы с Python). Пример: Программист C# с опытом Python. Для модификации используйте функцию `map`.
- Функция `f4` должна сгенерировать для каждой специальности зарплату от 100 000 до 200 000 рублей и присоединить её к названию специальности. Пример: Программист C# с опытом Python, зарплата 137287 руб. Используйте `zip` для обработки пары специальность — зарплата.

## Текст программы:

main.rs

```
use serde_json::Value;
use rand::Rng;
use std::fs::File;
use std::io::Read;
use std::io::Error;
use logger_macros::log_duration;

fn read_file(filename: &str) -> Result<String, Error> {
    let mut file = match File::open(&filename) {
        Ok(f) => f,
        Err(e) => return Err(e),
    };
};
```

```

let mut text = String::new();
match file.read_to_string(&mut text) {
    Ok(_) => Ok(text),
    Err(e) => Err(e),
}
}

```

```

fn f1(profs: &Vec<Value>) -> Vec<String>{
    let mut professions: Vec<String> = profs.iter().map(|v| v["job-
name"].to_string().to_lowercase().replace("\"", "")).collect::<Vec<_>>();
    professions.sort();
    professions.dedup();
    professions
}

```

```

#[log_duration]
fn f2(profs: Vec<String>) -> Vec<String>{
    profs.into_iter().filter(|v| v.starts_with("программист")).collect::<Vec<_>>()
}

```

```

#[log_duration]
fn f3(profs: Vec<String>) -> Vec<String>{
    profs.into_iter().map(|v| v+" с опытом Python").collect::<Vec<_>>()
}

```

```

#[log_duration]
fn f4(profs: Vec<String>) -> Vec<String>{
    let mut rng = rand::thread_rng();
    let salaries = (0..profs.len()).map(|_|
rng.gen_range(100000..200000)).collect::<Vec<_>>();
    profs.into_iter().zip(salaries).map(|(v, u)| format!("{}", зарплата {} руб.", v,
u)).collect()
}

```

```

fn main() {
    let data = read_file("/Users/hunter/sem3/lab2_rust/data_light.json").expect("Invalid
file");
    let values: Vec<Value> = serde_json::from_str(&data).expect("Invalid JSON");
    let u: Vec<String> = f4(f3(f2(f1(&values))));
    for i in 0..u.len(){
        println!("Job name: {}", u[i]);
    }
}

```

logger\_macros/lib.rs

```
use proc_macro::TokenStream;
use quote::quote;
use syn::{parse_macro_input, ItemFn};

#[proc_macro_attribute]
pub fn log_duration(args: TokenStream, item: TokenStream) -> TokenStream {
    log_duration_impl(args, item)
}

fn log_duration_impl(_args: TokenStream, input: TokenStream) -> TokenStream {
    let input = parse_macro_input!(input as ItemFn);
    let ItemFn {
        sig,
        vis,
        block,
        attrs,
    } = input;

    let statements = block.stmts;
    let function_identifier = sig.ident.clone();

    quote!(
        #(#attrs)*
        #vis #sig {
            let __start = std::time::Instant::now();
            let __result = {
                #(#statements)*
            };

            println!("{}", " {} - {}, dump:", stringify!(&function_identifier),
__start.elapsed().as_micros());
            for s in &__result {
                println!("{}", " {} ", s);
            }
            println!("{}", "");
            return __result;
        }
    )
    .into()
}
```

logger\_macros/cargo.toml

```
[package]
name = "logger_macros"
version = "0.1.0"
edition = "2021"
```

```
[lib]
name = "logger_macros"
path = "src/lib.rs"
proc-macro = true
```

```
[dependencies]
darling = "0.20.10"
proc-macro2 = "1.0.89"
syn = "2.0.87"
quote = «1.0.37"
```

cargo.toml

```
[package]
name = "process_data"
version = "0.1.0"
edition = "2021"
```

```
[dependencies]
serde_json = "1.0"
rand = "0.8.4"
logger_macros = { path = "../logger_macros" }
```

## Пример выполнения:

```
hunter@MacBook-Pro-Andrey debug % ./process_data
f2 - 477, dump:
программист
программист / senior developer
программист 1с
программист с#
программист с++
программист с++/с#/java
программист/ junior developer
программист/ технический специалист
программист-разработчик информационных систем

f3 - 5, dump:
программист с опытом Python
программист / senior developer с опытом Python
программист 1с с опытом Python
программист с# с опытом Python
программист с++ с опытом Python
программист с++/с#/java с опытом Python
программист/ junior developer с опытом Python
программист/ технический специалист с опытом Python
программист-разработчик информационных систем с опытом Python

f4 - 137, dump:
программист с опытом Python, зарплата 119583 руб.
программист / senior developer с опытом Python, зарплата 194338 руб.
программист 1с с опытом Python, зарплата 115729 руб.
программист с# с опытом Python, зарплата 117679 руб.
программист с++ с опытом Python, зарплата 128956 руб.
программист с++/с#/java с опытом Python, зарплата 166806 руб.
программист/ junior developer с опытом Python, зарплата 184463 руб.
программист/ технический специалист с опытом Python, зарплата 183929 руб.
программист-разработчик информационных систем с опытом Python, зарплата 148005 руб.

Job name: программист с опытом Python, зарплата 119583 руб.
Job name: программист / senior developer с опытом Python, зарплата 194338 руб.
Job name: программист 1с с опытом Python, зарплата 115729 руб.
Job name: программист с# с опытом Python, зарплата 117679 руб.
Job name: программист с++ с опытом Python, зарплата 128956 руб.
Job name: программист с++/с#/java с опытом Python, зарплата 166806 руб.
Job name: программист/ junior developer с опытом Python, зарплата 184463 руб.
Job name: программист/ технический специалист с опытом Python, зарплата 183929 руб.
Job name: программист-разработчик информационных систем с опытом Python, зарплата 148005 руб.
```

Вывод с функции f1 опущен, так как он слишком огромен, чтобы поместиться в этот отчёт