

Создание счетчика слов на языке программирования Rust

Саяпин Никита Анатольевич

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

Цель статьи — разработка программы на языке Rust для подсчёта количества слов в тексте. Реализация проекта выполняется в среде разработки Cargo и позволяет пользователю вводить строку текста, после чего программа возвращает число слов. Работа демонстрирует основы синтаксиса Rust, работу с переменными, стандартной библиотекой и строками.

Ключевые слова: Rust, счётчик слов, Cargo, программирование, текстовые данные.

Creating a Word Counter in Rust

Sayapin Nikita Anatolyevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This paper presents the development of a word counter in Rust. The project is implemented using the Cargo build system and allows the user to input a text string, then outputs the number of words. The work demonstrates basic Rust syntax, handling of variables, the standard library, and string operations.

Keywords: Rust, word counter, Cargo, programming, text data.

1. Введение

1.1 Актуальность

Современные языки программирования предоставляют широкий инструментарий для обработки текстовой информации. Rust — молодой и быстро развивающийся язык, который сочетает безопасность, скорость и современный синтаксис. Создание простого счётчика слов является базовой задачей обработки текста и может использоваться как учебный пример для знакомства с Rust.

1.2 Обзор исследований

В статье И.С. Александрова представлено описание разработки программного обеспечения для моделирования траектории полёта объекта на языке программирования Rust с учётом изменяющихся параметров. Описана реализация алгоритмов расчёта траектории, проверки пересечений с регионами и вычисления ускорения с учётом изменяющихся параметров,

таких как лобовое сопротивление и гравитационное ускорение [1]. К.С. Ткаченко рассматривает программы на языках высокого уровня Swift и Rust для решения классической задачи о построении расписания обслуживания для двух устройств. Приводятся полные исходные тексты программ и результаты работы [2]. Статья С.О. Иванова посвящена исследованию влияния языка программирования на эффективность и надежность программного обеспечения [3]. Р.Г. Смирнов разработал программу, суть которой – изменение представления изображения и его декомпозиция до компонентов, которые возможно эффективно анализировать с помощью инструментария программирования. [4].

1.3 Цель исследования

Целью исследования является разработка и анализ простой программы на языке Rust, подсчитывающей количество слов во вводимом пользователем тексте. Под «словом» в рамках данной работы понимается непрерывная последовательность непробельных символов, выделяемая стандартной функцией `split_whitespace()`. Такой подход позволяет абстрагироваться от особенностей пунктуации и быстро получить надёжный результат на разнородных данных.

Объектом исследования выступает текстовая строка произвольной длины, поступающая на стандартный ввод программы. Предмет исследования – алгоритм подсчёта слов и особенности его реализации на Rust с использованием стандартной библиотеки.

Ожидаемый результат — консольная утилита, корректно подсчитывающая слова в реалистичных вводах (латиница/кириллица, несколько пробелов, переносы строк). Дополнительно рассматривается организация проекта с помощью Cargo и воспроизводимый процесс сборки/запуска для последующего расширения функциональности.

2 Материалы и методы

Для реализации проекта использовался язык программирования Rust версии 1.89.0, а также система сборки и управления проектами Cargo, которая входит в стандартный дистрибутив Rust. Работа велась в среде разработки Visual Studio Code с установленным плагином `rust-analyzer`, обеспечивающим подсветку синтаксиса, автодополнение и запуск кода.

2.1 Подготовка

На первом этапе устанавливается официальный дистрибутив Rust через менеджера `rustup`, который включает компилятор `rustc`, систему сборки `cargo` и документацию. Установку можно выполнить с страницы `rustup` или из терминала командой `curl | sh` (Windows — через `rustup-init.exe`). После завершения установки проверяется доступность инструментов командами `rustc --version` и `cargo --version`; успешный вывод версий подтверждает корректную настройку окружения (Рис.1).

```
C:\Users\Nktt>rustc --version
rustc 1.89.0 (29483883e 2025-08-04)

C:\Users\Nktt>cargo --version
cargo 1.89.0 (c24e10642 2025-06-23)

C:\Users\Nktt>_
```

Рисунок 1 – Вывод о успешной установке языка

Следующий шаг — создание нового бинарного проекта с помощью Cargo. В терминале выбирается рабочая папка, после чего выполняется команда `cargo new word_counter --bin`.

Cargo создаёт одноимённый каталог с минимальной структурой проекта, инициализирует git (если доступен) и подготавливает файл манифеста `Cargo.toml`, а также исходник `src/main.rs` с шаблонной функцией `main()`. Переходим в каталог проекта (`cd word_counter`) и убеждаемся, что структура соответствует ожидаемой: корневой `Cargo.toml` и папка `src` с файлом `main.rs` (Рис.2).

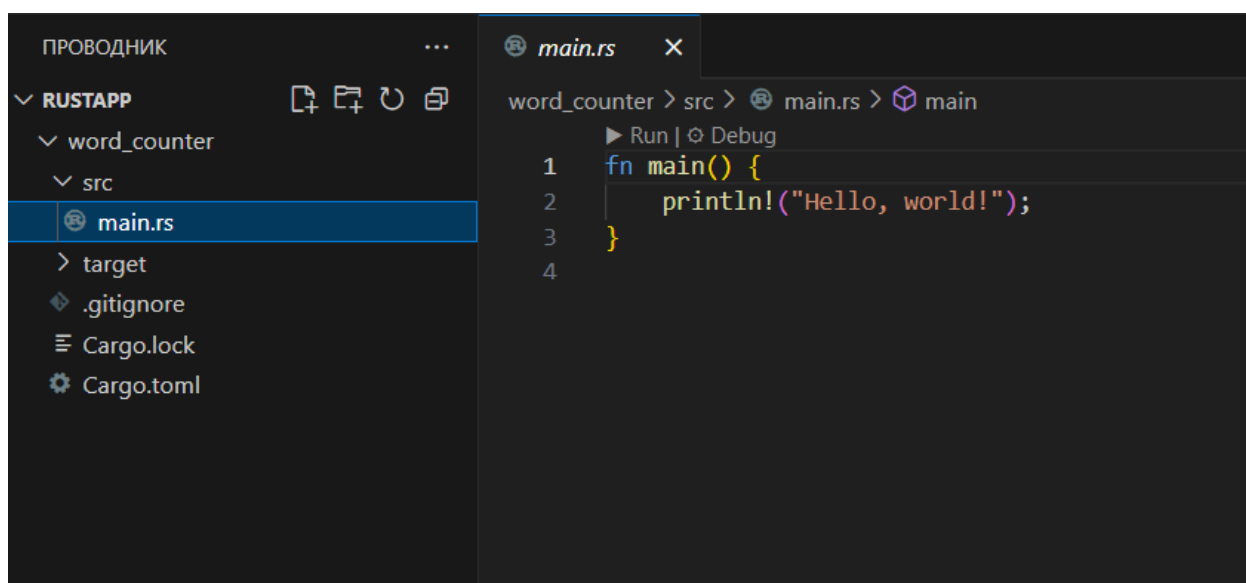
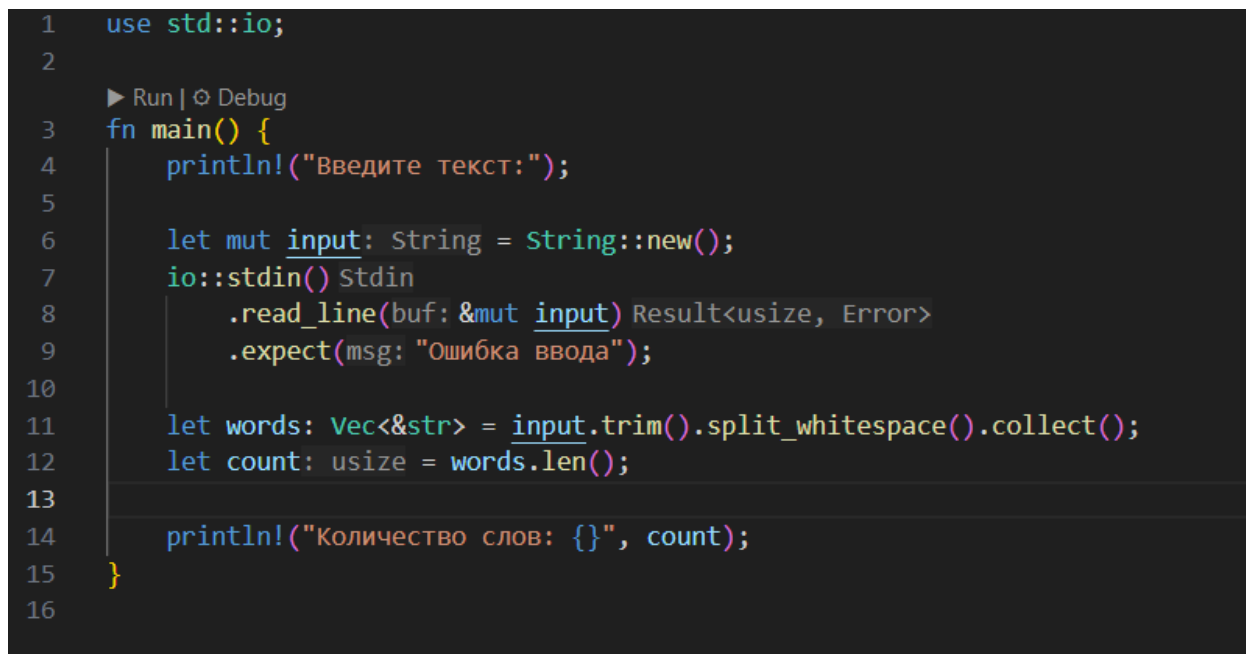


Рисунок 2 – Директория проекта

Для удобства разработки проект открывается в редакторе Visual Studio Code. Рекомендуется установить расширение `rust-analyzer`, обеспечивающее подсветку синтаксиса, автодополнение, переход к определению и быстрый запуск задач Cargo. Открыв папку проекта в VS Code, появится дерево файлов и готовый для модификации `main.rs`.

3. Результаты

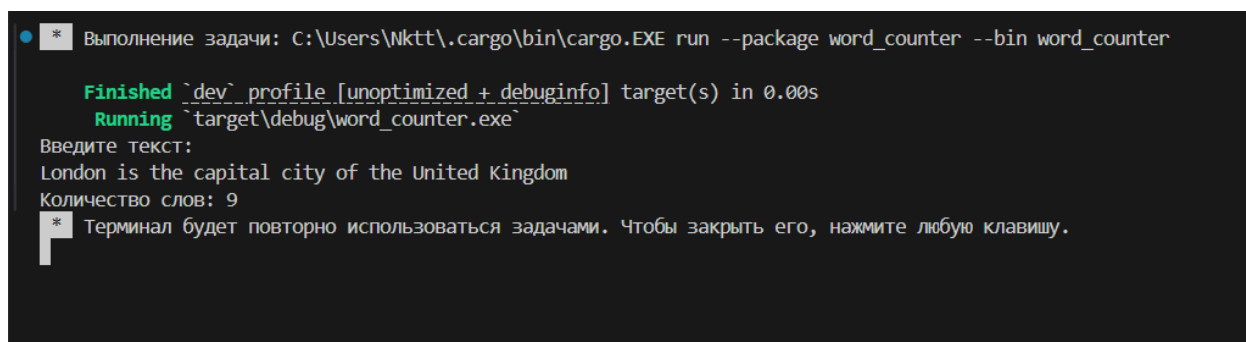
Можно переходить к объявлению и инициализации основных переменных для реализации программы счётчика слов. Используется стандартный модуль `std::io` для чтения строки с клавиатуры, а также методы работы со строками для разделения текста по пробелам (Рис. 2).



```
1 use std::io;
2
3 ▶ Run | ⌕ Debug
4 fn main() {
5     println!("Введите текст:");
6
7     let mut input: String = String::new();
8     io::stdin().stdin
9         .read_line(buf: &mut input) Result<usize, Error>
10        .expect(msg: "Ошибка ввода");
11
12    let words: Vec<&str> = input.trim().split_whitespace().collect();
13    let count: usize = words.len();
14
15    println!("Количество слов: {}", count);
16 }
```

Рисунок 2 – Полный код программы

Программа сначала выводит приглашение для пользователя, затем считывает введённую строку и сохраняет её в переменную `input`. Метод `trim()` удаляет лишние пробелы по краям строки. Функция `split_whitespace()` разбивает строку на отдельные слова, используя пробелы и символы переноса строки как разделители. Полученный результат сохраняется в вектор `words`. После этого определяется длина вектора с помощью `len()`, которая и является количеством слов в тексте (Рис. 3).



```
• * Выполнение задачи: C:\Users\Nktt\.cargo\bin\cargo.EXE run --package word_counter --bin word_counter

Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.00s
Running `target\debug\word_counter.exe`
Введите текст:
London is the capital city of the United Kingdom
Количество слов: 9
* Терминал будет повторно использоваться задачами. Чтобы закрыть его, нажмите любую клавишу.
```

Рисунок 3 – Пример работы программы

4 Выводы

Разработанная программа успешно выполняет подсчёт слов в тексте. Она корректно обрабатывает строки с пробелами и символами переноса.

Использование стандартных функций Rust позволило реализовать задачу в минимальном объёме кода и с высокой читаемостью.

Библиографический список

1. Рахмани Д., Александров И.С. Разработка программы на языке Rust для моделирования траектории полёта объекта с учётом изменяющихся параметров // Международный журнал гуманитарных и естественных наук. 2025. № 3-1(102). С. 242-252.
2. Ткаченко К. С. Разработка программ на языках Swift и Rust для решения классической задачи о построении расписания обслуживания для двух устройств // Системный администратор. 2023. № 10(251). С. 90-92.
3. Иванов С. О., Ильин Д.В., Большаков И.Ю. Сравнительное тестирование языков программирования // Вестник Чувашского университета. 2017. № 3. С. 222-227.
4. Свидетельство о государственной регистрации программы для ЭВМ №2022611439 Российская Федерация. Программа для точной сегментации объектов на RUST № 2022610407 : заявл. 14.01.2022 : опубл. 25.01.2022 / Смирнов Р. Г.; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова».