

Генератор случайных паролей на языке Rust

Кокухина Софья Александровна

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

Цель исследования — разработка программы на языке Rust, генерирующей случайные пароли с использованием библиотеки rand. В ходе работы показаны основы работы со строками, генерацией случайных чисел, коллекциями и функциями. Программа позволяет задавать длину пароля и формировать его из набора символов: букв, цифр и специальных знаков. Работа демонстрирует практическое применение Rust для решения задач в области информационной безопасности

Ключевые слова Rust, генератор паролей, rand, программирование, безопасность.

Password generator in Rust

Kokukhina Sofya Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article presents the development of a Rust program for generating random passwords using the rand library. The project demonstrates basic operations with strings, random number generation, collections, and functions. The program allows the user to specify the password length and generates a password consisting of letters, digits, and special characters. This work illustrates practical applications of Rust in the field of information security.

Keywords Rust, password generator, rand, programming, security.

1. Введение

1.1 Актуальность

Информационная безопасность напрямую зависит от надёжности паролей, используемых пользователями. Простые пароли легко подбираются злоумышленниками, поэтому для защиты данных необходимо использовать случайно сгенерированные комбинации символов. Создание генератора паролей на языке Rust является полезной учебной задачей: с одной стороны, она знакомит студентов с основами работы со строками и случайными числами, с другой — демонстрирует важность защиты информации.

1.2 Обзор исследований

В статье Д.Л. Буй и М.М. Сидякина рассматривается возможность использования языка программирования Rust для разработки программного обеспечения встраиваемых систем. Выявлены преимущества и недостатки использования Rust в контексте встраиваемых систем. Определены ключевые особенности языка, связанные с безопасностью, производительностью и поддержкой аппаратно-ориентированного программирования [1]. П.А.Агафонов и М.С. Асташкин рассказали, что большинство часто используемых языков имеют сборщик мусора, который самостоятельно освобождает память неиспользуемого объекта. Такой способ имеет недостаток - уменьшение производительности приложения. Язык Rust имеет безопасный и эффективный способ управления памятью, основанный на аффинных типах [2]. Статья Д.М. Берлизова посвящена реализации программной транзакционной памяти для языка программирования rust [3]. Д.А. Смирнов рассматривает необходимость использования новых языков программирования в современной разработке программного обеспечения, освещается важность новых языков для решения сложных задач, предлагаются улучшенные инструменты и библиотеки [4].

1.3 Цель исследования

Целью исследования является реализация программы на языке Rust, которая будет генерировать случайные пароли заданной длины, используя библиотеку rand и коллекции символов.

2 Материалы и методы

Для реализации проекта используется язык Rust версии 1.79.0 и система сборки Cargo, входящая в стандартный дистрибутив Rust. В качестве среды разработки применяется Visual Studio Code с плагином rust-analyzer, который обеспечивает автодополнение и подсветку синтаксиса.

Перед началом работы создается новый проект командой cargo new password_generator --bin (Рис.1)

```
C:\Users\Nktt\Documents\Учеба\RustApp>cargo new password_generator --bin
  Creating binary (application) `password_generator` package
note: see more `Cargo.toml` keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html
C:\Users\Nktt\Documents\Учеба\RustApp>_
```

Рисунок 1 – Создание проекта из консоли

Затем в файле Cargo.toml добавлялась зависимость от библиотеки rand (Рис.2)

```
password_generator > Cargo.toml
1  [package]
2  name = "password_generator"
3  version = "0.1.0"
4  edition = "2024"
5
6  [dependencies]
7  rand = "0.8"
```

Рисунок 2 – Добавление зависимости

3 Результаты и обсуждения

Теперь стоит перейти к объявлению и инициализации основных переменных. Создаются строки с наборами символов: латинские буквы, цифры и специальные символы. Также используется вектор для объединения этих наборов в единый пул символов (Рис. 3).

```
main.rs • Cargo.toml
password_generator > src > main.rs > ...
1  const LETTERS: &str = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
2  const DIGITS: &str = "0123456789";
3  const SYMBOLS: &str = "!@#$%^&*()-_+=<>?";
4
5  let all_chars = format!("{}", LETTERS, DIGITS, SYMBOLS);
6  |
```

Рисунок 3 – Объявление переменных программы

Далее реализуется функция `generate_password`, которая принимает параметр длины пароля и возвращает строку сгенерированного значения. Внутри функции используется метод `choose` из библиотеки `rand` для выбора случайных символов из общего пула (Рис. 4).

```
fn generate_password(length: usize, chars: &str) -> String {
    let mut rng = rand::thread_rng();
    let chars_vec: Vec<char> = chars.chars().collect();
    (0..length)
        .map(|_| *chars_vec.choose(&mut rng).unwrap())
        .collect()
}
```

Рисунок 4 – Функция генерации пароля

В основной функции `main` организуется ввод длины пароля от пользователя. Программа вызывает функцию генерации и выводит результат (Рис. 5).

```
fn main() {
    println!("Введите длину пароля:");
    let mut input = String::new();
    io::stdin().read_line(&mut input).expect("Ошибка ввода");
    let length: usize = input.trim().parse().expect("Введите число!");

    let all_chars = format!("{}", LETTERS, DIGITS, SYMBOLS);
    let password = generate_password(length, &all_chars);

    println!("Сгенерированный пароль: {}", password);
}
```

Рисунок 5 – Основной код программы

После завершения кода проект запускается командой `cargo run`, в консоли программа запрашивает длину пароля, после чего выводит результат (Рис.6).

```
password(C:\Users\Nktt\Documents\Учеба\RustApp\password_generator>cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.01s
пароль: Running `target\debug\password_generator.exe`
Введите длину пароля:
5
Сгенерированный пароль: n)F4i

C:\Users\Nktt\Documents\Учеба\RustApp\password_generator>cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.03s
Running `target\debug\password_generator.exe`
Введите длину пароля:
24
Сгенерированный пароль: 9?S+*W1d?@q0$zi-4Yhv4=ZF
```

Рисунок 6 – Пример работы программы

3

Выводы. В ходе работы реализован генератор случайных паролей на языке Rust. Проект позволил закрепить навыки работы со строками, коллекциями и внешними библиотеками. Использование `rand` обеспечило простоту реализации и высокое качество случайности. Данный проект может быть расширен добавлением графического интерфейса или настройкой параметров (например, исключение определённых символов).

Разработанная программа успешно генерирует случайные пароли указанной длины. Она корректно обрабатывает ввод пользователя и формирует строку, включающую буквы, цифры и специальные символы. Полученные пароли являются случайными и могут использоваться для повышения уровня информационной безопасности.

Библиографический список

1. Буй Д. Л., Сидякин И. М. Применение языка программирования Rust для разработки программного обеспечения встраиваемых систем // Наукосфера. 2023. № 10-1. С. 93-98.
2. Агафонов П. А., Асташкин М. С. Сравнение способов управления памятью в различных языках программирования // Информационные технологии в процессе подготовки современного специалиста, межвузовский сборник научных трудов. Выпуск 25. Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского. 2021. С. 8-14.
3. Берлизов Д. М. Реализация программной транзакционной памяти для языка программирования Rust // Материалы 54-й Международной научной студенческой конференции МНСК-2016: Информационные технологии. Новосибирский национальный исследовательский государственный университет. 2016. С. 147.
4. Смирнов Д. А., Баланев К. С., Могиленко Н. В. Необходимость использования новых языков программирования // Сборник статей VII Международной научно-практической конференции. Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.). 2023. С. 38-40.