

**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»

Отчет по домашнему заданию
“Генератор паролей в командной строке на языке Rust.”

Выполнил:
студент группы ИУ5-35Б:
Тенишев А.А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.В.
Подпись и дата:

Москва, 2024 г

Описание задания

Разработка универсального инструмента для генерации безопасных паролей, позволяющего пользователям гибко задавать параметры: длина пароля, минимальное количество различных типов символов, возможность включения или исключения специальных символов, а также генерация нескольких паролей за один раз.

Основные функции:

1. Генерация паролей заданной длины. (параметр -l)
2. Поддержка настройки минимального количества: букв (строчных и прописных) (параметры -s и -u), цифр (параметр -n), специальных символов (параметр -i).
3. Возможность полного исключения специальных символов. (параметр -x)
4. Генерация нескольких паролей одновременно. (параметр -c)

Текст программы.

```
use clap::{Arg, Command}; // Библиотека для обработки аргументов
use rand::{distributions::Alphanumeric, thread_rng, Rng, seq::SliceRandom}; //
Библиотека для псевдорандома

// функция добавления выбранного количества символов в строку
fn add_to_string(input: &str, num_chars: usize, char_set: &str) -> String {
    let mut rng = rand::thread_rng();
    let mut result = input.to_string();

    let char_vec: Vec<char> = char_set.chars().collect();

    for _ in 0..num_chars {
        let random_char = char_vec.choose(&mut rng).unwrap();
        result.push(*random_char);
    }

    result
}

fn main(){
    // Декларируем наборы символов
    let letters = String::from("abcdefghijklmnopqrstuvwxyz");
    let uletters = String::from("ABCDEFGHIJKLMNOPQRSTUVWXYZ");
    let numbers = String::from("0123456789");
    let specials = String::from("!@#$%^&*()-_+=[]{}|;:',.<>~/~`");
    let mut custom = String::from("");
```

```
// работа с cli
let matches = Command::new("Passgen")
    .version("1.0")
    .author("IU5-35B")
    .about("Password generation")
    .arg( // длина пароля
        Arg::new("len")
            .short('l')
            .long("len")
            .takes_value(true)
            .default_value("12")
            .help("Set the password length"),
    )
    .arg( // минимальное количество букв в пароле
        Arg::new("let")
            .short('s')
            .long("let")
            .takes_value(true)
            .default_value("0")
            .help("Minimum number of letters in the password"),
    )
    .arg( // минимальное количество цифр в пароле
        Arg::new("num")
            .short('n')
            .long("num")
            .takes_value(true)
            .default_value("0")
            .help("Minimum number of numbers in the password"),
    )
    .arg( // минимальное количество заглавных букв в пароле
        Arg::new("ulet")
            .short('u')
            .long("ulet")
            .takes_value(true)
            .default_value("0")
            .help("Minimum number of upper case letters in the password"),
    )
    .arg( // отсутствие специальных символов в пароле
        Arg::new("symb")
            .short('i')
            .long("symb")
            .takes_value(true)
            .help("Include special symbols in the password"),
    )
    .arg( // отсутствие специальных символов в пароле
        Arg::new("excl")
            .short('x')
            .long("excl")
            .takes_value(false)
            .help("Exclude special symbols from the password"),
    )
```

```

    )
    .arg( // число сгенерированных паролей
        Arg::new("count")
            .short('c')
            .long("count")
            .takes_value(true)
            .default_value("1")
            .help("Generate multiple passwords"),
    )
    .get_matches();

let length: usize = matches
    .get_one::("len")
    .unwrap()
    .parse()
    .expect("Length must be a number");

let slength: usize = matches
    .get_one::("let")
    .unwrap()
    .parse()
    .expect("Length must be a number");

let nlength: usize = matches
    .get_one::("num")
    .unwrap()
    .parse()
    .expect("Length must be a number");

let uslength: usize = matches
    .get_one::("ulet")
    .unwrap()
    .parse()
    .expect("Length must be a number");

let count: usize = matches
    .get_one::("count")
    .unwrap()
    .parse()
    .expect("Count must be a number");

let mut genpass = String::new();

//цикл для генерации сразу нескольких паролей за раз
let mut i = 0;
while i < count{
    // Добавление символов в порядке, гарантируя минимальное количество
каждого типа
    genpass = add_to_string(&genpass, slength, &letters); // Добавляем буквы
нижнего регистра

```

```

        genpass = add_to_string(&genpass, nlength, &numbers); // Добавляем цифры
        genpass = add_to_string(&genpass, uslength, &uletters); // Добавляем
буквы верхнего регистра

        // Для символов: добавляем оставшуюся часть, если включены символы
        let mut symbols = letters.clone() + &uletters + &numbers;

        if !matches.is_present("excl") {
            symbols = symbols + &specials; // Добавляем специальные символы, если
не исключены
        }

        // Если общее количество символов меньше нужной длины, заполняем
оставшиеся места случайными символами
        if genpass.len() < length {
            let remaining_length = length - genpass.len();
            genpass = add_to_string(&genpass, remaining_length, &symbols);
        }

        // Перемешиваем строку для случайного порядка символов
        let mut rng = thread_rng();
        let mut genpass_vec: Vec<char> = genpass.chars().collect();
        genpass_vec.shuffle(&mut rng);
        genpass = genpass_vec.iter().collect();

        println!("Вариант №{: {}}: {}", i+1, genpass);
        i = i + 1;
    }
}

```

Результаты выполнения.

1 пароль

./target/release/DZ

Вариант №1: Vb0qZ2[i0^d*

10 паролей длиной 10 только из букв верхнего регистра

./target/release/DZ -c 10 -l 10 -u 10

Вариант №1: QQXTTIBPIR

Вариант №2: EXRYURNVUU

Вариант №3: PPDXBSDIEV

Вариант №4: WXQHAYAYOR

Вариант №5: FIHJMFATRV

Вариант №6: XHANPZFANB

Вариант №7: FQRLBHKRTU

Вариант №8: FDFNYZDVGI

Вариант №9: UKBJNWC MAT

Вариант №10: NEQVNOQXIY

3 пароля длиной 12 символов с хотя бы 6 цифрами

./target/release/DZ -c 3 -l 12 -n 6

Вариант №1: q3/0q902^9'6

Вариант №2: 8x*89jpx8256

Вариант №3: K%^ 3292/,5L2

3 пароля длиной 12 символов с хотя бы 6 цифрами без спец символов

./target/release/DZ -c 3 -l 12 -n 6 -x

Вариант №1: 1015VL404w56

Вариант №2: 6b4XrBjv7597

Вариант №3: SYv299JM8P15

7 паролей длиной 9 символов с 3 цифрами, 3 буквами нижнего регистра и 3 буквами верхнего регистра

./target/release/DZ -c 7 -l 9 -n 3 -u 3 -s 3

Вариант №1: 8J2DxKx7k

Вариант №2: Q3Kq3v8nE

Вариант №3: p7k4tX2KX

Вариант №4: ZroH1E9t1

Вариант №5: Yicv0mE09

Вариант №6: 6Cvu9k00L

Вариант №7: 5FP1gk0Rw