

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и
управления

ОТЧЕТ ПО ПРАКТИКЕ

Создать программу для генерации случайных
паролей заданной длины и сложности

Вариант 1

2 курс, группа 2ИВТ1-1

Выполнил:

_____ Д. А. Лиев
«___» _____ 2024 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2024 г.

Майкоп, 2024 г.

1. Введение

1.1. Формулировка цели

Целью данной работы является написание программы для генерации случайных паролей заданной длины и сложности.

1.1.1. Теория

Генерация случайных паролей важна для обеспечения безопасности данных. Сильный пароль должен обладать следующими характеристиками. Во-первых, достаточная длина: чем длиннее пароль, тем сложнее его взломать методом перебора. Во-вторых, сложность: использование различных типов символов (буквы, цифры, специальные символы) увеличивает количество возможных комбинаций и затрудняет взлом.

Основные требования к случайным паролям:

- 1) пароли должны генерироваться случайным образом, чтобы предотвратить предсказуемость;
- 2) для повышения стойкости должны использоваться различные символы;
- 3) пользователь должен иметь возможность задать параметры генерации, такие как длина и сложность пароля.

Для генерации случайных паролей используются псевдослучайные генераторы чисел (PRNG). В C++ доступны стандартные библиотеки для работы с случайными числами и строками, которые можно использовать для этой задачи.

1.1.2. Практика

Рассмотрим шаги по созданию программы для генерации случайных паролей на C++:

- 1) пользователь должен иметь возможность задать длину пароля и выбрать уровень сложности, включая использование цифр, специальных символов, заглавных и строчных букв;
- 2) используются стандартные библиотеки `<random>` для генерации случайных чисел. Создается набор символов в зависимости от выбранного уровня сложности. Генерируется случайная последовательность символов заданной длины;
- 3) программа реализована на языке C++ с использованием консольного ввода/вывода;
- 4) проведено тестирование для проверки корректности генерации паролей разной длины и сложности. Также проверена непредсказуемость и отсутствие повторяемости паролей.

Важно отметить, что программа предоставляет пользователю гибкие настройки для генерации паролей, включая выбор длины и набора используемых символов. Это позволяет создавать пароли разного уровня сложности, удовлетворяющие требованиям безопасности.

В результате проделанной работы была создана программа для генерации случайных паролей на языке C++, отвечающая заданным требованиям по длине и сложности. Программа успешно прошла тестирование и может быть использована для создания безопасных паролей. В будущем возможно расширение функционала программы, включая графический интерфейс и дополнительные параметры настройки паролей.

2. Ход работы

2.1. Программа

Код выполненной программы:

```
1  #include <iostream>
2  #include <string>
3  #include <random>
4  #include <ctime>
5
6  using namespace std;
7
8  string generatePassword(int length, bool useDigits, bool useSpecial, bool
   ↪ useUpper, bool useLower) {
9      const string digits = "0123456789";
10     const string special = "!@#$%^&*()-_+=[]{}|;: '\", . <>? / \\ ";
11     const string upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
12     const string lower = "abcdefghijklmnopqrstuvwxyz";
13
14     string characterPool;
15
16     if (useDigits) characterPool += digits;
17     if (useSpecial) characterPool += special;
18     if (useUpper) characterPool += upper;
19     if (useLower) characterPool += lower;
20
21     if (characterPool.empty()) {
22         throw invalid_argument("Необходимо выбрать хотя бы один тип
   ↪ символов для генерации пароля.");
23     }
24
25     string password;
26     mt19937 generator(static_cast<unsigned>(time(nullptr)));
```

```

27     uniform_int_distribution<> distribution(0, characterPool.size() - 1);
28
29     for (int i = 0; i < length; ++i) {
30         password += characterPool[distribution(generator)];
31     }
32
33     return password;
34 }
35
36 int main() {
37     setlocale(LC_ALL, "Rus");
38     system("chcp 1251");
39
40     int length;
41     string input;
42
43     cout << "Введите длину пароля: ";
44     cin >> length;
45
46     cout << "Использовать цифры? (да/нет): ";
47     cin >> input;
48     bool useDigits = (input == "да");
49
50     cout << "Использовать специальные символы? (да/нет): ";
51     cin >> input;
52     bool useSpecial = (input == "да");
53
54     cout << "Использовать заглавные буквы? (да/нет): ";
55     cin >> input;
56     bool useUpper = (input == "да");
57
58     cout << "Использовать строчные буквы? (да/нет): ";
59     cin >> input;
60     bool useLower = (input == "да");
61
62     try {
63         string password = generatePassword(length, useDigits,
64             ↪ useSpecial, useUpper, useLower);
65         cout << "Сгенерированный пароль: " << password << endl;
66     }
67     catch (const invalid_argument & e) {
68         cerr << e.what() << endl;
69     }
70
71     return 0;
72 }

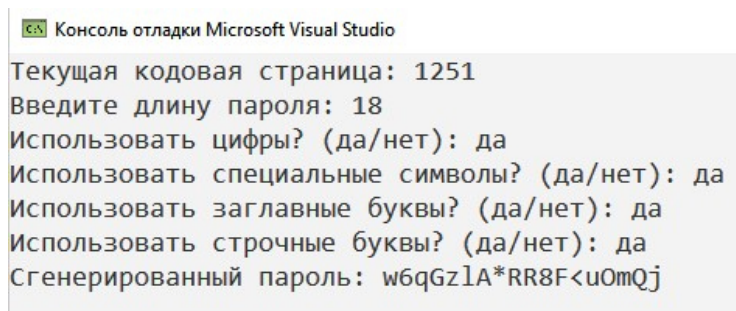
```

2.2. Результат работы программы

Программа работает следующим образом:

- 1) пользователь вводит длину пароля;
- 2) пользователь указывает, будет ли использоваться каждый из следующих типов символов: цифры, специальные символы, заглавные буквы, строчные буквы;
- 3) программа генерирует пароль, используя выбранные типы символов.

Результат работы программы (Рис. 1.):



```
Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Введите длину пароля: 18
Использовать цифры? (да/нет): да
Использовать специальные символы? (да/нет): да
Использовать заглавные буквы? (да/нет): да
Использовать строчные буквы? (да/нет): да
Сгенерированный пароль: w6qGz1A*RR8F<u0mQj
```

Рис. 1. Результат работы программы

Список литературы

- [1] Кнут Д.Э. Всё про $\text{T}_\text{E}\text{X}$. — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ в примерах. 2005 г.