

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ПРИЛОЖЕНИЯ

Чубукина Мария Сергеевна

студент, Поволжский государственный университет телекоммуникаций и информатики, РФ, г. Самара

CUSTOM APPLICATION DEVELOPMENT

Maria Chubukina

Student, Povolzhsky State University of Telecommunications and Informatics, Russia, Samara

Целью работы было разработать надежную систему хранения паролей, организованную в виде набора записей. Каждая запись состоит из названия, имени пользователя, пароля, комментария. Записи группируются по тематическим с определенными названиями (например: пин-коды, интернет, почта, прочее).

Хранитель паролей – приложение, которое помогает пользователю работать с паролями и PIN-кодами.

Хранитель паролей делятся на три основных категории:

Десктоп — хранят пароли к программному обеспечению, установленному на жестком диске компьютера.

Портативные — хранят пароли к программному обеспечению на мобильных устройствах и носителях.

Сетевые — менеджеры паролей онлайн, где пароли сохранены на веб-сайтах.

Порядок создания и сохранения следующий:

1. Авторизация
2. Генерация пароля и ввод в поле
3. Создаем атрибут ,по которому будем искать созданный пароль, и сохраняем пароль
4. Поиск пароля по атрибуту
5. Редактирование/ удаление пароля

Конкретно для написания кода было решено выбрать среду Visual Studio. Visual Studio - полнофункциональная интегрированная среда разработки для Windows, Android, iOS, а также веб-приложений и облачных служб на многих языках программирования, в частности C#, JavaScript, Python, C++ , разработанная компанией Microsoft.

Для графического представления приложения можно использовать WindowsForm. Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework.

C# — простой, современный объектно-ориентированный и безопасный язык программирования. C# принадлежит к известной семье языков C.

Разработанное решение реализовано в двух проектах (см. рис.1): ClassLibrary1 и WindowsFormsApplication2. Проект ClassLibrary1 реализует библиотеку с основными функциями программы. Файл Class1 содержит классы User и Password. Проект WindowsFormsApplication2 представляет собой приложение Windows Forms, обеспечивающее пользовательский интерфейс к разработанным классам. Содержит 5 форм.

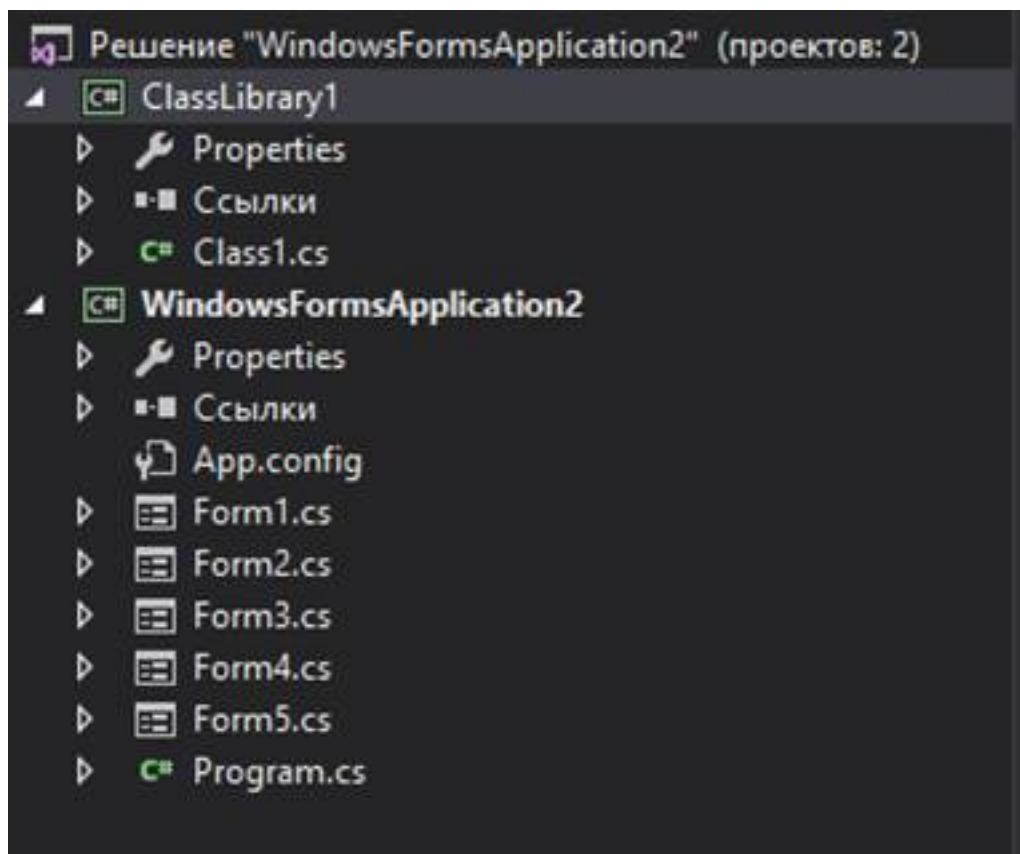


Рисунок 1. Файловая система приложения

Ядро приложения состоит из двух классов:

Класс User - класс, описывающий пользователя

Класс Password- класс для создания и хранения паролей

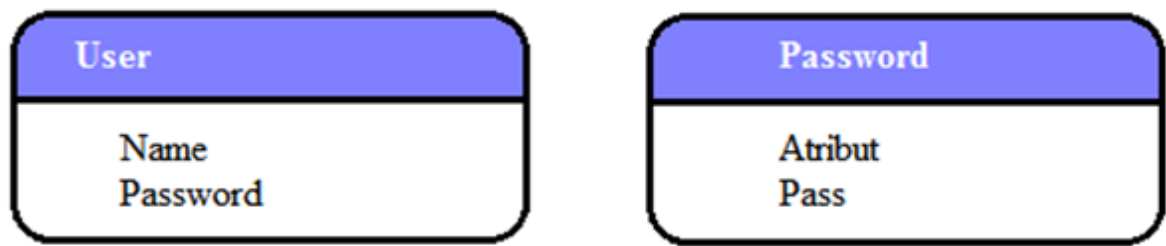


Рисунок 2. Диаграмма классов приложения

Далее приведены методы и поля, которые использовались при создании пользовательского приложения

Класс User - класс, описывающий пользователя

`public static void Save(User user)` - сохраняет значение пользователя в список/файл

`public static string Find(string n)` - проверяет наличие пользователя в списке

`public static bool ComparePass(User user, string pass)` - сравнивает введенный пароль с паролем в списке

`public static bool CompareName(string n)` - проверяет наличие данного имени в списке

User переменные:

`string name` - логин пользователя

`string password` - пароль пользователя

Класс Password

`public static string Generate(int k, bool mb, bool lb, bool ulb, bool rb, bool urb, bool msb, bool sb)` - метод генерации randomного пароля

`public static void Save(string a, string pass)` - метод сохранения пароля

`public static string Find(string a)` - поиск пароля по атрибуту

`public static List<Password> FillDatagrid()` - возвращает список паролей

`public static void EditSave(string a, string na, string np)` - сохраняет отредактированный пароль

`public static void DeleteSave(string a)` - удаляет выбранный пароль

Password переменные:

`string atribut` - атрибут для поиска пароля

`string pass` - пароль

Form1 - основная форма приложения

`private void button6_Click(object sender, EventArgs e)` - обработчик события, отвечающий за

кнопку "авторизация".

private void button2_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "Сгенерировать".

private void button1_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "сохранить".

public string ReturnPass() - возвращает значение сгенерированного/введенного пароля в textBox1

private void button3_Click(object sender, EventArgs e) - обработчик события, отвечающий за кнопку "поиск".

private void button4_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "очистить".

private void button5_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "редактировать".

private void button7_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "открыть".

private void button8_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "удалить".

Form2 - форма авторизации

private void button2_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "регистрация".

private void button1_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "войти".

public string ReturnUserName() - возвращает значения textBox1

Form3 - форма регистрации

private void button1_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "сохранить".

Form4 - форма сохранения пароля

private void button1_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "сохранить".

public string ReturnAtribut() - возвращает значения textBox1

Form5 - форма редактирования пароля

string a - переменная, содержащая атрибут исходного пароля

string p - переменная, содержащая исходный пароль

private void button1_Click(object sender, EventArgs e) - обработчик событий, отвечающий за кнопку "сохранить изменения".

Далее немного рассмотрим как работает разработанное приложение.

Приложение позволяет пользователю создать сгенерированный пароль. Как пароль был

создан пользователь может его редактировать, искать и удалить.

Нужно авторизоваться ,чтобы начать работу.

Далее выбирает модификации генерации.

В поле «Сгенерированный пароль» вводит пароль в соответствии с выбранными модификациями. Нажимаем кнопку «Сохранить» и в поле «Введите атрибут для поиска» вводит параметры по которым будет осуществляться поиск созданного пароля.

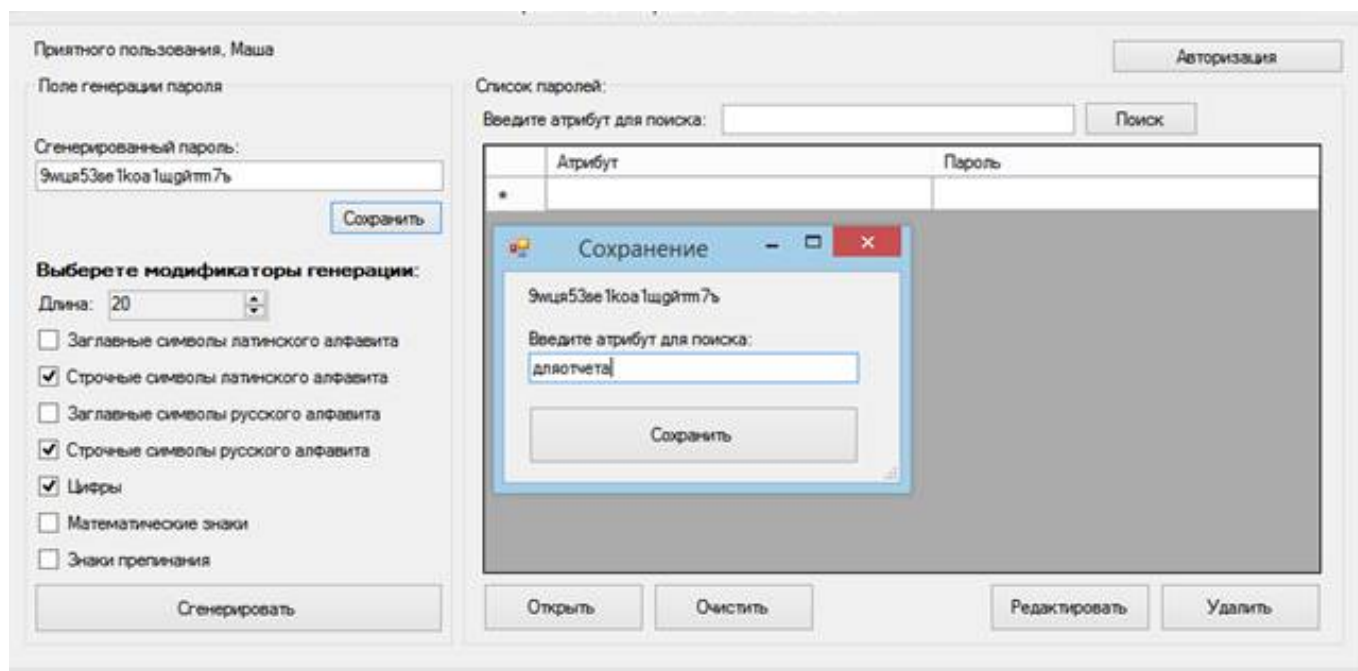


Рисунок 3. Сохранение сгенерированного пароля и ввод атрибута для поиска

В разделе «Список паролей» в поле «Введите атрибут для поиска»вводит параметр ,который создал при создании пароля. При нажатии кнопки «Поиск» пароли,которые имеют заданные атрибуты выйдут на экран.

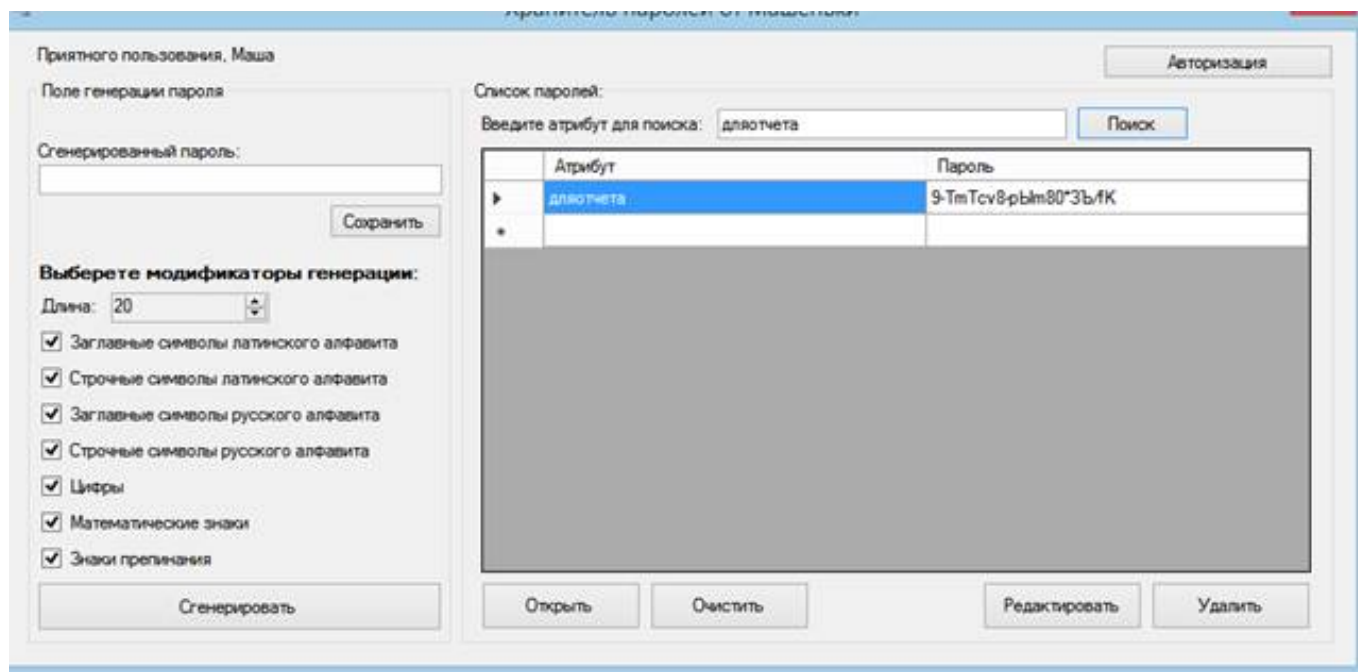


Рисунок 4. Поиск пароля по атрибуту

Для редактирования/ удаления пароля нужно выделить строчку с помощью нажатия на поля рядом с полем «Атрибут».

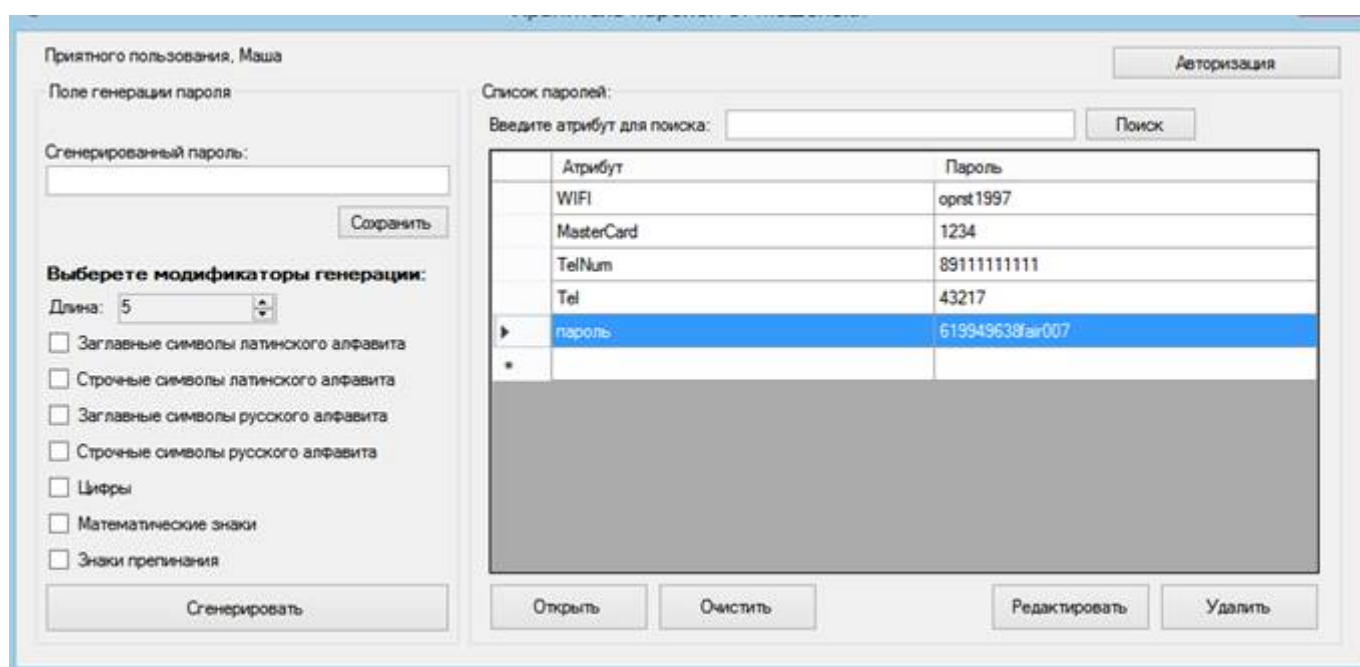


Рисунок 5. Редактирование / удаление пароля

В процессе работы приложения могут возникнуть следующие сообщения:

Результатов не найдено!- если пользователь неверно ввел атрибут для поиска

Выберите пароль для редактирования!-если пользователь нажал кнопку «Редактировать»,

а пароль не выбрал

Нет пароля для сохранения!-если пользователь не ввел пароль в поле «Сгенерированный пароль» и нажал кнопку «Сохранить»

Пользователь с данным именем не найден!- если пользователь пытается авторизоваться, но ввел неправильное имя пользователя или не зарегистрировался.

Список литературы:

1. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. – СПб.: Питер, Русская Редакция, 2007. – 656 с.
2. Руководство по программированию на C# [Электронный ресурс]-
<https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx> -
3. Троелсен Э. Язык программирования C# и платформа .NET 4. – М.: Вильямс, 2010. – 1392 с.
4. Шилдт Г. C# 4.0 полное руководство. – М.: Вильямс, 2011. – 1056 с.