

## Практическая работа №5

### Сохранение настроек приложения

#### 1 Цель работы

- 1.1 Научиться сохранять настройки в клиентском приложении на C#.

#### 2 Литература

- 2.1 <https://learn.microsoft.com/ru-ru/visualstudio/ide/managing-application-settingsdotnet?source=recommendations&view=vs-2022>
- 2.2 <https://docs.microsoft.com/ru-ru/visualstudio/ide/how-to-add-app-config-file?view=vs-2019>
- 2.3 <https://docs.microsoft.com/ru-ru/troubleshoot/developer/visualstudio/csharp/languagecompilers/store-custom-information-config-file>

#### 3 Задание

- 3.1 Чтение и запись параметров в Properties.Settings

3.1.1 Создать оконное приложение с главной формой, формой авторизации и формой настроек.

В обозревателе решений нажать ПКМ на проект, выбрать пункт «Свойства». Перейти к вкладке «Параметры».

Добавить в приложение строковые параметры: логин, email и пароль, задать им значения по умолчанию.

3.1.2 Реализовать открытие формы авторизации с главной формы. Добавить на форму авторизации поля ввода логина и пароля. При нажатии на кнопку «Авторизоваться» сравнивать значения из полей ввода с соответствующими значениями параметров из Properties.Settings.

Если на форме авторизации введенные данные совпадают с хранящимися в Properties.Settings, открыть форму настроек с полями ввода логина, пароля и email. При загрузке формы настроек отображать на ней значения из Properties.Settings.

Если на форме авторизации введенные данные не совпадают с хранящимися в Properties.Settings, вывести сообщение об ошибке.

3.1.3 Добавить на форму настроек кнопку «Сохранить» и написать в ней код, сохраняющий изменения настроек.

- 3.2 Чтение и запись параметров в app.config

Выполнить в отдельном проекте задание 3.1, используя вместо Properties.Settings созданный файл конфигурации приложения app.config.

- 3.3 Чтение и запись параметров в текстовый файл

Выполнить в отдельном проекте задание 3.1, используя вместо Properties.Settings внешний файл userData.txt, хранящий данные только об одном пользователе (1 строка – логин, 2 строка – пароль, 3 строка – email).

### 3.4 Шифрование данных

3.4.1 Добавить в любое приложение из п.3.1-3.3 метод для хэширования пароля алгоритмом SHA256.

*Пример хэширования строки:*

```
byte[] inputBytes = Encoding.UTF8.GetBytes(исходнаяСтрока);  
byte[] hashedBytes = SHA256.Create().ComputeHash(inputBytes);  
string hash = Convert.ToHexString(hashedBytes);
```

3.4.2 Выполнить запись хэшированного пароля при нажатии на кнопку «Сохранить» в форме настроек.

3.4.3 При вводе пароля на форме авторизации сравнивать хэши введенного и хранящегося пароля.

### 3.5 Чтение и запись параметров в json-файл

3.5.1 Добавить в любое приложение из п.3.1-3.3 две пользовательские темы, меняющий внешний вид приложения

3.5.2 Реализовать возможность переключения между темами

3.5.3 Реализовать сохранение названия последней выбранной темы в файле config.json, чтобы она сохранялась при перезапуске приложения.

3.5.4 \* Добавить возможность сохранения пар логин-выбраннаяТема. Для реализации сериализовать не объект, а список (или словарь) объектов, Для поиска выбранной пользователем темы найти его логин и применить соответствующую тему.

## 4 Порядок выполнения работы

4.1 Выполнить все задания из п.3.

4.2 Ответить на контрольные вопросы.

## 5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

## 6 Контрольные вопросы

6.1 Как добавить настройки в приложение на C#?

6.2 Как программно считать значение параметра из настроек?

6.3 Как программно изменить значение параметра в настройках?

6.4 Как выполнить сохранение значений параметров в настройках?