

Лабораторная работа № 8

Тема: Авторизация в базе данных в среде Visual Studio на языке C#

Цель работы: Разработка базы данных в среде Visual Studio на языке C#

Оборудование: IBM – совместимые компьютеры.

Место проведения: Компьютерный класс.

Техника безопасности: См. инструкцию.

Теоретическое описание

Авторизация пользователя

Авторизация (от английского «Authorization») – понятие, изначально использовавшееся в банковской сфере, в области платежных карт и означает процедуру идентификации лица для предоставления доступа к неким ресурсам или возможностям, имеющим ограниченный доступ.

Как правило, процедура авторизации заключается в введении Пользователем в специальную форму своих уникальных идентификационных данных - логина и пароля. Далее эти данные проверяются на наличие в базе данных Пользователей ресурса, и определяется, к какой категории относится данный Пользователь. В результате процедуры авторизации Пользователь получает, либо не получает необходимый доступ (например, Пользователю может быть отказано, если он находится в бан-листе данного ресурса). Логин и пароль ранее должны быть получены Пользователем в результате процедуры регистрации на данном ресурсе. Нередко, зарегистрировавшись на некотором сайте (программе), Пользователь в дальнейшем теряет свои регистрационные данные. На этот случай на сайте должна быть предусмотрена удобная процедура восстановления пароля.

Ход работы

1. Открыть проект приложения, созданного в Лабораторной работе №
2. Добавить форму для авторизации в программе, для этого выбрать в меню Проект – Добавить форму Windows... (Рисунок 1).

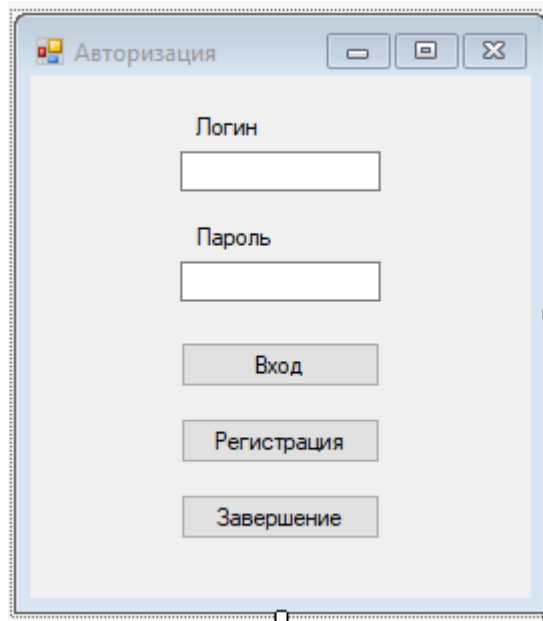


Рисунок 1 – Форма авторизации в базе данных

3. Установить позицию формы по центру экрана.
4. Для кнопки «Завершение» ввести код завершающий работу приложения.
5. В обозревателе решений (Рисунок 2) выбрать Program.cs и в открывшемся окне установить имя формы, которая будет открываться при запуске приложения - `Application.Run(new Form1());` (в данном случае открывается Form1).

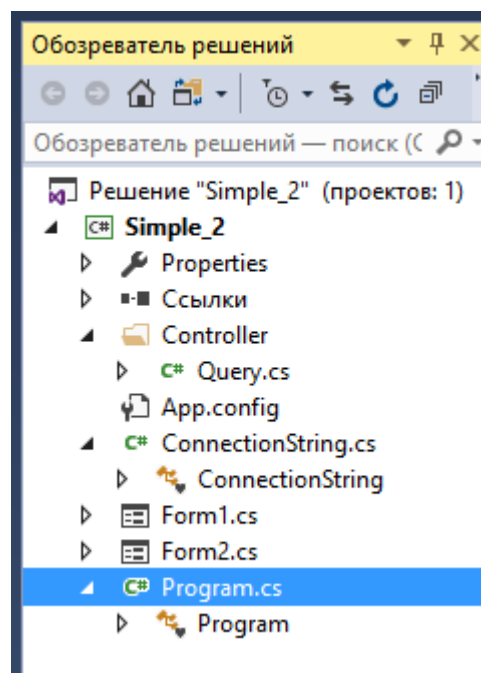


Рисунок 2

6. В СУБД MS Access разработать базу данных с возможностью сохранения логина и пароля пользователя (к существующей базе добавить таблицу User).
7. Добавить в базу данных (MS Access) таблицу Users (рисунок 3)

Users : таблица			
	ID	Log	Pas
▶	1	admin	admin
	2	user	user
	3	1	1
	6	5	5
	7	6	6
	8	7	7
	9	2	2
	10	3	3
	11	12	12
*	(Счетчик)		

Рисунок 3 – Таблица Users

8. Ввести в таблицу несколько значений, среди которых задать логин и пароль – **admin** (рисунок 4).

1	1	1
2	2	2
4	admin	admin
7	3	3

Рисунок 4 – Таблица Users с данными

9. В коде формы Form2 добавить модуль подключения к базе данных
`using System.Data.OleDb; //Подключение базы данных`
 10. Для кнопки «Вход» ввести код

```
private void button1_Click(object sender, EventArgs e)
{
    // подключение файла базы данных
    OleDbConnection con = new OleDbConnection (@"Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=Test1.mdb");
    // запрос на проверку данных
    OleDbDataAdapter dataAdapter = new OleDbDataAdapter("Select Count(*) From Users
where Log =' " + textBox1.Text + "' and Pas =' " + textBox2.Text + "'", con);
    DataTable dt = new DataTable();
    dataAdapter.Fill(dt);
    if (dt.Rows[0][0].ToString() == "1")
    {
        this.Hide();
        Form1 ss = new Form1(); // форма после ввода верных значений логина и
        пароля
        ss.Show(); // открыть форму Form1
    }
    else
    {
        MessageBox.Show("Неправильно введен Логин и (или) Пароль");
    }
}
```

11. Откорректировать код в соответствии со своими данными в базе данных и приложении.

12. Если логин и пароль совпадают со значениями в базе данных, откроется другая форма с данными, если данные не совпадают выводится сообщение (Рисунок 3).

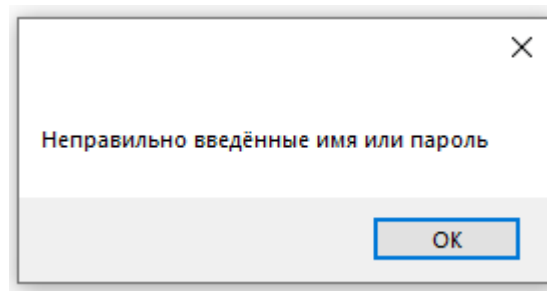


Рисунок 3

13. Добавление, удаление, обновление данных пользователей реализовать аналогично выполненным ранее в классе **Query**, но теперь для обработки данных зарегистрированных пользователей.

14. Проверить работоспособность приложения.