**Тема: Создание дополнительных форм ч.1**

**Задание 1.** Разработать приложение **«Авторизация»**.

При запуске программы открывается окно «Авторизация…» (рисунок 1). Пользователь вводит логин и пароль, которые сверяются со значениями, хранящимися в базе данных.

Если пользователь забыл пароль, то он переходит по ссылке в окно «Восстановление пароля». После успешного восстановления пароля пользователь может войти в систему.

Если пользователь не зарегистрирован в системе, то переходит по ссылке в окно «Регистрация пользователя». После успешной регистрации пользователь может войти в систему.

Если пользователь ввел правильные (имеющиеся в базе данных) логин и пароль, то ему разрешается доступ к приложению, иначе выдается соответствующее сообщение.

**Требования к созданию учётной записи пользователя:**

- логин - должен быть уникальным в системе;

- пароль – не менее 6 символов;

- личные данные (фамилия, имя, e-mail - обязательно).

Технические особенности:

***Реализация окна «Авторизация…»***

- проверка на правильность ввода имени пользователя и пароля. При неверном вводе данных не выводить сообщения с использование класса MessageBox, а отображать уведомления, которое отображается установленное время (рисунок 2);

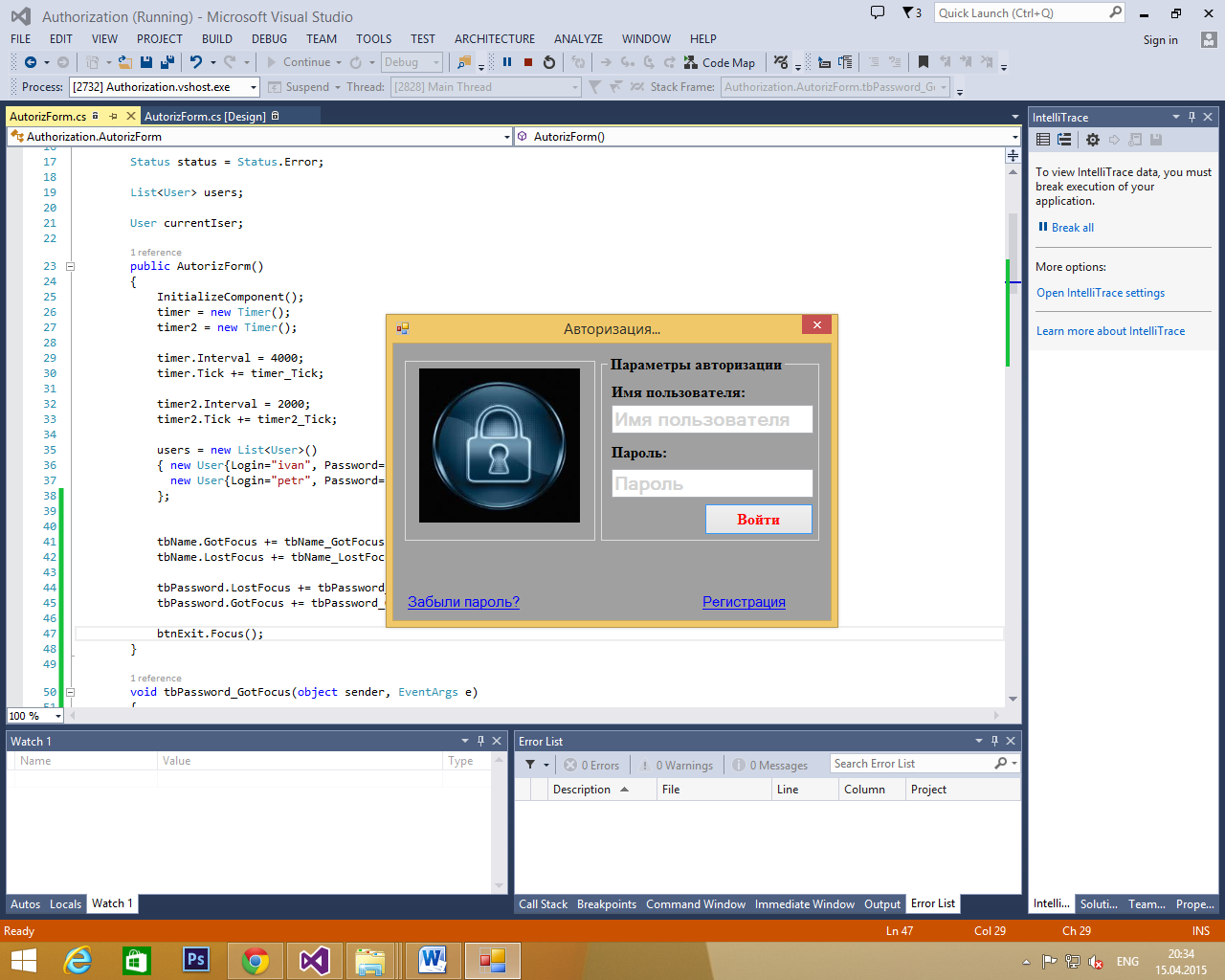
- окно должно появляться в центре, не иметь кнопок свернуть/развернуть, не иметь возможности изменить размеры;

- окно должно быть красиво оформлено;

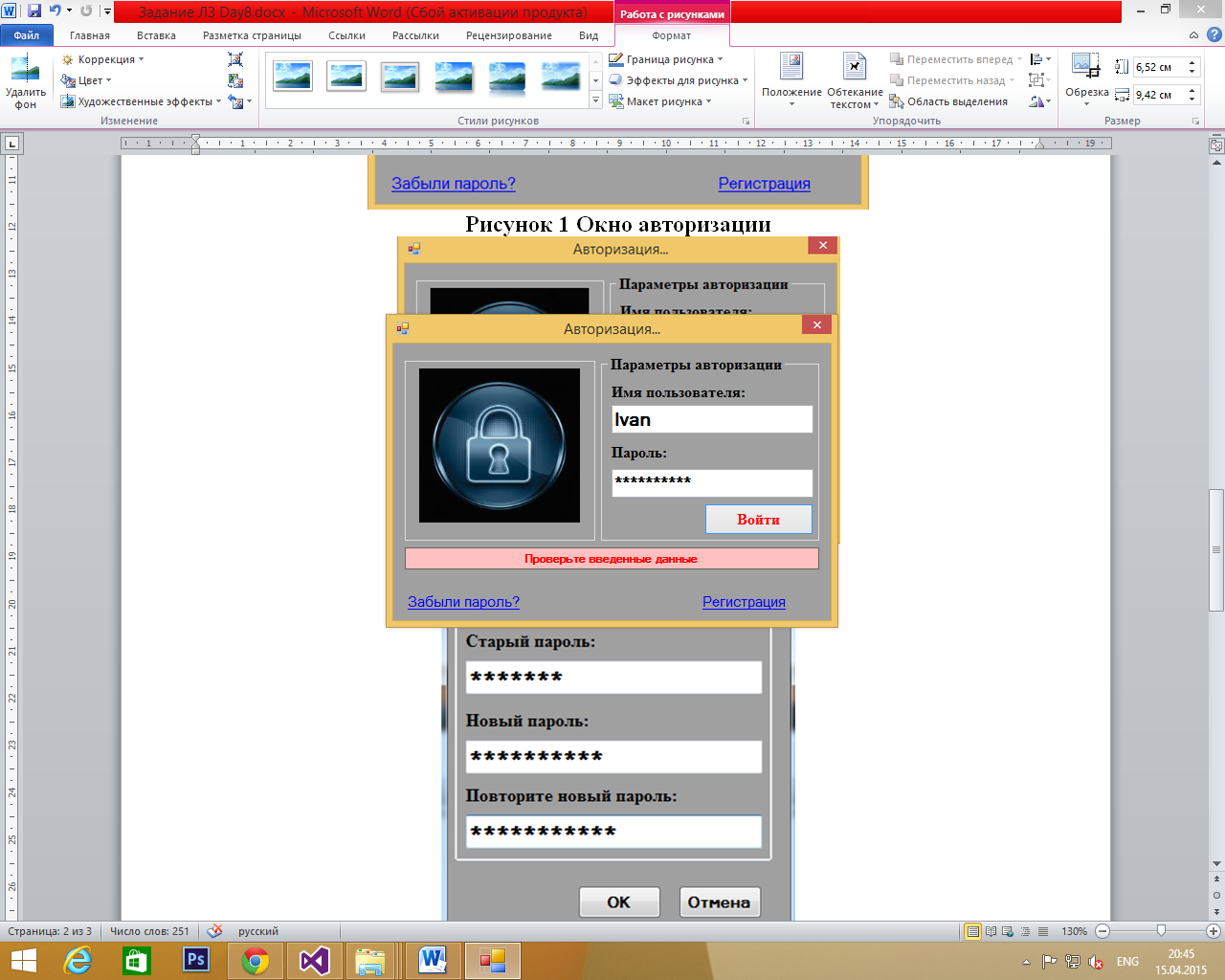
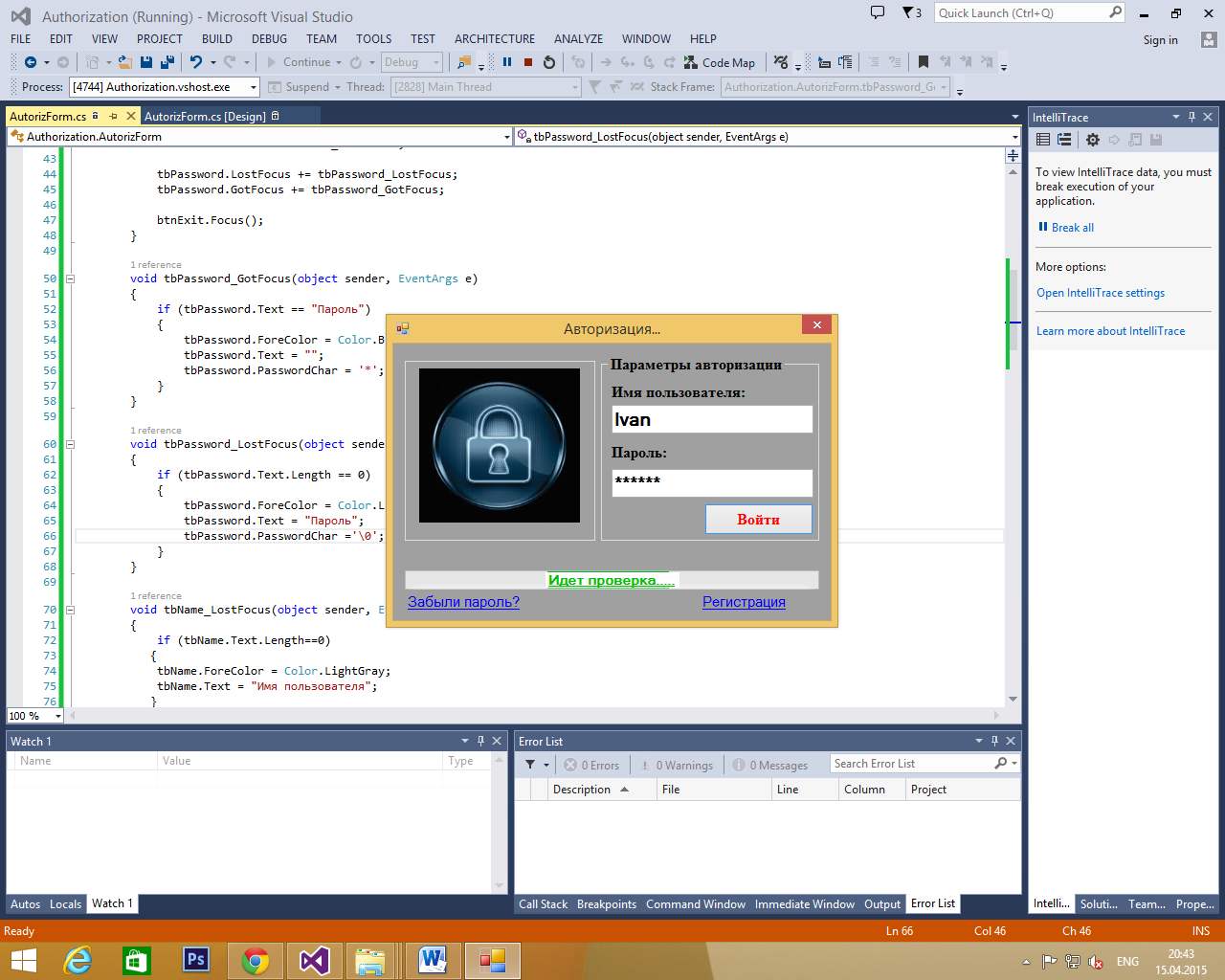
- в ходе проверки введенных данных должна быть имитация с использованием элемента ProgressBar (рисунок 2).

- реализации в поле «Пароль» отображения «звездочек» - использовать свойство **tbPassword.PasswordChar = '\*';** для сброса значения данного свойства - **tbPassword.PasswordChar ='\0';**

- для полей «Имя пользователя» и «Пароль» должны быть настроены события **GotFocus** и **LostFocus,** которые позволяют отображать подсказки в данных полях (серым цветом) (рисунок 1).



**Рисунок 1 Окно авторизации**



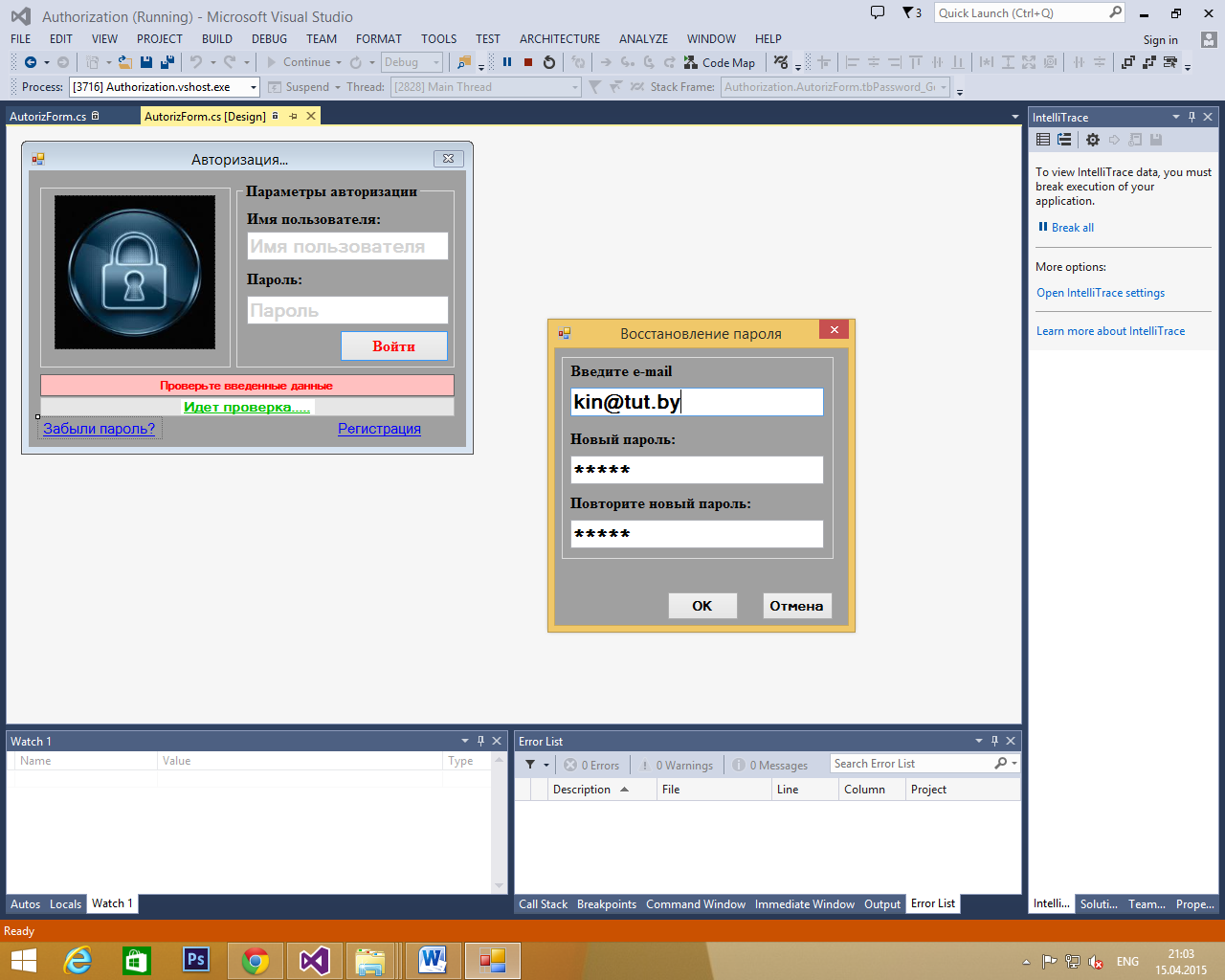
**Рисунок 2 Проверка данных Рисунок 3 Неверные данные**

***Реализация окна «Восстановление пароля»***

- для восстановления пароля пользователь нажимает «Забыли пароль?» (элемент управления LinkLabel), после нажатия окно «Авторизация…» исчезает, открывается окно «Восстановление пароля».

- для возможности изменения пароля пользователь должен ввести e-mail, если такой e-mail имеется, то должно произвестись изменение пароля. Также необходимо сделать проверку на одинаковость введенных новых паролей. Если пользователь ввел несуществующий e-mail, то выдается ошибка и пользователь возвращается в окно «Авторизация».

- после успешного восстановления пароля пользователь возвращается в окно «Авторизация…», окно «Восстановление пароля» исчезает.



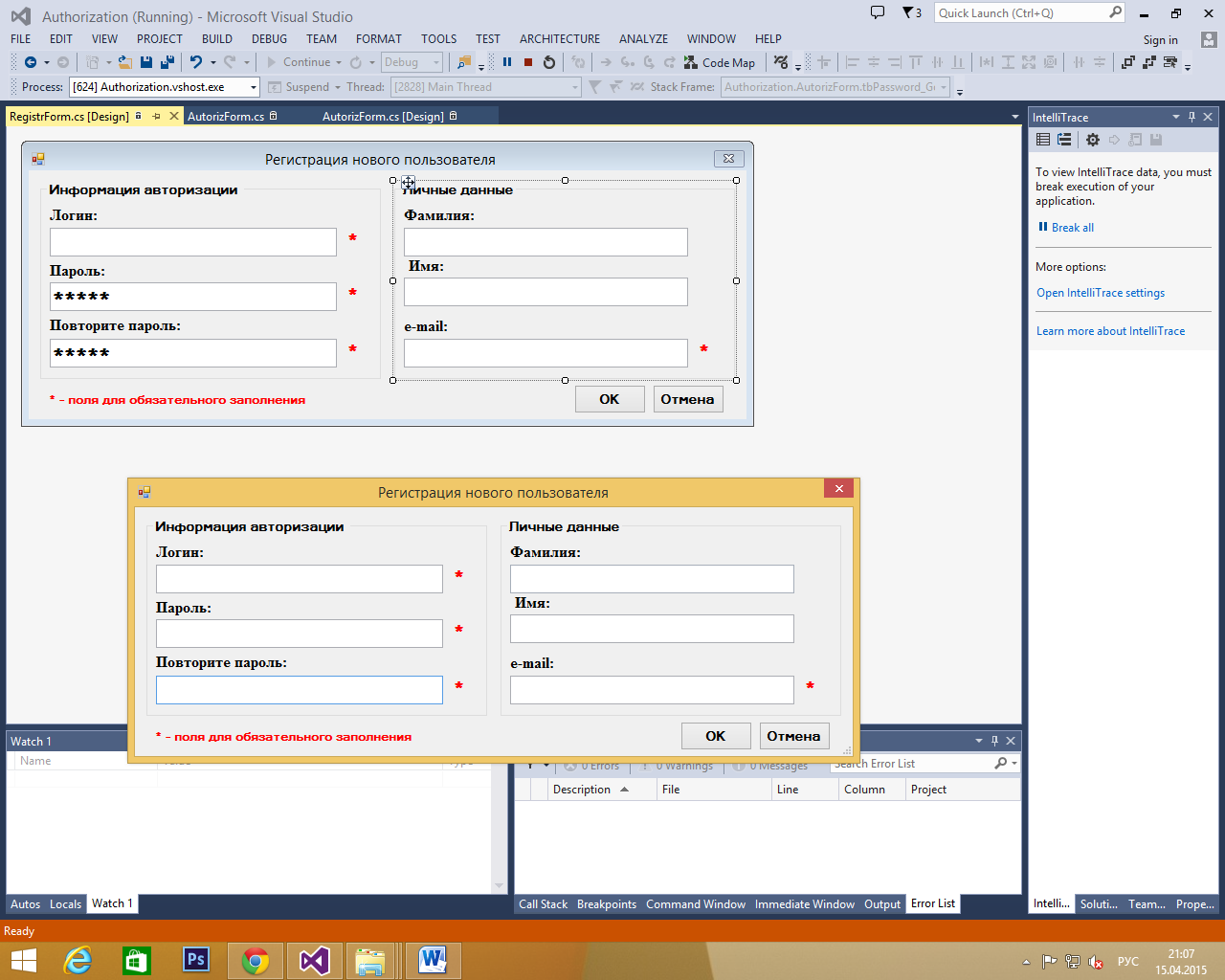
***Реализация окна «Регистрация пользователя»***

- для регистрации нового пользователя пользователь нажимает «Регистрация» (элемент управления LinkLabel), после нажатия окно «Авторизация…» исчезает, открывается окно «Регистрация нового пользователя».

- при регистрации нового пользователя поля, отмеченные звездочкой должны быть обязательно заполнены, - для этого сделать проверку.

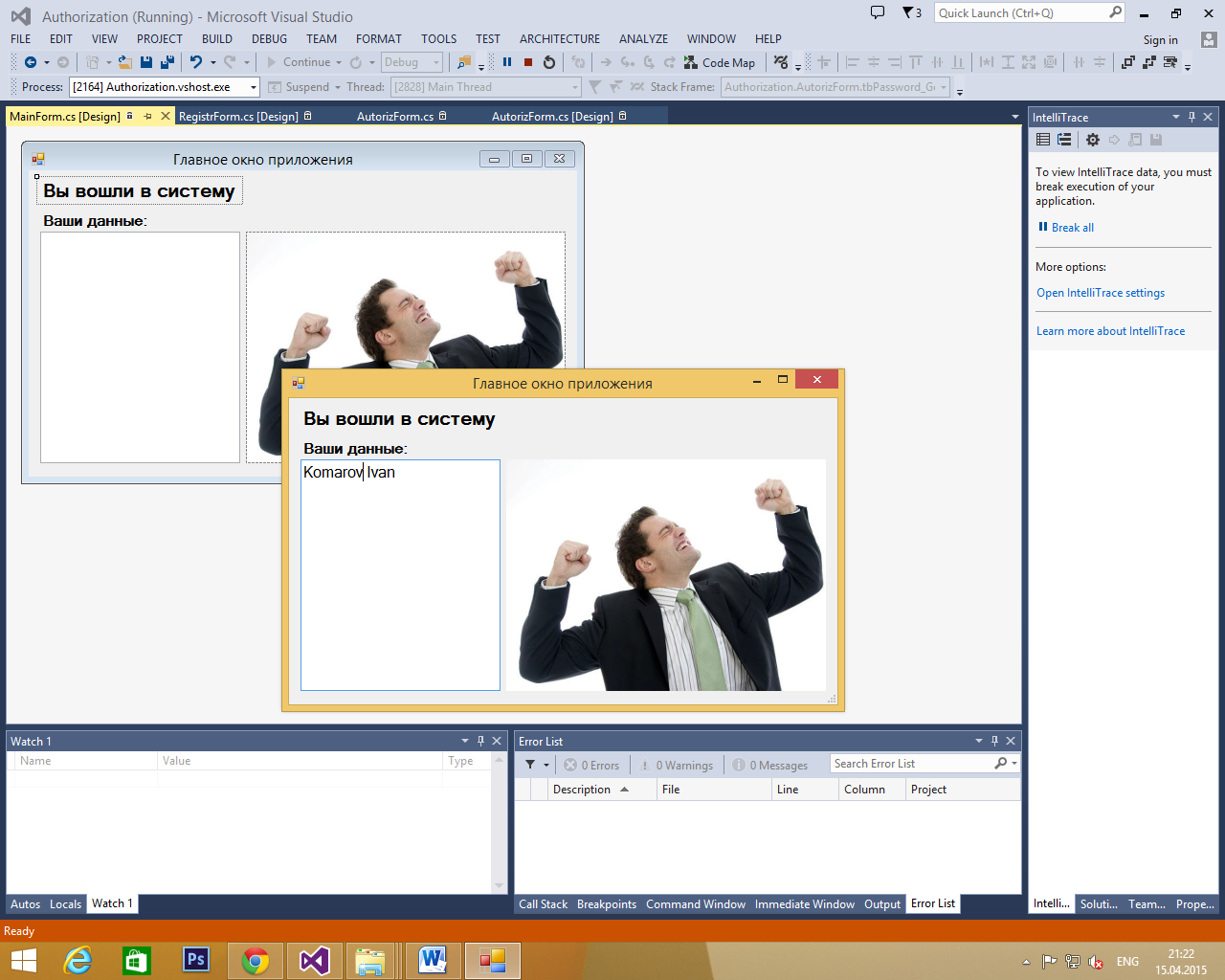
- также необходимо сделать проверку на одинаковость введенных паролей.

- после успешной регистрации пользователь возвращается в окно «Авторизация…», окно «Регистрация пользователя» исчезает.



***Реализация окна «Главное окно приложения»***

- при успешном вводе имени пользователя и пароля открывается окно с именем «Главное окно приложения», в котором отображаются приветственное сообщение с указанием личных данных пользователя.



**Требования к базе данных:**

**- MS SQL, имя базы данных «registration»**

- логин - должен быть уникальным в системе;

- пароль – не менее 6 символов;

- личные данные (фамилия, имя, e-mail - обязательно).

**Дополнительные задания:**

- установить Timeout подключения к базе - 5сек.

- настроить подключение к базе по логину и паролю (SQL server Authentification)

- функциональность проверки наличия в системе пользователя с таким же именем реализовать в виде хранимой процедуры;

- придумать и реализовать использование транзакции.

**Создание таблицы:**

**CREATE TABLE [dbo].[Users] (**

**[Id] BIGINT IDENTITY (1, 1) NOT NULL,**

**[Name] NCHAR (20) NOT NULL,**

**[Password] NCHAR (15) NOT NULL,**

**[Email] NCHAR (20) NOT NULL,**

**PRIMARY KEY CLUSTERED ([Id] ASC)**

**);**

**CREATE TABLE [dbo].[UsersInfo] (**

**[UserId] BIGINT IDENTITY (1, 1) NOT NULL,**

**[LastName] NCHAR (20) NOT NULL,**

**[FirstName] NCHAR (50) NOT NULL,**

**[Address] NCHAR (150) NULL,**

**[Phone] NCHAR (40) NULL,**

**FOREIGN KEY ([UserId]) REFERENCES [dbo].[Users] ([Id])**

**);**

**INSERT INTO [Users] ([Name], [Password], [Email]) VALUES ('Fox','111111',' Fox @mail.by');**

**INSERT INTO [Users] ([Name], [Password], [Email]) VALUES ('SunKot','852632','Sun@mail.by');**

**INSERT INTO [Users] ([Name], [Password], [Email]) VALUES ('Rulon','Run456','rulon@mail.by');**

**INSERT INTO [Users] ([Name], [Password], [Email]) VALUES ('Loput7','qwerty','Loput@mail.by');**

**INSERT INTO [Users] ([Name], [Password], [Email]) VALUES ('Rewa','asd444',' Rewa @mail.by');**

**INSERT INTO [UsersInfo] (LastName, FirstName, Address, Phone) VALUES ('Ivanova','Liza','Grodno','+37529741235');**

**INSERT INTO [UsersInfo] (LastName, FirstName, Address, Phone) VALUES ('Smirnov','Alex','Brest','+375253255334');**

**INSERT INTO [UsersInfo] (LastName, FirstName, Address, Phone) VALUES ('Sidorov','Vova','','+37529865421');**

**INSERT INTO [UsersInfo] (LastName, FirstName, Address, Phone) VALUES ('Krot','Inna','Minsk','+37544234534456');**

**INSERT INTO [UsersInfo] (LastName, FirstName, Address, Phone) VALUES ('Guk','Nikolaj','Minsk','+375339482456');**