**Лабораторная работа по курсу**

**«Использование библиотеки Windows Forms»**

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

**Цель:**

Закрепить у слушателей практические навыки и теоретические знания при взаимодействии с элементами управления. Научиться создавать приложения **Windows Forms**.

**Необходимые инструменты:** MS Visual Studio 2013

**Документация:** Конспект, Литература

**Ориентировочное время исполнения:** 2 часа.

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

При запуске программы открывается окно «Авторизация…» (рисунок 1). Пользователь вводит логин и пароль, которые сверяются со значениями, хранящимися в БД (файловая или SQL server).

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) 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 (в БД);

- 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);

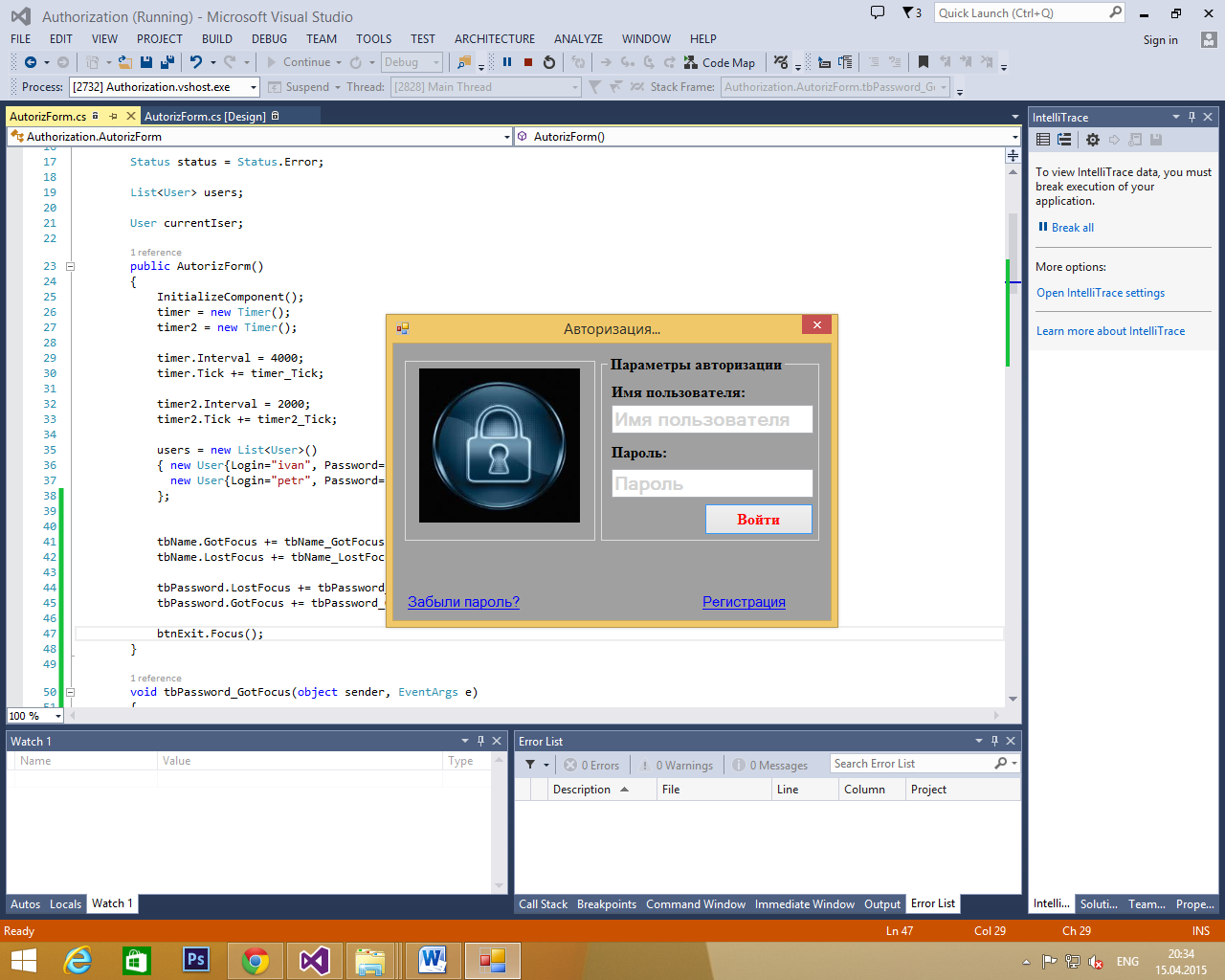
- окно должно появляться в центре, не иметь кнопок свернуть/развернуть, не иметь возможности изменить размеры;

- окно должно быть красиво оформлено;

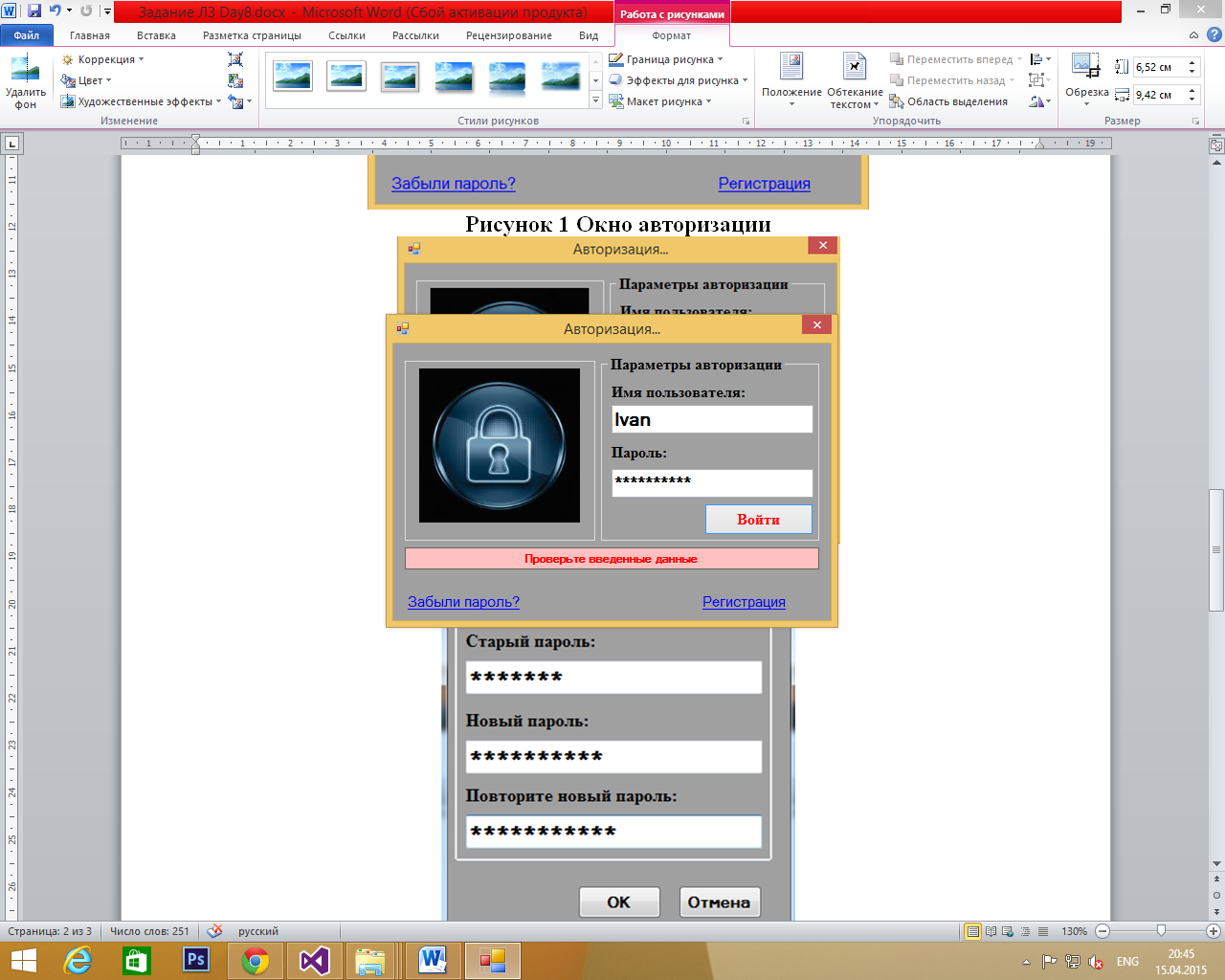
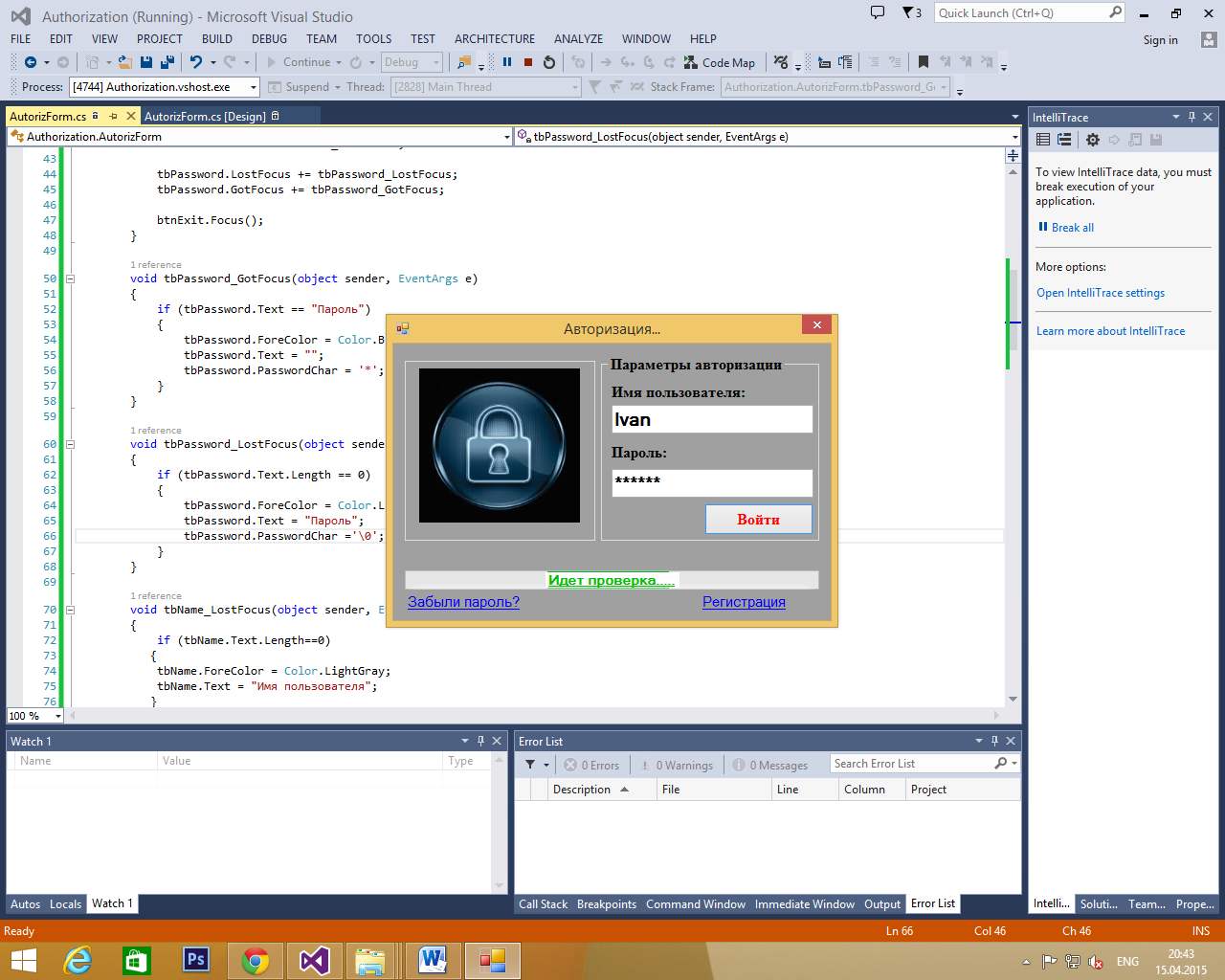
- в ходе проверки введенных данных должна быть имитация с использованием элемента ProgressBar (рисунок 2).

- 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**



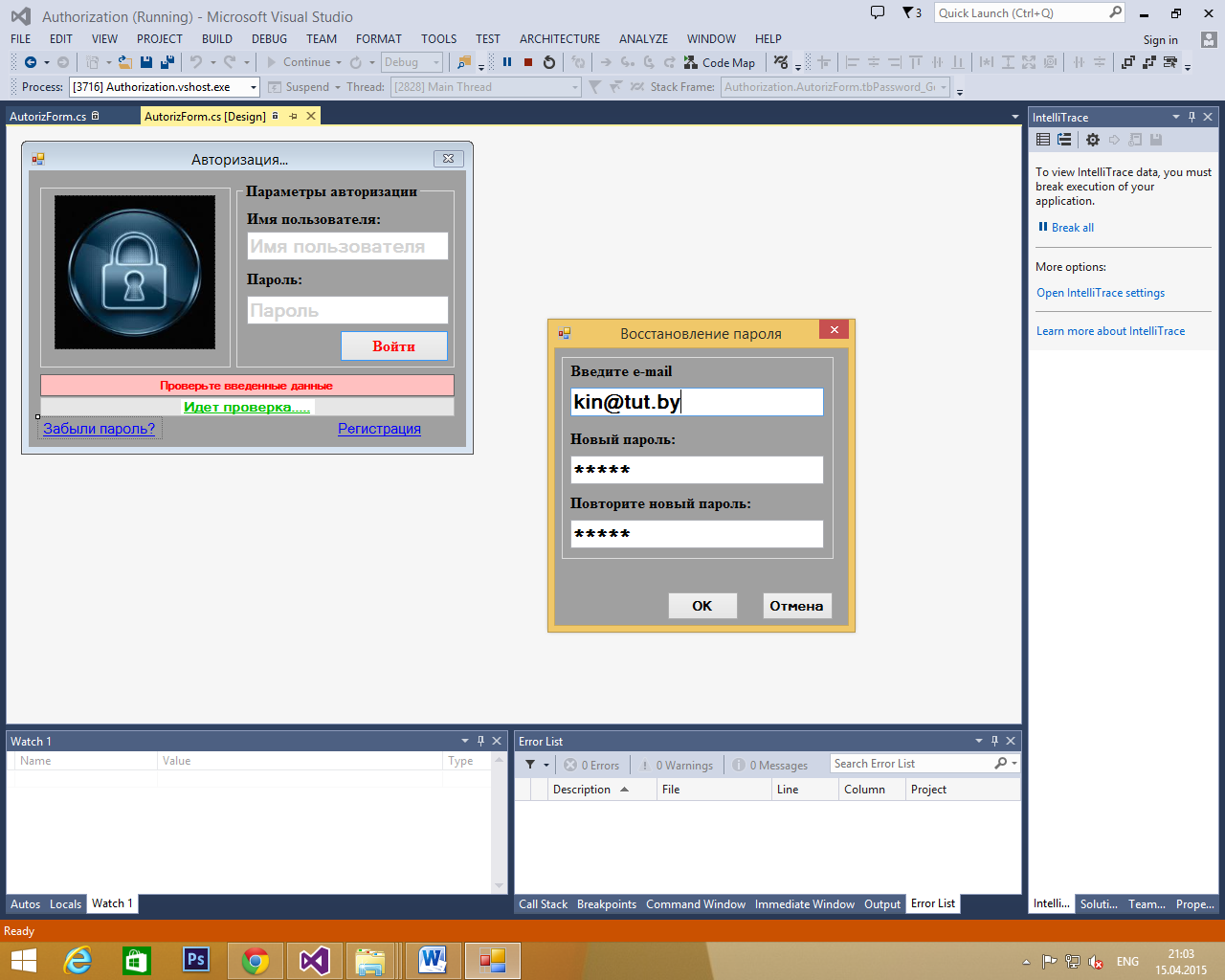
**Рисунок 2 Проверка данных Рисунок 3 Неверные данные**

***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.



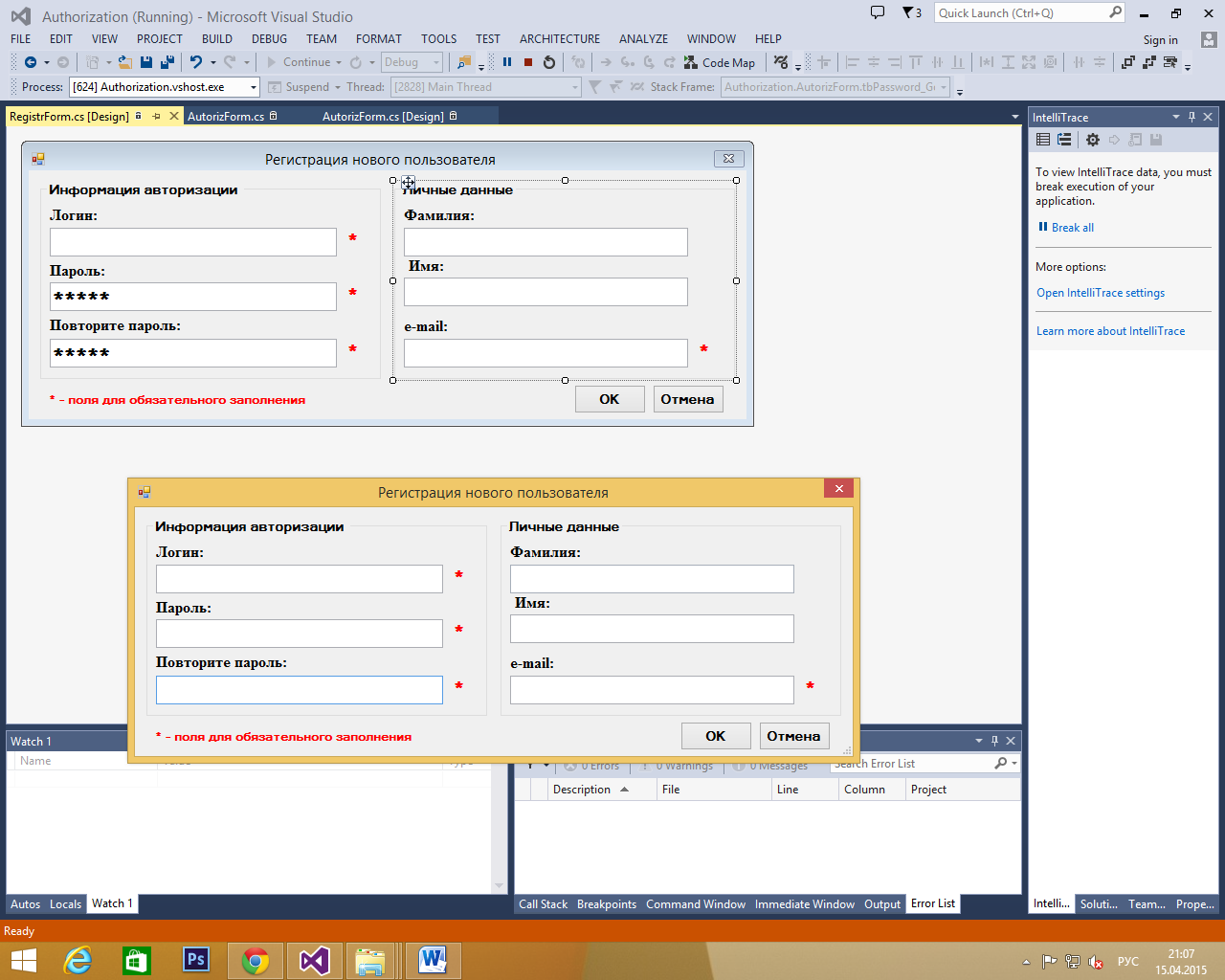
***Реализация окна «Регистрация пользователя»***

- 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.

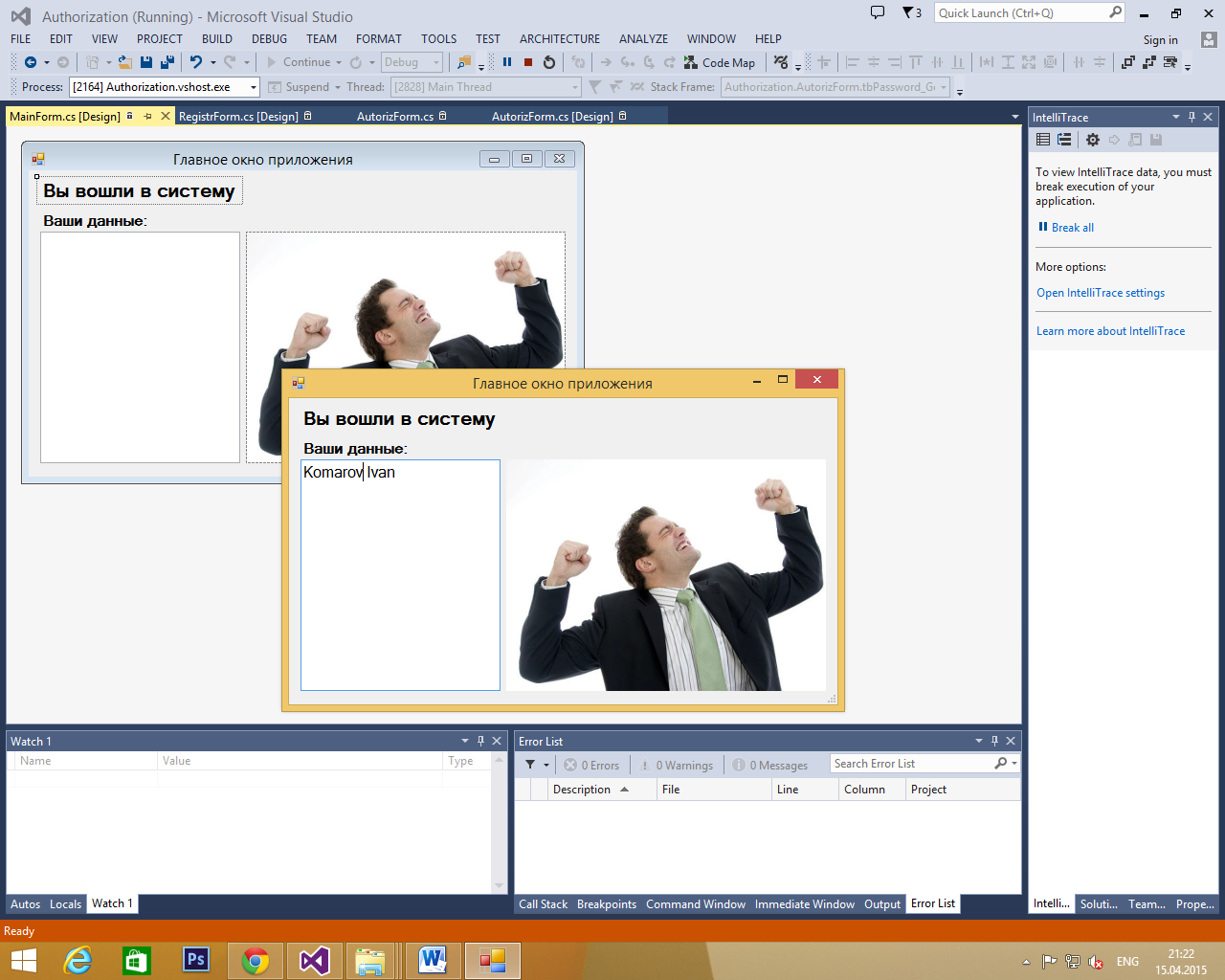
- также необходимо сделать проверку на одинаковость введенных паролей.

- 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.



**Важно!!!**

1. Для отделения логики от UI (user interface) необходимо создать класс «**Репозиторий**», в котором реализована логика работы с «виртуальной базой данной».
2. Для корректности ввода email необходимо вспомнить регулярные выражения.
3. Сохранять пользователей в файл с использованием бинарной сериализации.

**Дополнительно:**

1. Реализовать шифрование пароля и email пользователя с использованием хеш-кода строки (по алгоритму MD5).

[*https://ru.wikipedia.org/wiki/MD5*](https://ru.wikipedia.org/wiki/MD5)

1. Реализовать классы валидаторы (для проверки корректных значений).