**Основы программирования в C#**

Для программирования на языке С# нам понадобиться среда разработки Microsoft Visual Studio. Данная среда поставляется в многочисленных редакциях (смотрите [отличие редакций Visual Studio](http://www.visualstudio.com/products/compare-visual-studio-products-vs)).

Для разработки классических приложений для операционной системы семейства Windows нам понадобиться MS Visual Studio for Windows Desktop. Это бесплатная редакция и для активации требуется лишь зарегистрироваться (если уже имеется аккаунт Microsoft, то достаточно подключить его к Visual Studio). Скачать данную среду разработки можно с [официального сайта проекта](http://www.visualstudio.com/downloads/download-visual-studio-vs).

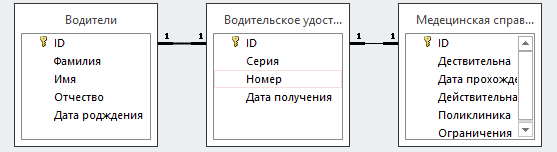
Так же если вы являетесь студентом, школьником, учащимся или преподавателем у Вас есть возможность получить абсолютно бесплатно расширенную версию MS Visual Studio Pro в рамках программы Microsoft DreamSpark (подробности по [ссылке](https://www.dreamspark.com/What-Is-Dreamspark.aspx)). Код активации аккаунта DreamSpark можно получить путем успешной сдачи 2-х любых тестов в Академии Microsoft. Для этого необходимо перейти на сайт [Интернет-университета информационных технологий](http://www.intuit.ru/) и записаться на бесплатные курсы [Академии Microsoft](http://www.intuit.ru/academies/companiesn/42/info). Я рекомендую начать с [этого курса](http://www.intuit.ru/studies/courses/13805/1223/info), так как он поможет познакомиться с возможностями Visual Studio.

Преимущество аккаунта [DreamSpark](https://www.dreamspark.com/What-Is-Dreamspark.aspx) и [Интернет-университета информационных технологий](http://www.intuit.ru/):

* Вы получаете бесплатно лицензионное ПО (Visual Studio, SQL Server, Windows Server и многое другое).
* Вы получаете возможность публиковать свои приложение в Магазине Windows и получать за это деньги абсолютно легальным способом.
* У вас появляется возможность получить бесплатно обучающий материал, в самых различных IT-отраслях.

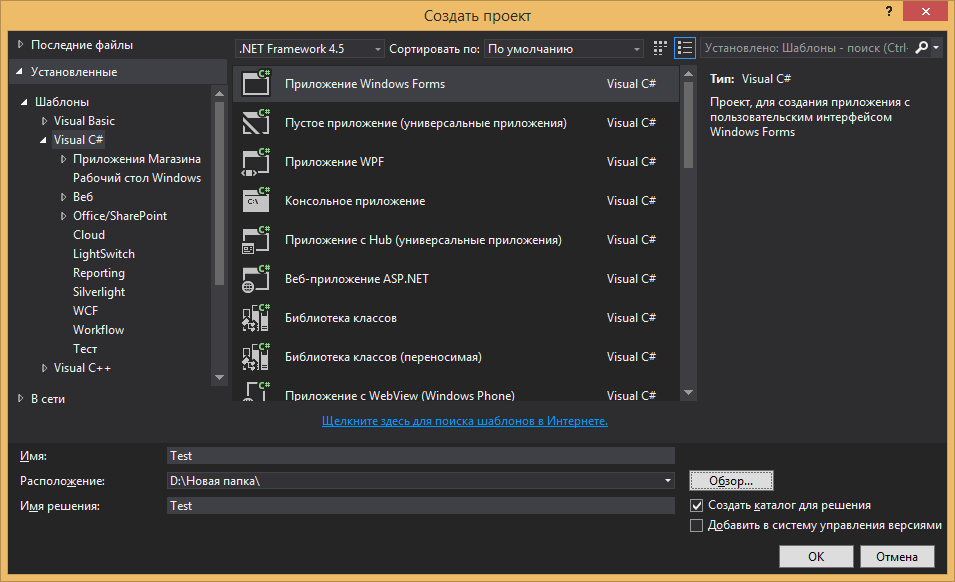
Рассмотрим пример простейшей программы на C# c использование базы данных MS Access и SQL-запросов.

Для начала подготовим саму БД. К примеру у нас будет в БД 3 таблицы связанные между собой. Схема данных представлена ниже.



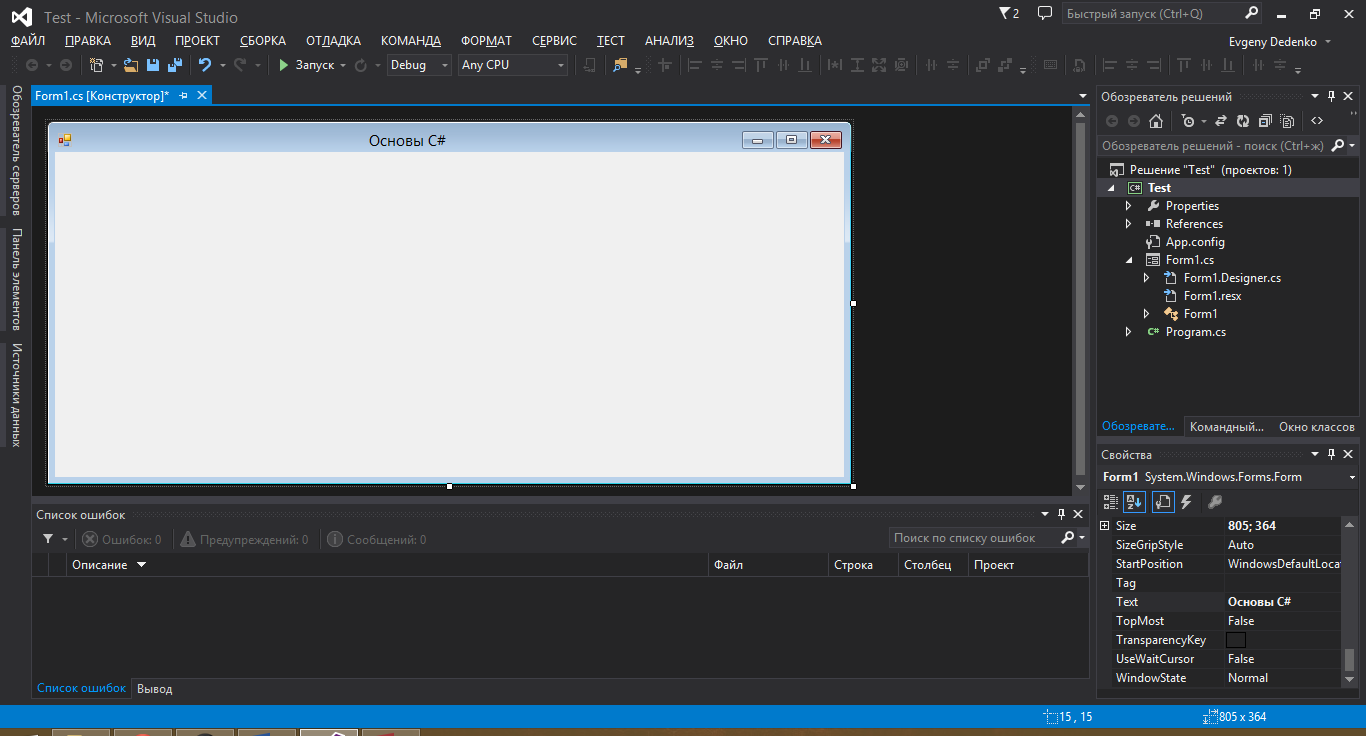
Далее создадим сам проект С# приложения Windows Forms. Для этого запустим Visual Studio (далее VS) и проделаем следующие действия:

В главном меню щелкните по вкладке **«Файл»,** в Контекстном меню выберите **«Проект»**. В открывшимся окне выберите **Visual C#** и **Приложение Windows Forms.**

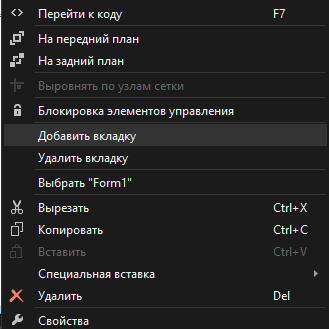


Так же в этом окне можно указать имя проекта, путь и выбрать используемую приложением версию **MS .NET Framework**.

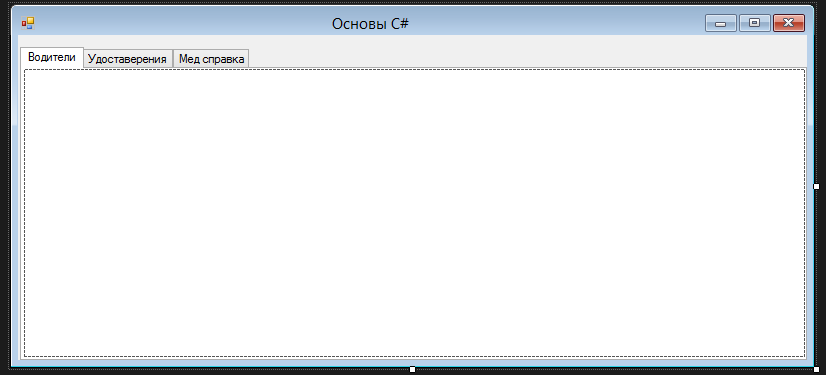
После чего появиться форма. Щёлкнем по ней одним нажатием левой кнопки мыши (далее ЛКМ). И перейдем свойство формы (по-умолчанию находиться в правом нижнем углу) и изменим надпись в верху формы. За эту надпись отвечает свойство «Text». В данном уроке мы поменяем надпись «Form1» на «Основы C#». Результат представлен на скриншоте ниже.



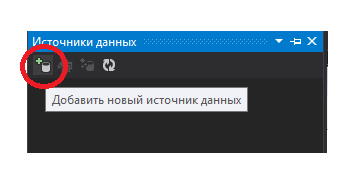
Далее для удобного отображение наших 3 таблиц БД добавим контейнер со вкладками. В VS этот элемент называется **«TabControl».** Для его добавления нужно нажать: **ВИД 🡪 ПАНЕЛЬ ЭЛЕМЕНТОВ 🡪 ВЫБРАТЬ TABCONTROL И ПЕРЕТАЩИТЬ НА ФОРМУ.** На форме появиться контейнер с двумя вкладками. Теперь подгоняем его положение на форме и размер. Далее добавим еще одну вкладку, для этого правой клавишей мыши жмем по нашему контейнеру и выбираем в контекстном меню пункт «Добавить вкладку». Контекстное меню представлено на рисунке ниже.



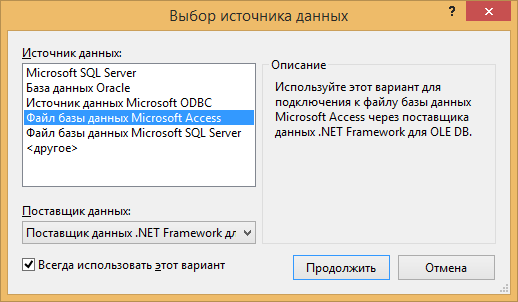
Далее в свойствах (аналогично форме) приманиваем вкладки. Должно получиться как на рисунке ниже.



Далее подключим заранее созданную БД к нашему проекту. Для этого проделаем следящие: **ВИД 🡪 ДРУГИЕ ОКНА 🡪 ИСТОЧНИКИ ДАННЫХ 🡪 ДОБАВИТЬ НОВЫЙ ИСТОЧНИК ДАННЫХ**

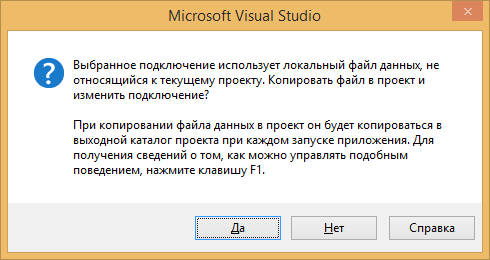


В появившемся окне выбираем «База данных» и жмем кнопку «Далее». После чего выбираем «Набор данных» и опять жмем «Далее». После чего жмем кнопку «Создать подключение». В появившемся окне выбираем «**Файл базы данных Access**» и жмём «Продолжить».

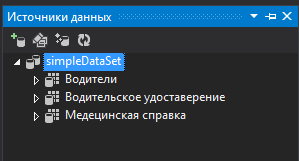


Далее выбираем путь к файлу нашей БД и все время жмем «Продолжить».

Далее VS нам предложит копировать в файл БД в проект (рисунок ниже), я нажал «Да», так как это далее облегчит процесс подключение к БД с помощью кода.

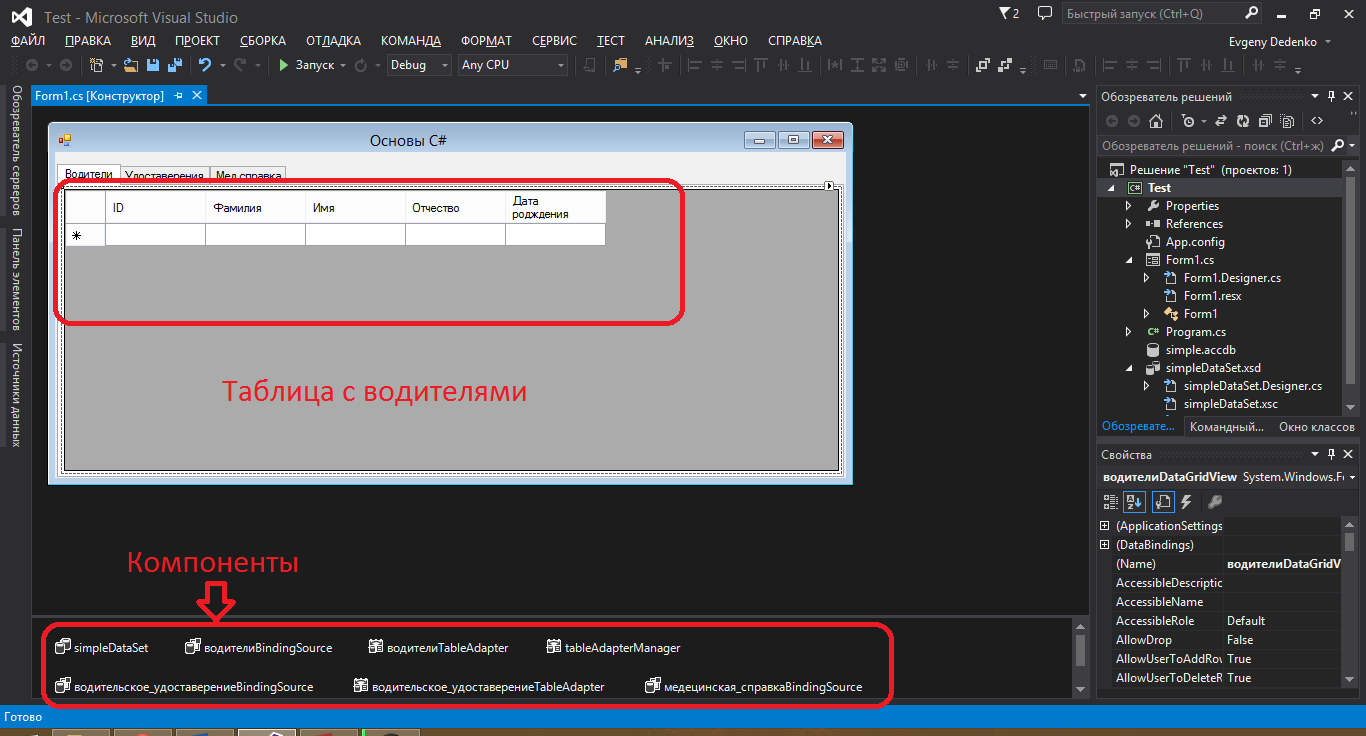


После чего выбираем все таблицы и представления. Теперь они отображаются в «Источнике данных».

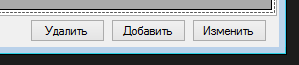


Теперь щелкаем по таблицы и протаскиваем на форму в каждую ячейку контейнера соответственно. Все необходимые компоненты добавиться автоматически.

Должно получиться как на рисунке.



Теперь добавим 3 кнопки на форму (для этого перетащите из Панели элементов 3 элемента Button).

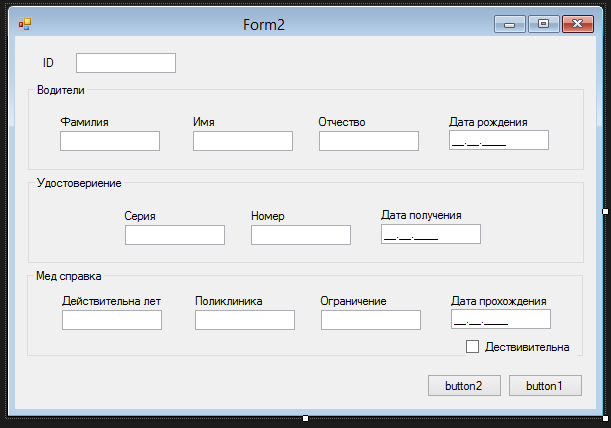
* Добавить
* Удалить
* Изменить

Создадим еще одну форму: **ПРОЕКТ 🡪 ДОБАВИТЬ ФОРМУ WINDOWS (**можно поменять имя формы**)**

Далее выносим компоненты как на картинке:

Для этого из Панели Элементов выносим следующие компоненты:

1. Label – служит для отображения надписи без возможности ввода данных
2. TextBox – служит для ввода и отображения данных
3. MaskedTextBox – служит для ввода и отображения данных с использованием масок ввода
4. ChekedBox – Компонент, позволяющий отобразить логическое вырожение «Да/Нет»
5. GrupBox – контейнер с надписью.
6. Button – кнопка.



После чего выбираем cherckbox, textbox-ы, maskedtexbox-ы и button-ы и меняем им всем свойство «**Modifiers»** на public**.** Этим мы меняем модификаторы доступа на общедоступные, т.е. к этим элементам можно будет обратиться с любой формы.

Далее переходим к коду первой формы и делаем функцию обновления:

private void update\_tabs()

{

//Следующий код сохраняет изменения в БД

this.Validate();

this.водителиBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.simpleDataSet);

//Следующий код обновляет 3 таблицы, которые расположены на форме

this.медецинская\_справкаTableAdapter.Fill(this.simpleDataSet.Медецинская\_справка);

this.водительское\_удоставерениеTableAdapter.Fill(this.simpleDataSet.Водительское\_удоставерение);

this.водителиTableAdapter.Fill(this.simpleDataSet.Водители);

}

Далее возвращаемся к первой форме и дважды щелкаем на кнопку «Изменить».

Ниже представлена процедура изменения:

private void button3\_Click(object sender, EventArgs e)

{

//В С# нельзя на прямую обратиться к компонету. Это связано с политикой безопасности.

//Для того что бы отратиться к обьекту нужно инициализировать метод.

Form2 f2 = new Form2(); // инициализируем медод для второй формы

f2.Text = "Изменение данных"; // Меняем текст в шапке нашей формы

// Меняем надписи на кнопках

f2.button1.Text = "Отмена";

f2.button2.Text = "Сохранить";

// запрет на редактирование ID

f2.textBox1.ReadOnly = true;

//вызов второй формы

f2.Show();

//Далее произведем код считывания из бд c помощью SQL-запроса

string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data Source='|DataDirectory|\simple.accdb';OLE DB Services=-1"; // Строка подлючения к БД

OleDbConnection connection = new OleDbConnection(connectionString); //Метод для обращения к файлу БД

OleDbCommand command = new OleDbCommand(); //Метод для выполнения операциий с БД

command.Connection = connection; //Связываем 2 придыдущие строчки

command.CommandText = "SELECT \* FROM Водители, [Водительское удоставерение], [Медецинская справка] WHERE Водители.ID = [Водительское удоставерение].ID AND [Водительское удоставерение].ID = [Медецинская справка].ID AND Водители.ID =" + водителиDataGridView.Rows[водителиDataGridView.CurrentCell.RowIndex].Cells[0].Value.ToString() + ";"; // строка SQL запроса

try // Следующий код выполняется если нет ошибок

{

connection.Open(); //Открываем БД

OleDbDataReader dr = command.ExecuteReader(); // Мсетод для чтения БД

if (dr.HasRows) // Условие выполнятеся пока в БД есть непрочтенные строки

{

while (dr.Read()) // цикл чтения

{

//заносим считанное на форму

f2.textBox1.Text = dr["Водители.ID"].ToString();

f2.textBox2.Text = dr["Фамилия"].ToString();

f2.textBox3.Text = dr["Имя"].ToString();

f2.textBox4.Text = dr["Отчество"].ToString();

f2.maskedTextBox1.Text = dr["Дата родждения"].ToString();

f2.textBox6.Text = dr["Серия"].ToString();

f2.textBox5.Text = dr["Номер"].ToString();

f2.maskedTextBox2.Text = dr["Дата получения"].ToString();

f2.checkBox1.Checked = Convert.ToBoolean(dr["Дествительна"]);

f2.maskedTextBox3.Text = dr["Дата прохождения коммисии"].ToString();

f2.textBox9.Text = dr["Действительна лет"].ToString();

f2.textBox8.Text = dr["Поликлиника"].ToString();

f2.textBox7.Text = dr["Ограничения"].ToString();

}

}

connection.Close(); // закрываем БД

}

catch (Exception ex) // этот код выполняется если есть ошибки

{

//вывод сообщения об ошибке ошибки

MessageBox.Show("Ошибка получения данных: " + Environment.NewLine + ex.ToString(), "Ошибка получения данных");

}

f2.button1.MouseDown += (sender1, e1) => // действия по кнопке "отмена"

{

update\_tabs(); // обновление таблиц

f2.Close(); // скрыть вторую форму

};

f2.button2.MouseDown += (sender1, e1) => // действия по кнопке "Сохранить"

{

MessageBoxButtons buttons = MessageBoxButtons.YesNo; // объявляем кнопки для диалога сохранения

DialogResult result; // объявляем переменную для результата диалога сохранения

result = MessageBox.Show("Сохранить изменения?", "Сохранение данных", buttons); // диалог сохранения

if (result == System.Windows.Forms.DialogResult.Yes) // если ответ диалога сохранения утвердительный выполняем следующий код

{

try

{

// записываем в буферные переменные значения с maskedTextBox

string s = f2.maskedTextBox1.Text;

string s1 = f2.maskedTextBox2.Text;

string s2 = f2.maskedTextBox3.Text;

connection.Open();

// в следующием sql запросе испльзуется функция замены Repalace(".","/"); Она меняет точку на слеш!

command.CommandText = "UPDATE Водители, [Водительское удоставерение], [Медецинская справка] SET Фамилия ='" + f2.textBox2.Text + "', Имя = '" + f2.textBox3.Text + "', Отчество = '" + f2.textBox4.Text + "', [Дата родждения] = #" + s.Replace(".", "/") + "#, Серия = '" + f2.textBox6.Text + "', Номер = '" + f2.textBox5.Text + "', [Дата получения] = #" + s1.Replace(".", "/") + "#, Дествительна = '" + Convert.ToInt32(f2.checkBox1.Checked) + "', [Дата прохождения коммисии] = #" + s2.Replace(".", "/") + "#, [Действительна лет]= '" + f2.textBox9.Text + "', Поликлиника = '" + f2.textBox8.Text + "', Ограничения = '" + f2.textBox7.Text + "' WHERE Водители.ID = [Водительское удоставерение].ID AND [Водительское удоставерение].ID = [Медецинская справка].ID AND Водители.ID =" + водителиDataGridView.Rows[водителиDataGridView.CurrentCell.RowIndex].Cells[0].Value.ToString() + ";";

command.ExecuteNonQuery(); // выполняем Sql запрос

connection.Close();

update\_tabs();

}

catch

{

MessageBox.Show("Не верно ввведены даныые! Проверте правильность ввода!", "Ошибка ввода!");

}

}

};

}

Далее напишем код для добавления в БД записи. Для этого нужно перейти в конструктор первой формы и дважды щелкнуть по кнопке «Добавить».

Создаться автоматически процедура, в теле которой будем писать следующий код:

private void button2\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Text = "Добавление водителя";

f2.button1.Text = "Отмена";

f2.button2.Text = "Добавить";

f2.Show();

// объявление буферных переменных для хранения даты

DateTime d = new System.DateTime();

DateTime dv = new System.DateTime();

DateTime ds = new System.DateTime();

f2.button2.MouseDown += (sender1, e1) =>

{

try

{

d = DateTime.Parse(f2.maskedTextBox1.Text); // вырожение присваивате переменной, при этом переводит его в краткий формат даты и времени

simpleDataSetTableAdapters.ВодителиTableAdapter vod = new simpleDataSetTableAdapters.ВодителиTableAdapter(); //инициализируем метод для обращение к таблицы "Водители"

vod.Insert(Convert.ToInt32(f2.textBox1.Text), f2.textBox2.Text, f2.textBox3.Text, f2.textBox4.Text, d); // функция вставляет в таблицу "Водители" строку

simpleDataSetTableAdapters.Водительское\_удоставерениеTableAdapter vod\_u = new simpleDataSetTableAdapters.Водительское\_удоставерениеTableAdapter(); //инициализируем метод для обращение к таблицы "Водительское удоставерение"

dv = DateTime.Parse(f2.maskedTextBox2.Text); // вырожение присваивате переменной, при этом переводит его в краткий формат даты и времени

vod\_u.Insert(Convert.ToInt32(f2.textBox1.Text), f2.textBox6.Text, f2.textBox5.Text, dv); // функция вставляет в таблицу "Водительское удоставерение" строку

simpleDataSetTableAdapters.Медецинская\_справкаTableAdapter mspr = new simpleDataSetTableAdapters.Медецинская\_справкаTableAdapter(); //инициализируем метод для обращение к таблицы "Медицинская справка"

ds = DateTime.Parse(f2.maskedTextBox3.Text); // вырожение присваивате переменной, при этом переводит его в краткий формат даты и времени

mspr.Insert(Convert.ToInt32(f2.textBox1.Text), Convert.ToBoolean(f2.checkBox1.Enabled), ds, Convert.ToInt32(f2.textBox9.Text), f2.textBox8.Text, f2.textBox7.Text); // функция вставляет в таблицу "Медицинская справка" строку

update\_tabs();

}

catch

{

MessageBox.Show("Ошибка ввода. Проверте правильность введенных данных.", "Ошибка ввода");

}

finally // выполняется в любом случае, независимо есть или нет ошибок

{

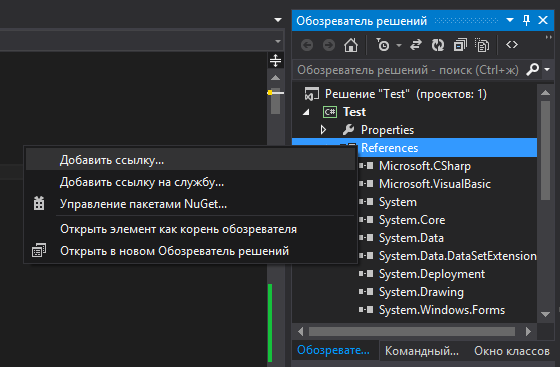
update\_tabs();

f2.Close();

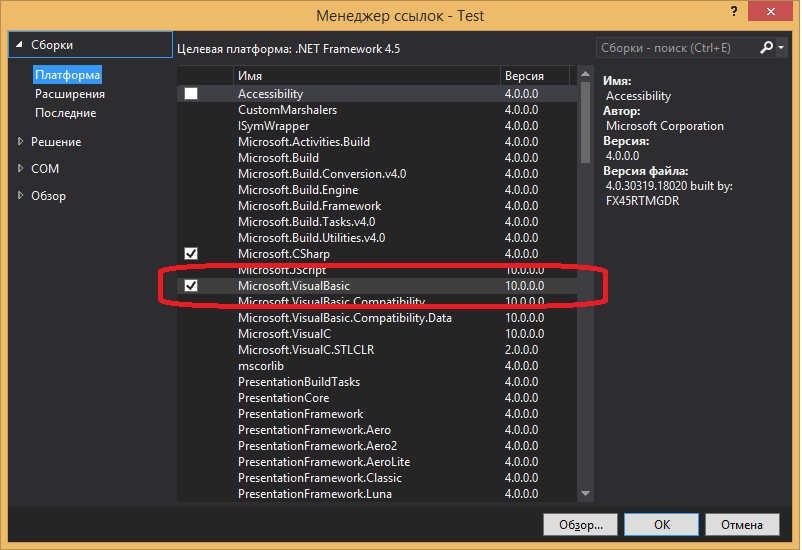
}

};

Далее сделаем процедуру удаления записи о водители по ID. В С# нет диалога для ввода, поэтому подключим возможности языка Visual Basic. Для этого в **Обозреватели решений** щелкаем ПКМ по **References** и выбираем пункт **Добавить ссылку…**



В менеджере ссылок ставим галочку **Microsoft.VisualBasic**



Далее на первой форме щелкаем двойным щелчком по кнопке «Удалить», и пишем следующий код:

private void button1\_Click(object sender, EventArgs e)

{

string g = "";

g = Interaction.InputBox("Введите ID удаляемого водителя", "Удаление водителя"); // диалог ввода из Visual Basic

if (g != "") // если пользователь ничего не ввел, то ничего не происходит

{

try

{

string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;" + @"Data Source='|DataDirectory|\simple.accdb';OLE DB Services=-1";

OleDbConnection connection = new OleDbConnection(connectionString);

OleDbCommand command = new OleDbCommand();

command.Connection = connection;

// запрос на удаление

command.CommandText = "DELETE FROM Водители WHERE ID =" + g + ";";

connection.Open();

command.ExecuteNonQuery();

connection.Close();

update\_tabs();

}

catch

{

MessageBox.Show("Неверно введен ID");

}

finally

{

update\_tabs();

}

}

}

Сделаем еще одну кнопку для принудительного обновления таблиц.

private void button4\_Click(object sender, EventArgs e)

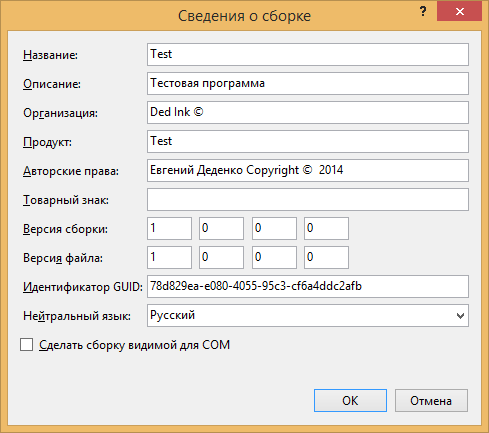
{

update\_tabs();

}

И напоследок добавим информацию о программе.

**ПРОЕКТ 🡪 СВОЙСТВА: Test 🡪 ПРИЛОЖЕНИЯ 🡪 СВЕДЕНИЯ О СБОРКЕ…**



**Скачать проект можно по ссылке:** [**https://yadi.sk/d/\_XfZtGp6cytCC**](https://yadi.sk/d/_XfZtGp6cytCC)