**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КАЛУЖСКОЙ ОБЛАСТИ**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Государственное автономное профессиональное образовательное учреждение**

**Калужской области**

**«Калужский технический колледж»**

***специальность Информационные системы  
(по отраслям)***

**ОТЧЕТ**

**ЛАБОРАТОРНАЯ РАБОТА**

**Тема «Разработка приложения автоматизации работы диспетчера касс автовокзала»**

**«Основы алгоритмизации и программирование»**

**Выполнил:** студент гр.2 ИС Щукин В.С.\_\_\_\_\_\_\_\_\_

**Проверил:** преподаватель Шмелёв Н.Б.\_\_\_\_\_\_\_\_\_

Дата защиты лабораторной работы:

Оценка:\_\_\_\_\_\_\_\_\_

**г. Калуга, 2021**

Цель: разработать приложение автоматизации работы диспетчера касс автовокзала

Ход работы:

[**1. Создание базы данных**](https://www.bestprog.net/ru/2016/03/23/009-c-%d0%ba%d1%83%d1%80%d1%81%d0%be%d0%b2%d0%b0%d1%8f-%d1%80%d0%b0%d0%b7%d1%80%d0%b0%d0%b1%d0%be%d1%82%d0%ba%d0%b0-%d0%bf%d1%80%d0%b8%d0%bb%d0%be%d0%b6%d0%b5%d0%bd%d0%b8%d1%8f-%d0%b0%d0%b2/#p01)

Базу данных проектируем в системе Microsoft Access. Имя файла базы данных даем [02\_02\_00\_009\_Baza\_ru.mdb](https://www.bestprog.net/wp-content/uploads/2016/03/02_02_00_009_Baza_ru.mdb).

В соответствии с условием задачи, имеем 6 таблиц. Пять таблиц содержат данные о:

* маршрутах перевозки;
* проданных билетах;
* диспетчера, который вносит информацию в базу данных;
* транспортное средство (автобус, маршрутное такси), на котором осуществляется перевозка;
* водителя, который выполняет перевозку.

**2. Создание проекта в Microsoft Visual Studio**

Установить шаблон «Visual C#«, тип проектаWindows Forms Application.

Имя приложения даем по умолчанию «WindowsFormsApplication1».

**3. Подключение базы данных**

На этом шаге осуществим подключение базы данных «[02\_02\_00\_009\_Baza\_ru.mdb](https://www.bestprog.net/wp-content/uploads/2016/03/02_02_00_009_Baza_ru.mdb)» к нашему приложению. Чтобы вызвать процесс подключения к базе данных, необходимо вызвать команду «Add New Data Source …» из меню Data. В окне выбора типа источника базы данных выбираем Database. В окне выбора модели подключения к базе данных выбираем Dataset (модель на основе набора данных). В окне выбора соединение с базой данных выбираем «New Connection…» и в следующем окне выбираем тип соединения «Microsoft Access Database File (OLE DB)«. Рекомендуется файл базы данных сохранить в той самой папке, что и файлы приложения. В следующих окнах оставляем все без перемен. На вопрос Save the Connection String to the Application Configuration File даем ответ «Yes» и строка соединения оставляем без перемен.

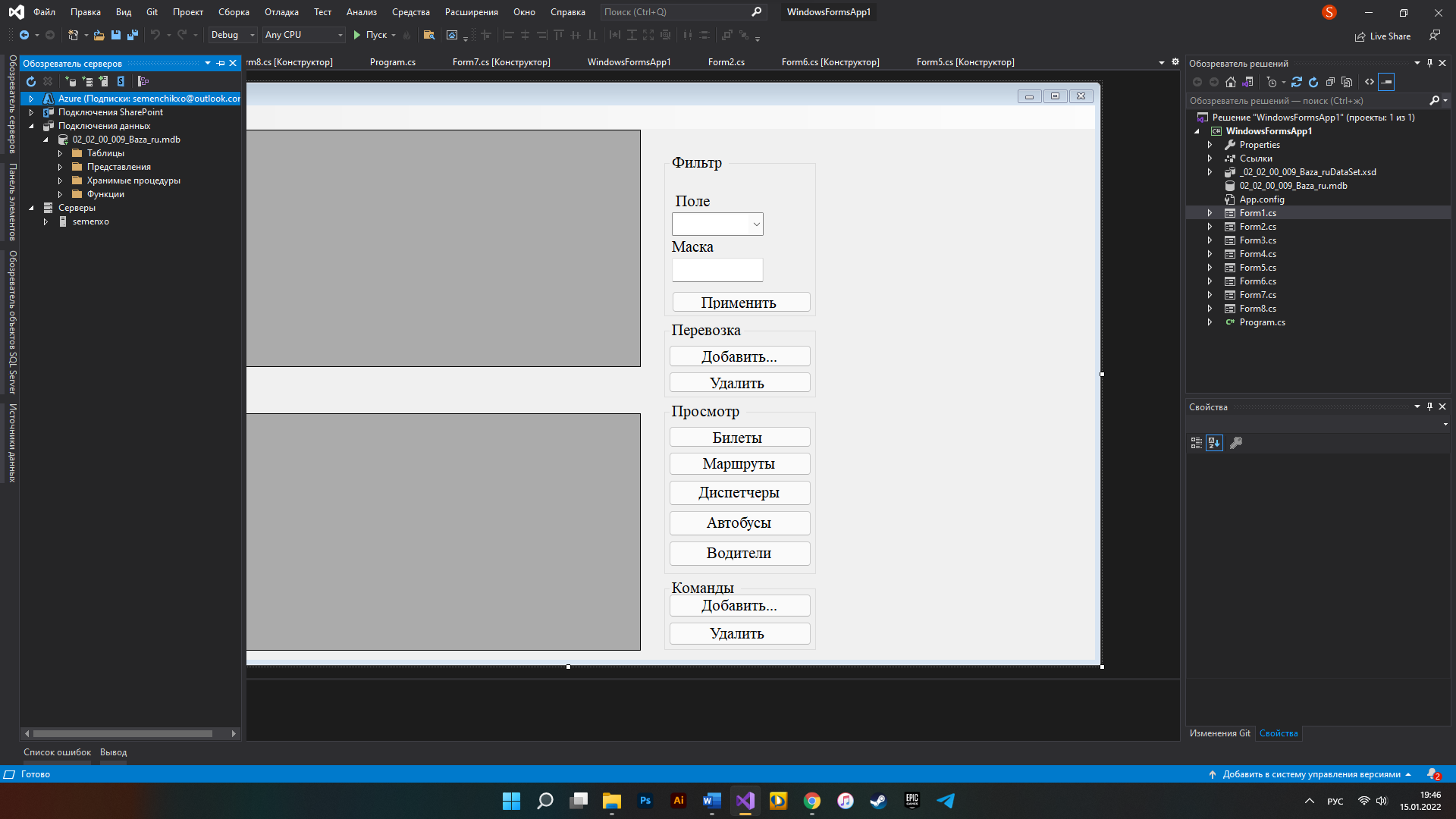


Рис. 1. Результат подключения к базе данных

Для того, чтобы программно управлять базой данных и получить доступ к методам, которые оперируют базой данных, нужно подключить пространство имен System.Data.OleDb.

Для этого, в основной форме Form1.cs в Solution Explorer выбираем режим просмотра кода (View Code) из контекстного меню и в начале файла добавляем следующую строку:

using System.Data.OleDb;

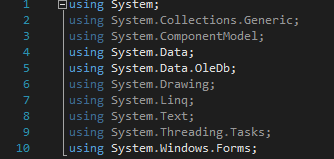


Рис. 2. Добавление строки

**4. Проектирование формы приложения**

Для проектирования формы используются элементы управления (controls) из панели Toolbox

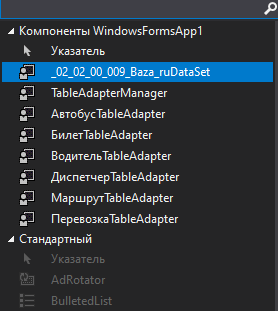


Рис. 3. Панель Toolbox

**4.1. Разработка меню**

Для построения меню используется элемент управления menuStrip из вкладки «Menus & Toolbars» панели ToolBox



Рис. 4. Элемент управления MenuStrip

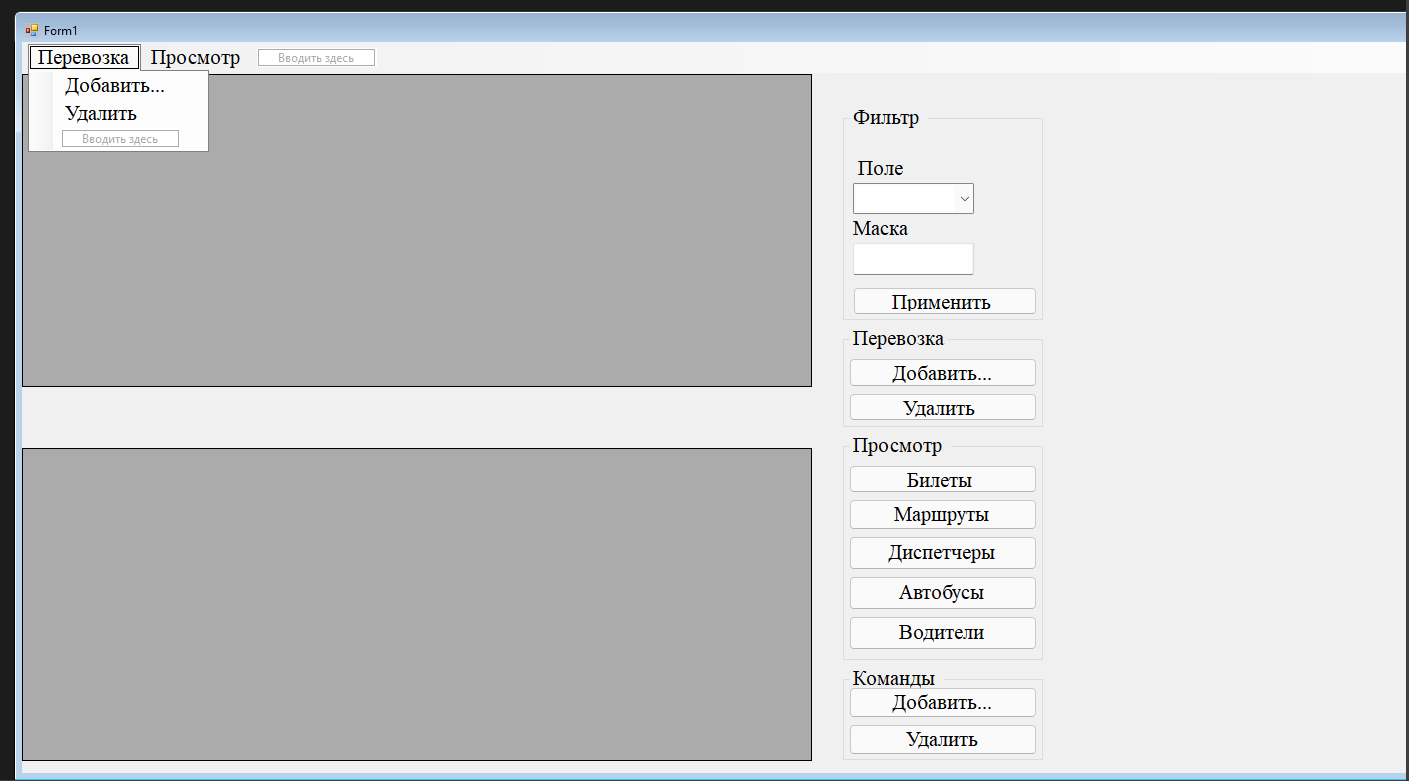


Рис. 5. Построение меню «Перевозка» и «Просмотр»

**4.2. Элементы управления DataGridView**

Изменяем размеры формы и выносим на форму два элемента управления DataGridView из вкладки Data панели Toolbox



Рис. 6. Элементы управления DataGridView

**4.3. Элементы управления типа GroupBox**

Активируем вкладку Containers из панели инструментов Toolbox.

Размещаем на форме 4 элемента управления типа GroupBox. В результате получаем 4 объекта с именами groupBox1, groupBox2, groupBox3, groupBox4. С помощью этих имен можно иметь доступ к свойствам (properties) и методам этих объектов.

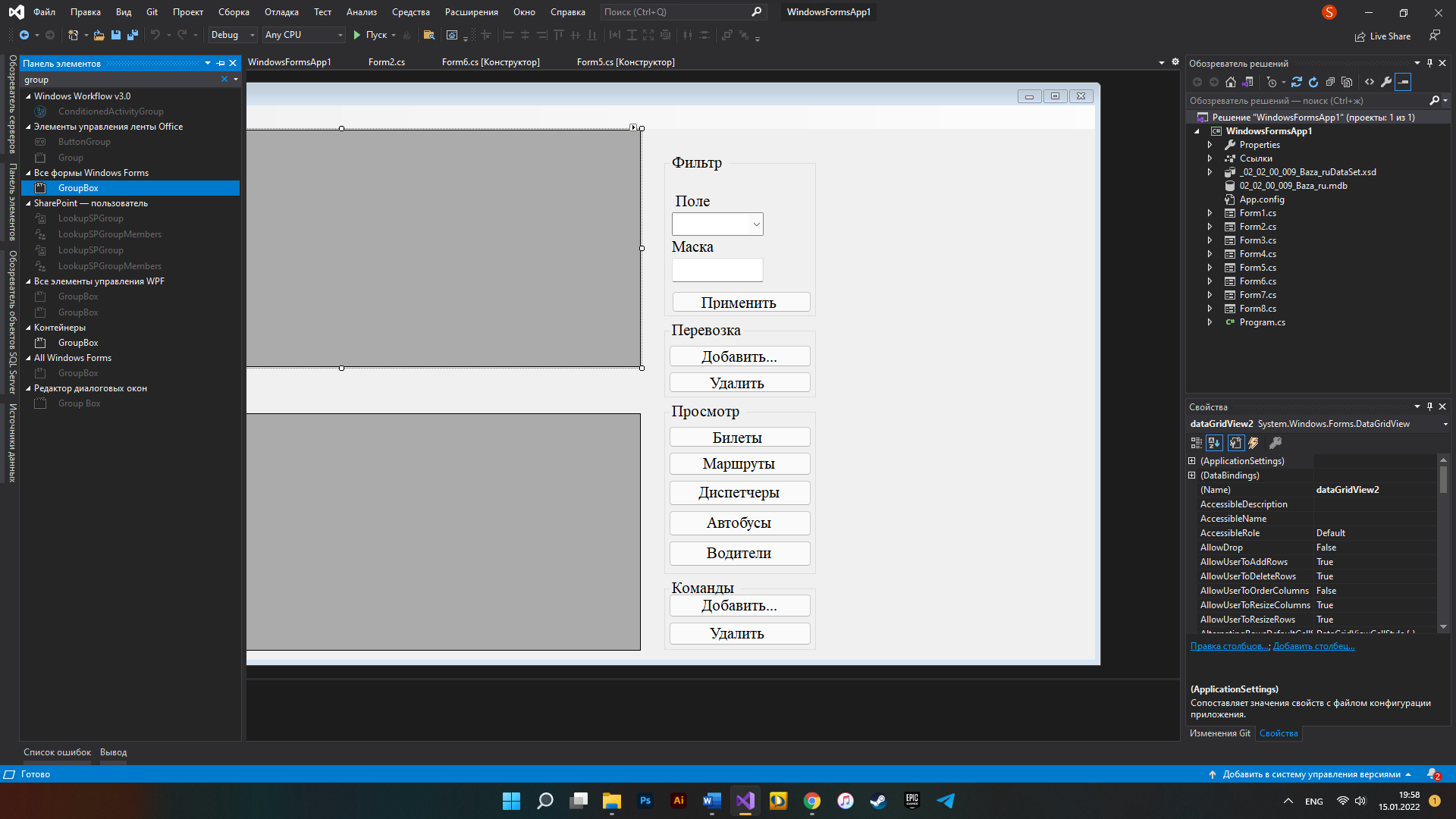


Рис. 7. Элементы управления типа GroupBox

Устанавливаем подпись для объекта groupBox1. Это осуществляется с помощью свойства Text. Значение свойства Text равно «Фильтр«. В этой группе будут размещаться объекты, которые отвечают за фильтрацию данных в базе данных.

Таким самым образом устанавливаем значения других трех объектов:  
– в объекте groupBox2 свойство Text = «Перевозка«;  
– в объекте groupBox3 свойство Text = «Просмотр«;  
– в объекте groupBox4 свойство Text = «Команды«.

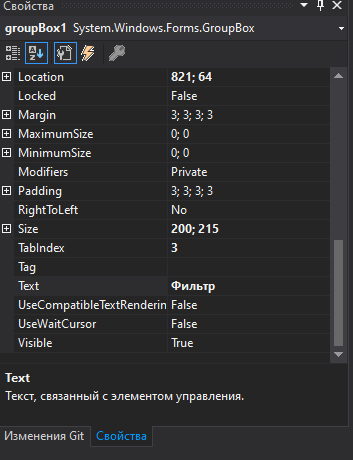


Рис. 8. Свойство Text объекта groupBox1

**4.4. Размещение элементов управления типа Label, Button, TextBox и ComboBox.**

Элементы управления типа Button и TextBox размещаются в вкладке «Common Controls» панели инструментов Toolbox.

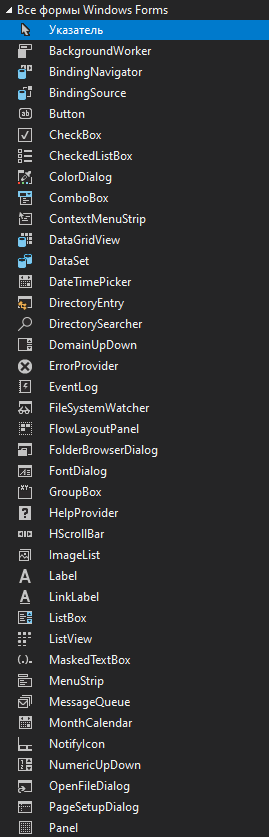


Рис. 9. Элементы управления Label, Button, TextBox и ComboBox из панели инструментов Toolbox

**4.5. Размещение элементов управления из группы «Фильтр»**

В области объекта groupBox1 («Фильтр«) размещаем один элемент управления типа ComboBox, два элемента типа Label, один элемент управления типа Button и один элемент управления типа TextBox

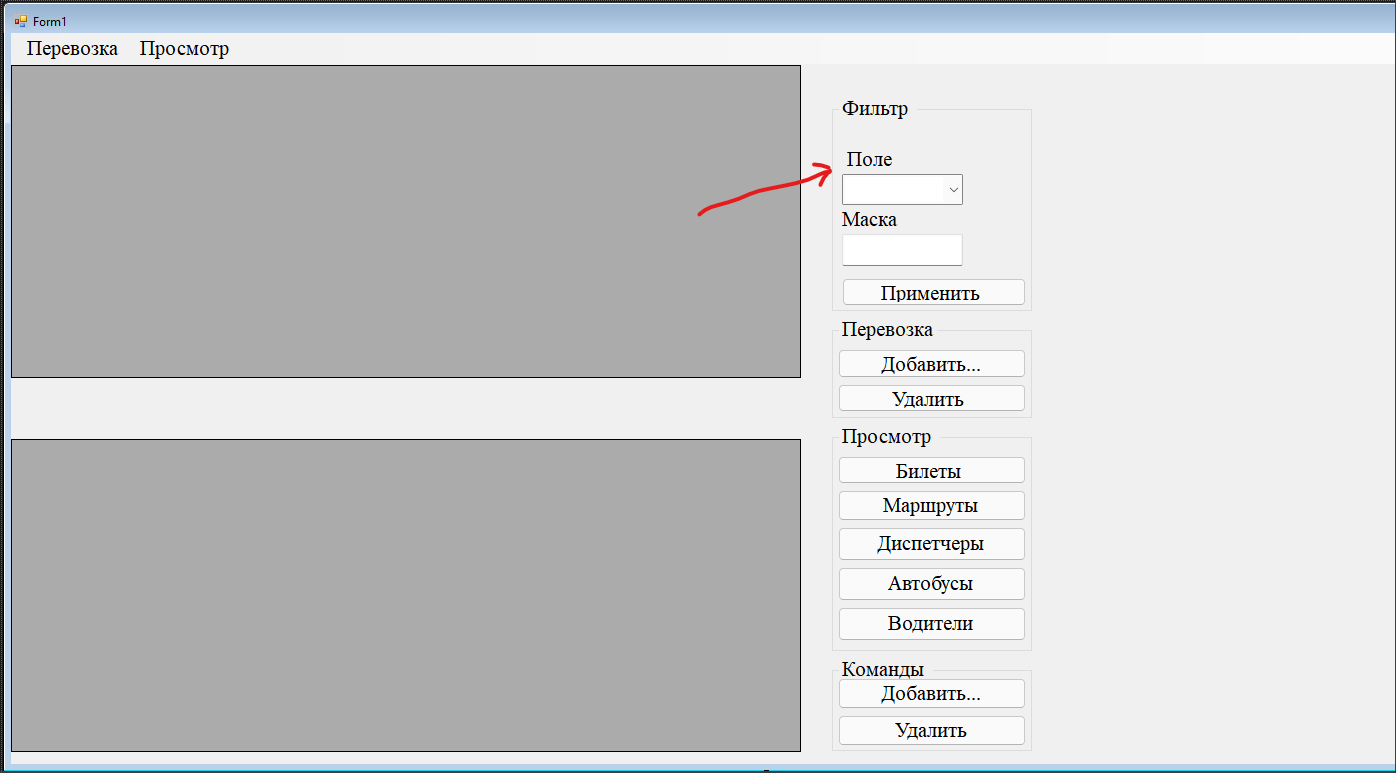


Рис. 10. Элементы управления label1, label2, button1, comboBox1, textBox1 в группе «Фильтр»

Устанавливаем следующие свойства элементов управления:

* в элементе управления label1 свойство Text равно значению «Поле«;
* в элементе управления label2 свойство Text = «Маска«;
* в button1 свойство Text = «Применить«.

Дополнительно изменяем размеры элементов управления button1 и textBox1 для более наглядного вида.

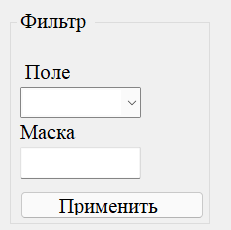


Рис. 11. Окно формы после формирования элементов управление с groupBox1

**4.6. Проектирование элементов управления, которые размещаются в группе «Перевозка»**

Таким же образом формируем группу «Перевозка» (groupBox2). Из панели Toolbox размещаем два элемента управления типа Button и формируем их свойства Text.. Будет создано два объекта типа Button с именами button2 и button3.

Свойство Text объекта button2 устанавливаем в значение «Добавить …». Свойство Text объекта button3 устанавливаем в значение «Удалить»

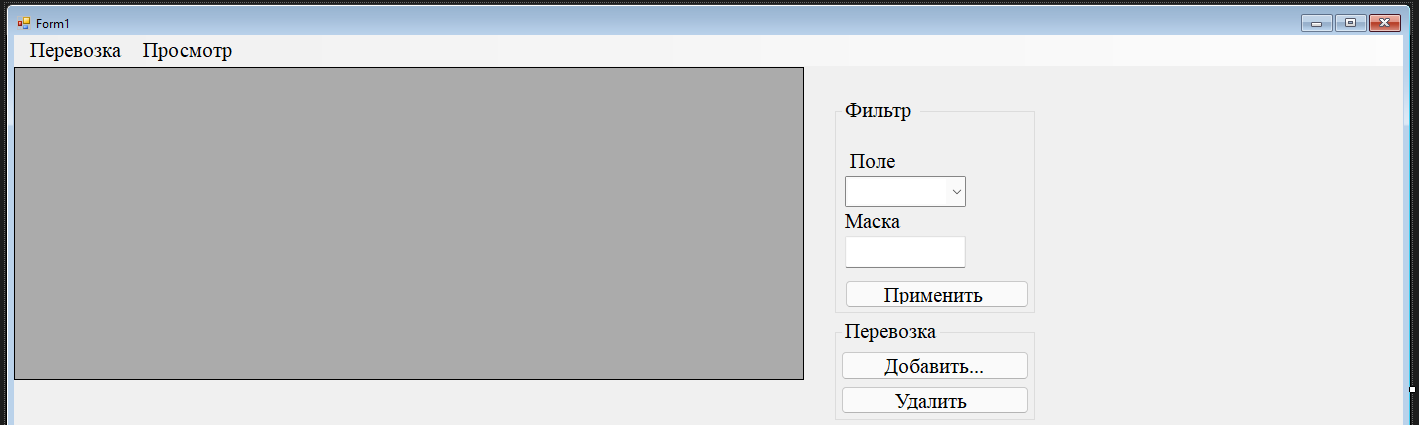


Рис. 12. Построение элементов управления, которые размещаются в группе «Перевозка»

**4.7. Проектирование элементов управления, которые размещаются в группе «Просмотр»**

Из панели Toolbox размещаем 5 элементов управления типа Button в области группы «Просмотр» (объект groupbox3).

Корректируем размеры этих элементов управления.

В результате будет создано 5 объектов с именами button4, button5, button6, button7, button8.

В каждом из этих объектов формируем свойство Text:  
– в объекте button4 свойство Text = «Билеты«;  
– в объекте button5 свойство Text = «Маршруты«;  
– в объекте button6 свойство Text = «Автобусы«;  
– в объекте button7 свойство Text = «Водители«;  
– в объекте button8 свойство Text = «Диспетчеры«.

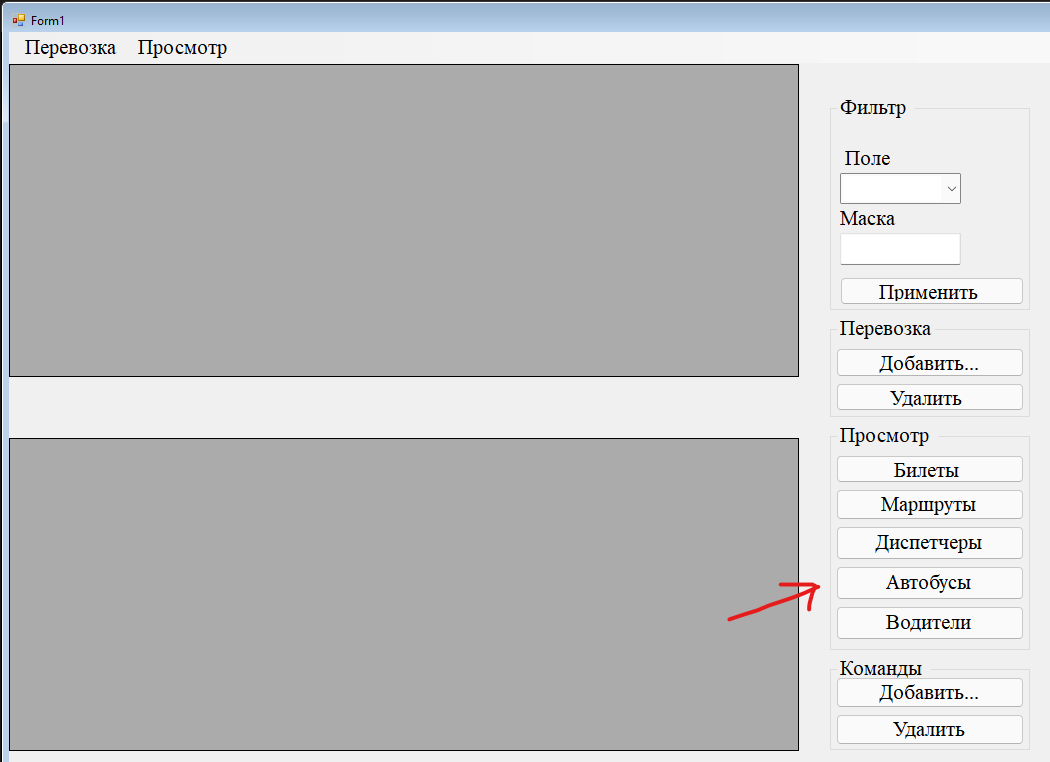


Рис. 13. Элементы управления из группы «Просмотр«

**4.8. Проектирование элементов управления, которые размещаются в группе «Команды»**

В группе «Команды» разместим два элемента управления типа Button.

Будет создано два объекта с именами button9 и button10.

Установим свойство Text этих объектов соответственно у значения «Добавить…» и «Удалить».

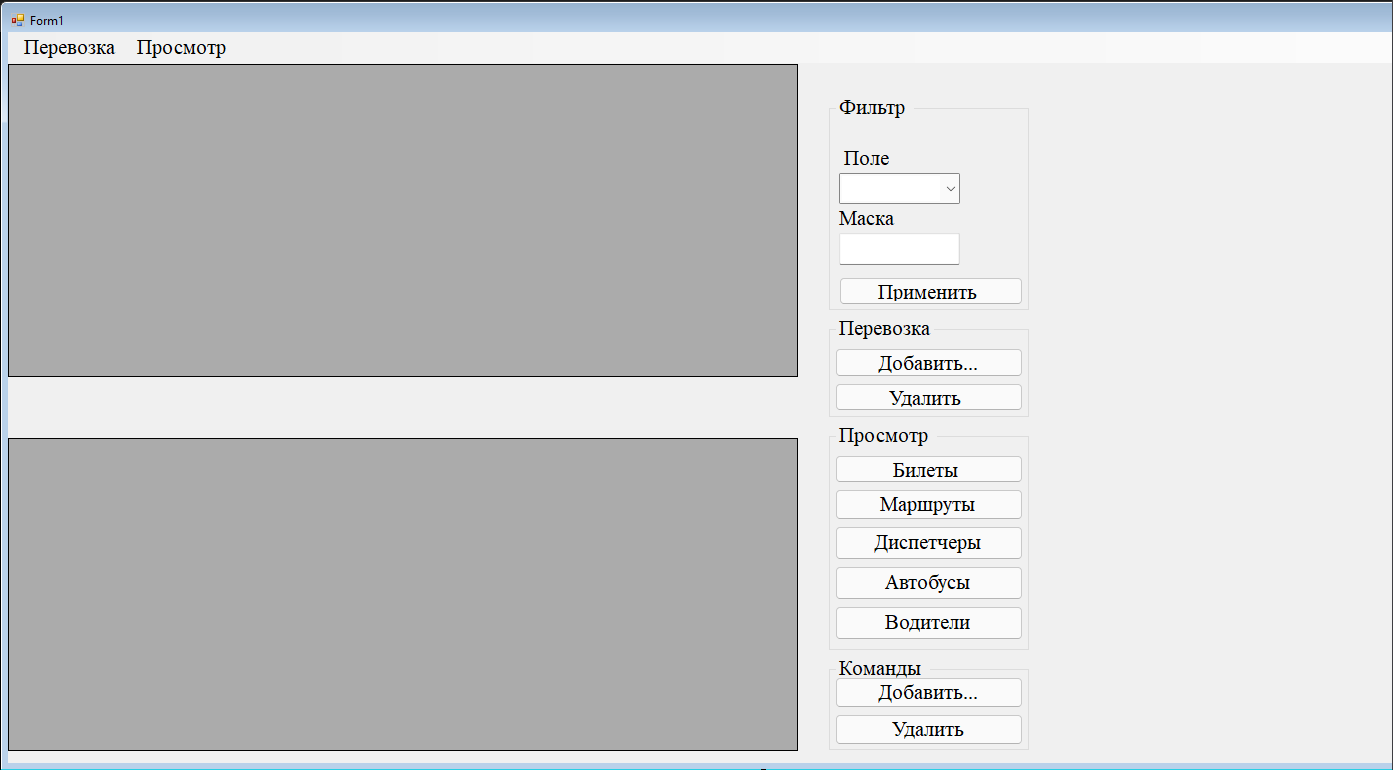


Рис. 14. Форма приложения после размещения всех элементов управления

**5. Формирование списка в элементе управления**

Элемент управления (объект) comboBox1, что используется в группе «Фильтр», должен содержать поля, к которым будет применяться фильтр для удобного вывода данных из базы данных.

В приложении, в будущем, фильтр будем применять к таким полям:

* номер перевозки;
* номер маршрута;
* пункт назначения;
* пассажир;
* водитель.

Для вызова процесса формирования списка вызовем редактор строк элемента управления comboBox1. Для этого, после активирования (выделения) comboBox1 делаем клик на кнопке «>» (стрелка вправо). Откроется меню, в котором выбираем «Edit Items…».

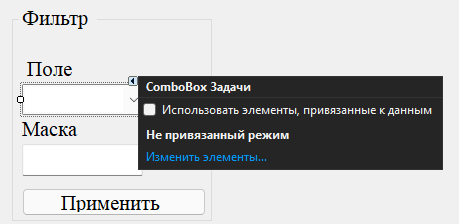


Рис. 15. Вызов меню формирования списка в comboBox1

В результате откроется окно «String Collection Editor», в котором заносим названия полей, для которых будет установлен фильтр.

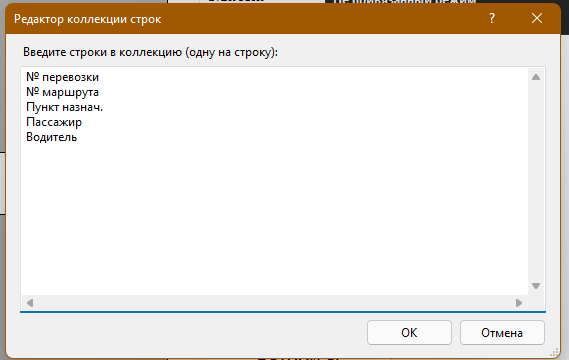


Рис. 16. Окно редактора строк элемента управления comboBox1

После нажатия на «OK» будут сформированы строки элемента управление comboBox1.

**6. Получение свойства ConnectionString**

В следующих программных кодах, где выполняются операции с базой данных, фигурирует свойство «ConnectionString«. Это свойство нужно уметь прочитать правильно.

Свойство «ConnectionString» содержит информацию о провайдере данных и о размещении файла базы данных на диске.

Для получения текста свойства ConnectionString нужно выполнить такие шаги:

* в окне Server Explorer выделить базу данных «[02\_02\_00\_009\_Baza\_ru.mdb](https://www.bestprog.net/wp-content/uploads/2016/03/02_02_00_009_Baza_ru.mdb)«;
* в окне Properties выбрать свойство Connection String и скопировать его в буфер обмена Clipboard.

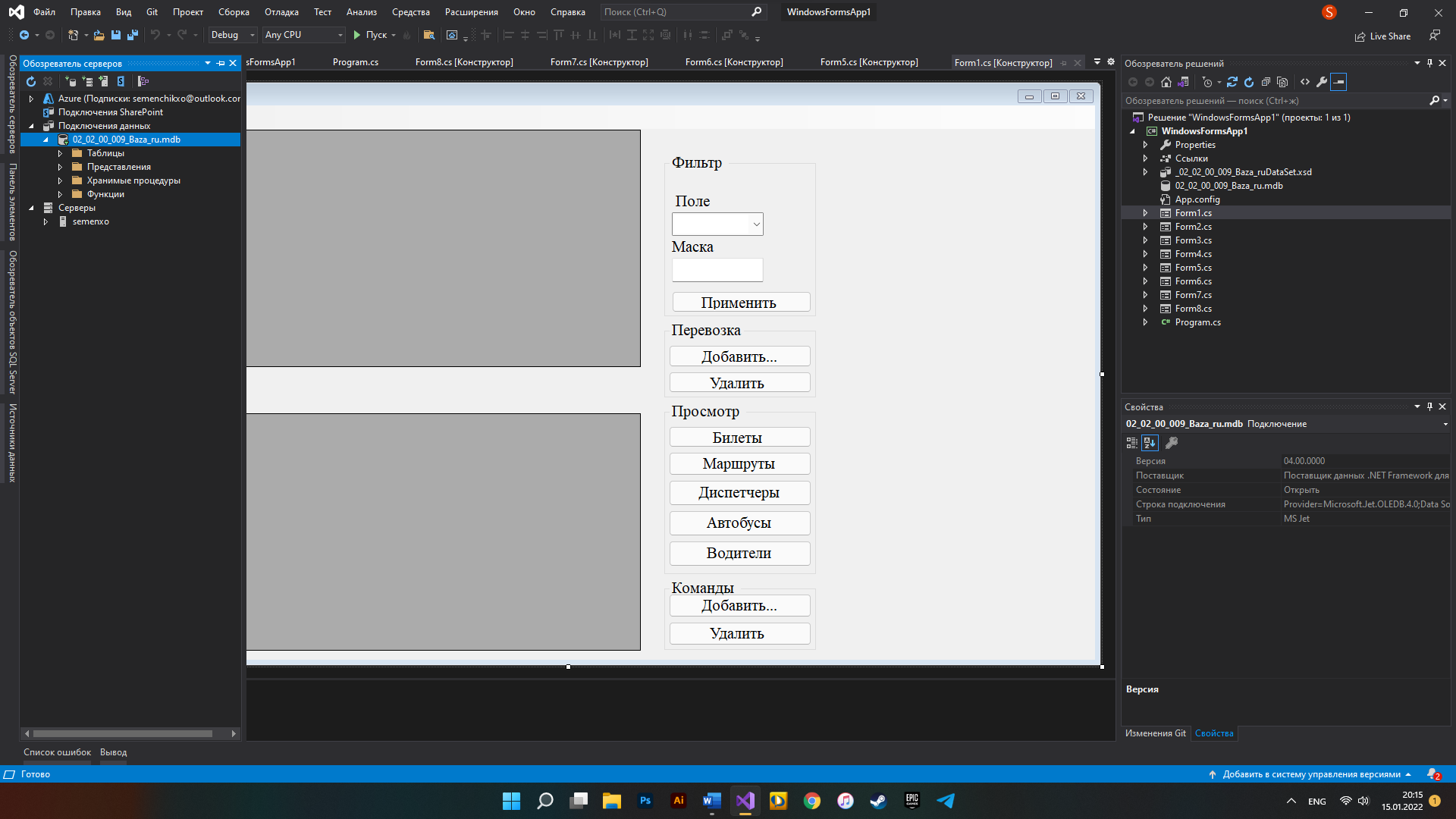


Рис. 17. Чтение свойства «ConnectionString«

В нашем случае, значение свойства «ConnectionString» равно:

Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Programs\C\_SHARP\Program\_02\_02\_00\_009r\WindowsFormsApplication1\WindowsFormsApplication1\02\_02\_00\_009\_Baza\_ru.mdb

*Следует заметить следующее*. При использовании строки «ConnectionString» в тексте программы на языке C#, символ ‘\‘ нужно заменить на ‘\\‘ в соответствии с синтаксисом языка.

**7. Добавление внутренних переменных общего доступа к форме «Form1.cs»**

Основной форме приложения соответствует класс Form1. При написании программного кода в классе будут использованы некоторые внутренние переменные общего назначения. Это переменные «ConnectionString» и act\_table.

Первая переменная будет содержать строку соединения с базой данных «ConnectionString«.

Вторая переменная содержит номер активной таблицы, которая будет выводиться в dataGridView2. Соответствие между номерами и таблицами базы данных следующее:

* 1 – таблица «Билет«;
* 2 – таблица «Маршрут«;
* 3 – таблица «Автобус«;
* 4 – таблица «Водитель«;
* 5 – таблица «Диспетчер«.

Осуществим добавление текста свойства «ConnectionString» в текст класса главной формы программы.

С помощью Solution Explorer откроем файл Form1.cs командой «View Code»

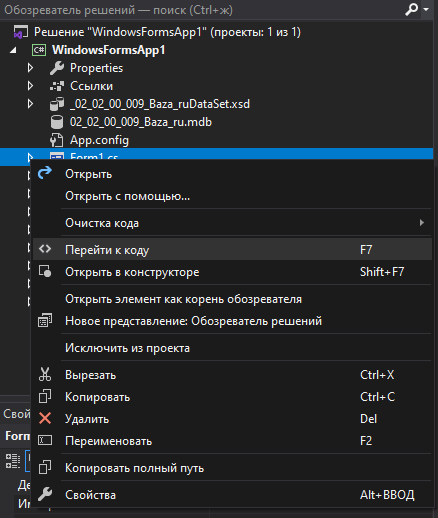


Рис. 18. Вызов команды перехода к просмотру программного кода формы Form1

Листинг файла формы следующий:

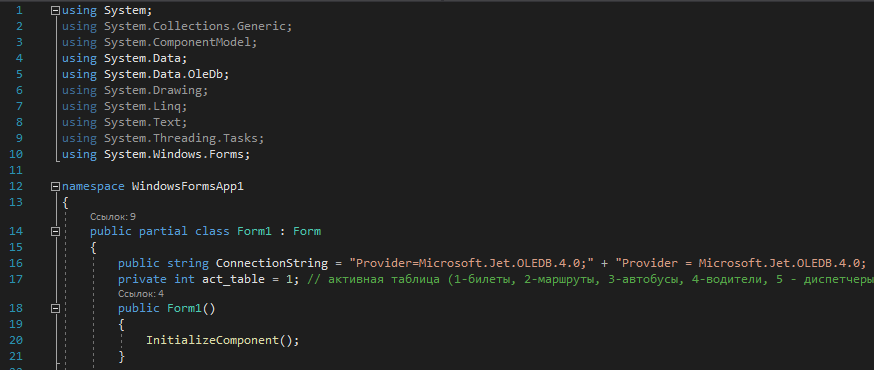
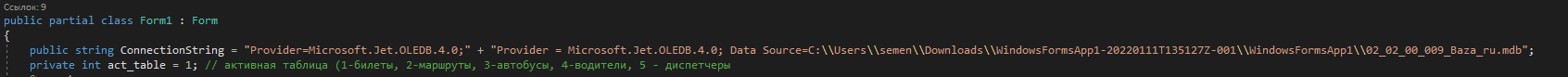
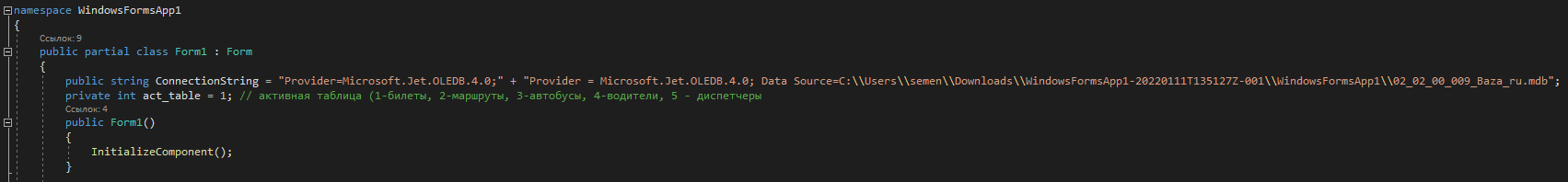


Рис. 19. Листинг файла формы

Модифицируем этот листинг таким образом, чтобы в нем были добавлены две переменные ConnectionString и act\_table. Переменную ConnectionString описываем как public, так как она будет использоваться в других классах (формах). Для удобного отображения в окне проектирования, строка переменной ConnectionString разбита на части



Листинг класса Form1 основной формы приложения имеет вид:



**8. Вывод данных в dataGridView1**

Данные которые будут выводиться в компоненте dataGridView1 будут получены из разных таблиц. Чтобы происходило фильтрование данных, нужно учитывать строку, которая выделена в comboBox1 и маску которая набрана в элементе управления textBox1. В comboBox1 пользователь имеет возможность выбирать поле, на основании которого будет происходить фильтрация данных.

**8.1. Текст SQL-запроса**

В таблице приведены поля, которые будут выводиться в dataGridView1.

[](https://www.bestprog.net/wp-content/uploads/2016/03/02_02_00_009_table07r.jpg)

Поля, которые нужно выводить в соответствии с синтаксисом языка SQL следующие:

 [Перевозка].[Номер]

[Маршрут].[Номер маршрута]

[Маршрут].[Пункт назначения]

[Маршрут].[Время отправки]

[Маршрут].[Время прибытия]

[Билет].[Место]

[Билет].[Ф\_И\_О]

[Билет].[Стоимость]

[Водитель].[Ф\_И\_О]

Запрос на языке SQL для построения новой таблицы имеет следующий вид:

**SELECT**

[Перевозка].[Номер]

[Маршрут].[Номер маршрута]

[Маршрут].[Пункт назначения]

[Маршрут].[Время отправки]

[Маршрут].[Время прибытия]

[Билет].[Место]

[Билет].[Ф\_И\_О]

[Билет].[Стоимость]

[Водитель].[Ф\_И\_О]

**FROM**

[Перевозка],

[Маршрут],

[Билет],

[Водитель]

**WHERE**

([Перевозка].[ID\_Marshrut]=[Маршрут].[ID\_Marshrut]) AND

([Перевозка].[ID\_Bilet] = [Билет].[ID\_Bilet]) AND

([Перевозка].[ID\_Voditel] = [Водитель].[ID\_Voditel])

В этом запросе не учитывается фильтр. Для учета фильтра, к запросу нужно добавить соответствующую строку. В приложении эта строка будет добавляться программно.

Например, для добавления фильтра по номеру перевозки, к тексту SQL-запроса нужно добавить следующую строку:

**AND** ([Перевозка].[Номер] = '" + textBox1.Text + "')

где textbox1.Text – текст фильтра в компоненте textbox1.

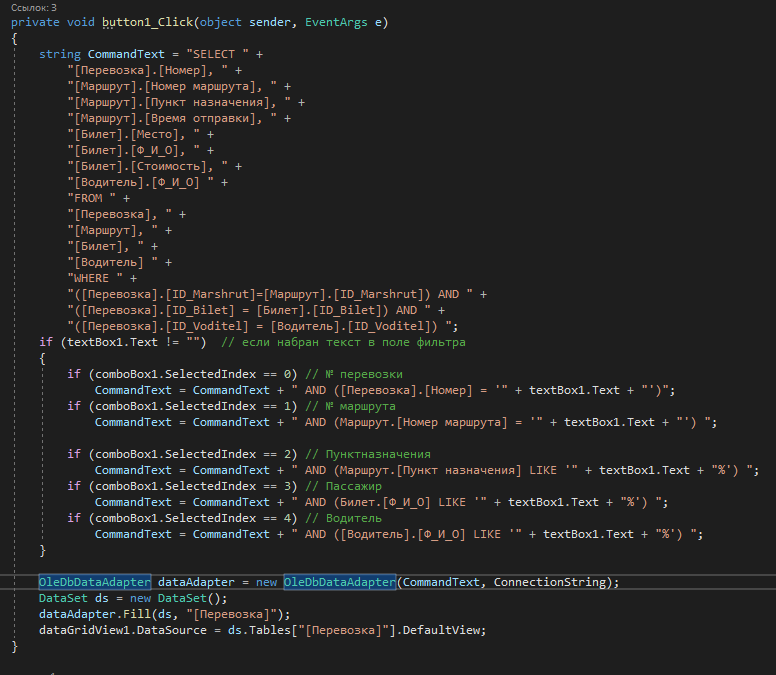
**8.2.** [**Программирование события**](https://www.bestprog.net/ru/2016/01/08/005-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%bc%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f-%d1%81%d0%be%d0%b1%d1%8b%d1%82%d0%b8%d1%8f-%d0%b2-c/) **клика на кнопке «Применить»**

Вызов вывода данных на основе фильтра осуществляется после нажатия на кнопке «Применить».

Процесс программирования события в MS Visual Studio подробно описан [здесь](https://www.bestprog.net/ru/2016/01/08/005-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%bc%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f-%d1%81%d0%be%d0%b1%d1%8b%d1%82%d0%b8%d1%8f-%d0%b2-c/). Вывод базы данных в элементе управления dataGridView описывается [здесь](https://www.bestprog.net/ru/2015/12/22/002-%d0%b2%d1%8b%d0%b2%d0%be%d0%b4-%d1%82%d0%b0%d0%b1%d0%bb%d0%b8%d1%86%d1%8b-%d0%b1%d0%b0%d0%b7%d1%8b-%d0%b4%d0%b0%d0%bd%d0%bd%d1%8b%d1%85-microsoft-access-%d0%b2-%d0%ba%d0%be%d0%bc%d0%bf%d0%be/).

Программируем событие клика на кнопке «Применить».

Обработчик события клика на кнопке «Применить» имеет следующий вид.



Сначала, в переменную CommandText, добавляется строка SQL-запроса которая выводит все данные из базы данных. Потом, в зависимости от фильтра, в значение CommandText дописывается соответствующая строка, завершающая SQL-команду.

После построения CommandText, создается объект типа OleDbDataAdapter, который пересылает наборы данных с вызывающего процесса и обратно. Адаптеры данных содержат набор с четырех внутренних объектов команд. Это команды чтения, вставки, изменения и удаления информации. Как видно из программного кода, конструктор объекта получает входными параметрами строку запроса на языке SQL и строку подключения к базе данных. Таким образом объект адаптера данных связан с нашей базой данных.

После создания адаптера данных (OleDbDataAdapter) создаем объект типа DataSet (набор данных):

DataSet ds = new DataSet();

Набор данных представляет собой своего рода промежуточный буфер для данных, которые могут отображаться. Набор данных представляет удобный механизм чтения и обновления данных и инкапсулирует множество таблиц и связей между ними.

Следующая команда – это заполнение набора данных (переменная ds) значениями записей из базы данных на основании SQL-запроса, который помещается в адаптере данных ds помощью метода Fill():

dataAdapter.Fill(ds, "[Перевозка]");

Для отображения (визуализации) данных, сформированных на основе SQL-запроса, нужно чтобы свойство DataSource компонента dataGridView1 ссылалось на некоторую таблицу набора данных ds. Например, таблицу «Перевозка«. Программный код операции отображения данных имеет следующий вид:

dataGridView1.DataSource = ds.Tables["[Перевозка]"].DefaultView;

После этого данные из таблицы «Перевозка» отобразятся на форме.

**8.3. Добавление метода button1\_Click() в метод Form1\_Load()**

Метод Form\_Load() вызывается в момент загрузки формы. Дописываем в этот метод вызов метода обработки события button1\_Click().

На данный момент листинг метода следующий.

private void Form1\_Load(object sender, EventArgs e)

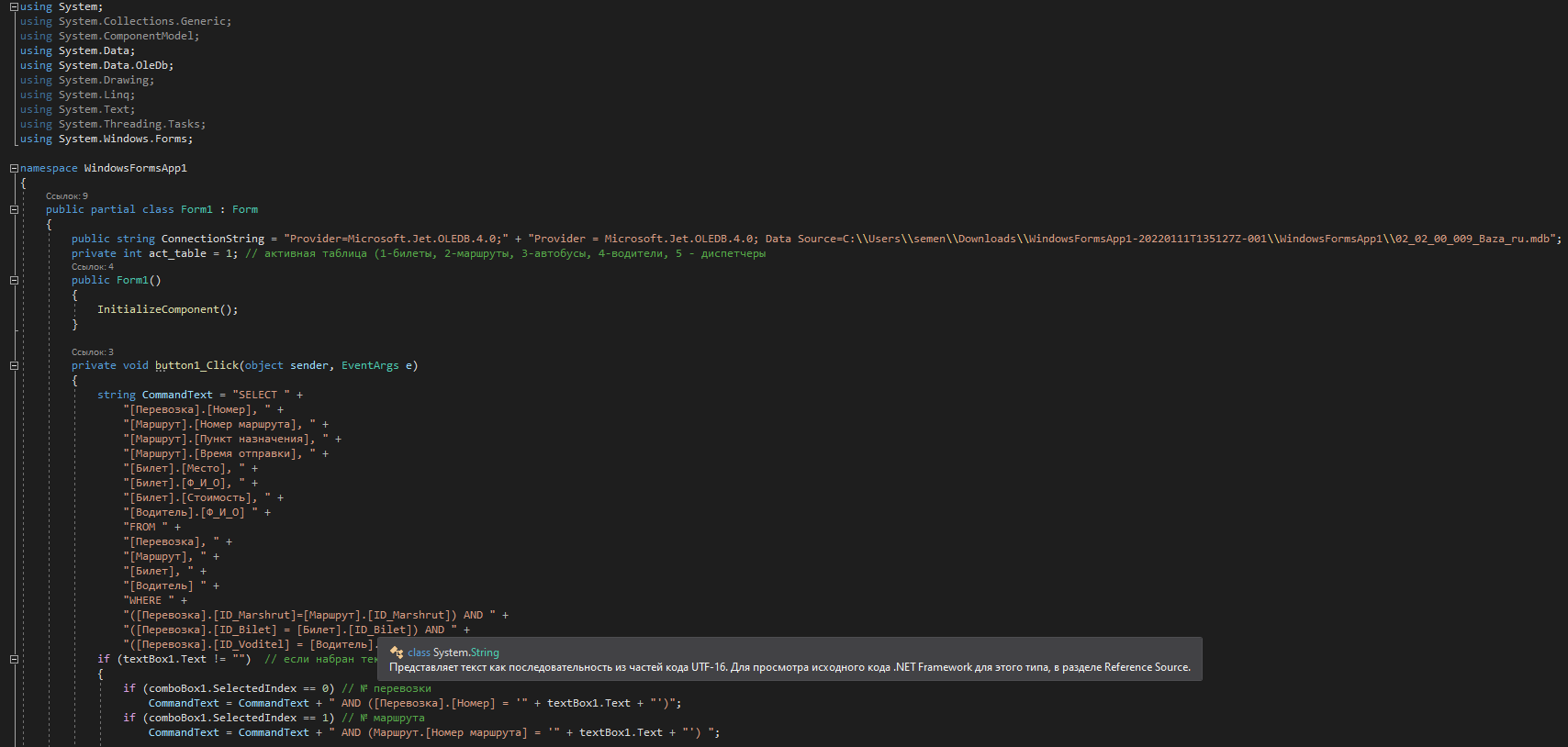
{

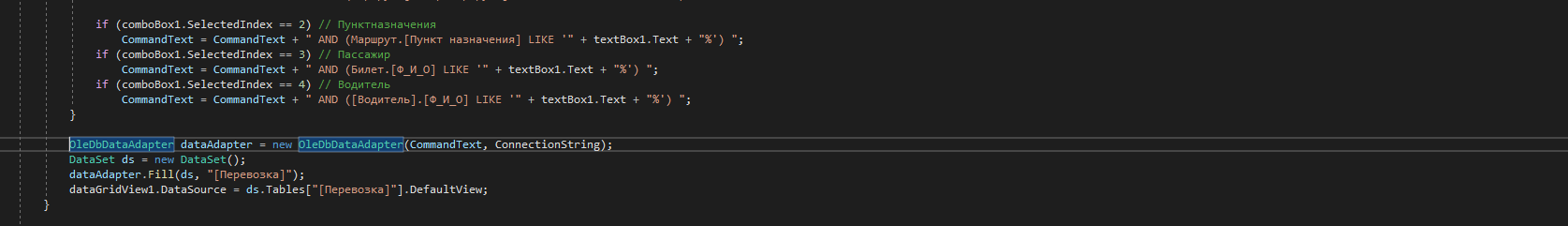
comboBox1.SelectedIndex = 0;

button1\_Click(sender, e);

}

**Итоговый листинг код:**





**9. Разработка формы добавления перевозки и создание его программного кода**

Добавление новой перевозки выполняется нажатием кнопки «Добавить…» из группы «Перевозка«. Для того, чтобы создавать поля, которые используются в отображении перевозки (элемент управление dataGridView1), нужно создать соответствующую форму.

**9.1. Создание формы «Form2.cs»**

Создание новой формы выполняется командой

Project -> Add Windows Form...

В открывшемся окне, выбираем «Windows Form«. Имя формы оставляем «Form2.cs«.

В результате, в «Solution Explorer» получим файлы «Form2.cs» и «Form2.Designer.cs«.

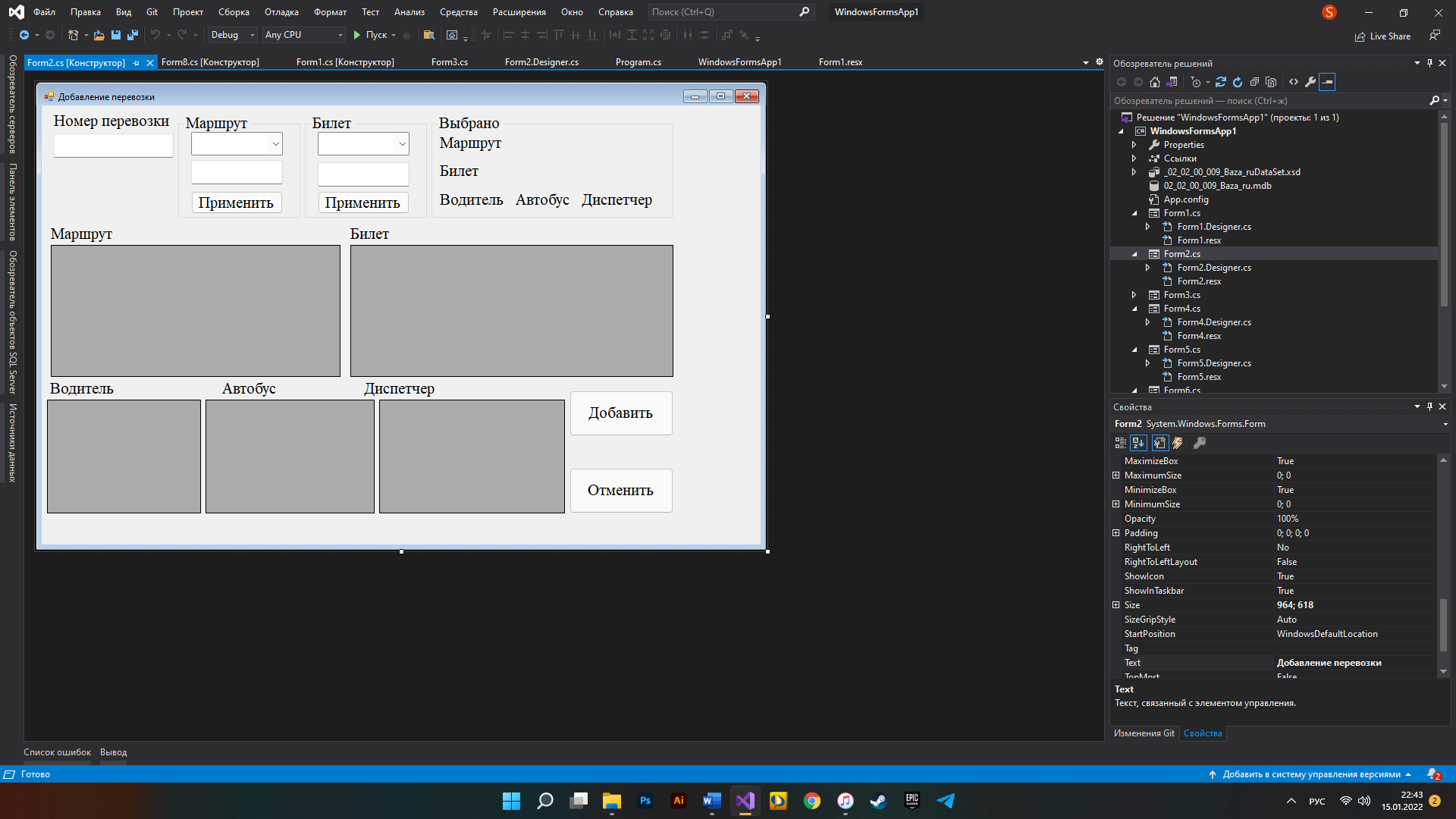


Рис. 20. Создание формы «Form2.cs»

Изменяем размеры формы «Form2.cs» и размещаем на форме элементы управления. Красным цветом выделены названия элементов управления, которые будут использоваться в программном коде.

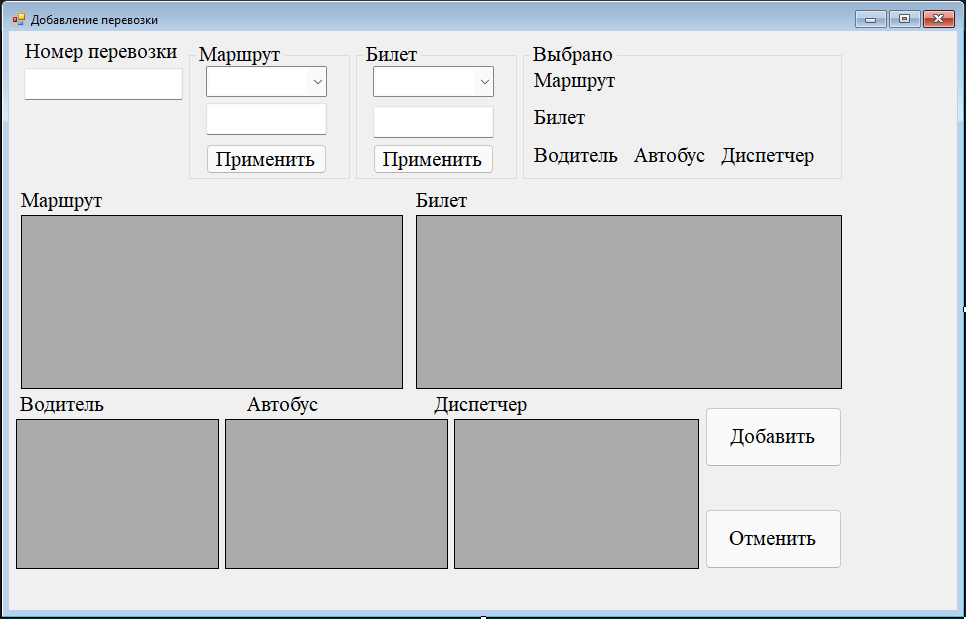
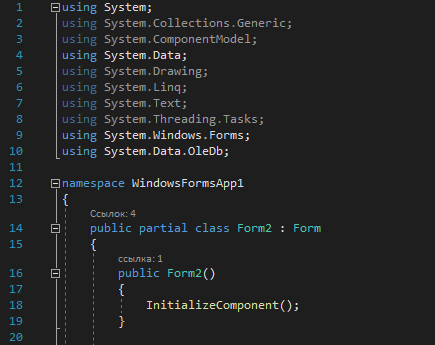


Рис. 21. Форма «Form2.cs» с описанием названий элементов управления

После создания самой формы будет создан класс формы с именем Form2. Листинг описания класса формы Form2 следующий:

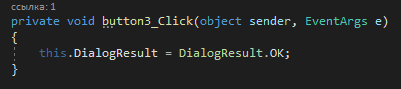


**9.2. Программирование событий клика на кнопках «Добавить» и «Отменить»**

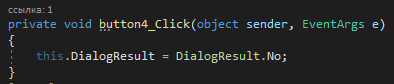
Вызовем событие Click, которое генерируется при клику на кнопке «Добавить» формы Form2. [Процесс программирования нового события](https://www.bestprog.net/ru/2016/01/08/005-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%bc%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f-%d1%81%d0%be%d0%b1%d1%8b%d1%82%d0%b8%d1%8f-%d0%b2-c/) подробно описан [здесь](https://www.bestprog.net/ru/2016/01/08/005-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d0%bf%d1%80%d0%be%d0%b3%d1%80%d0%b0%d0%bc%d0%bc%d0%b8%d1%80%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d1%8f-%d1%81%d0%be%d0%b1%d1%8b%d1%82%d0%b8%d1%8f-%d0%b2-c/).

В результате, в классе формы будет сгенерирован код, в котором нужно вставить текст обработчика события.

Листинг метода обработки события клика на кнопке «Добавить» следующий:



Листинг метода обработки события клика на кнопке «Отменить»:



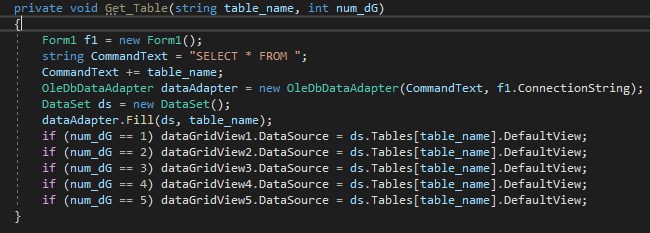
**9.3. Программирование методов отображения данных в элементах управления dataGridView1, dataGridView2, dataGridView3, dataGridView4, dataGridView5**

Для того, чтобы иметь доступ к методам и свойствам которые используются при работе с базами данных, в файле «Form2.cs» подключаем пространство имен:

using System.Data.OleDb;

Для отображения данных в элементах управления типа DataGridView создаем дополнительный метод Get\_Table(), который получает входящими два параметра. Параметр table\_name задает название таблицы в базе данных. Параметр num\_dG задает номер элемента управления типа DataGridView – от 1 до 5. Например, если num\_dG=2, то это значит, что отображаются данные в элементе управления dataGridView2.

Листинг метода Get\_Table() следующий:



Следующим шагом программируем событие Load, которое вызовется при загрузке формы Form2. Этому событию буде соответствовать метод Form2\_Load().

**9.4. Программирование обработчиков событий изменения активной ячейки в компонентах dataGridView1, dataGridView2, dataGridView3, dataGridView4, dataGridView5**

Во время изменения активной ячейки в элементах управления типа DataGridView, нужно выводить соответствующий текст в элементах управления:

* label2 (Маршрут);
* label3 (Билет);
* label4 (Водитель);
* label5 (Автобус);
* label6 (Диспетчер).

При изменении активной ячейки в элементах управления типа DataGridView генерируется событие CellEnter.

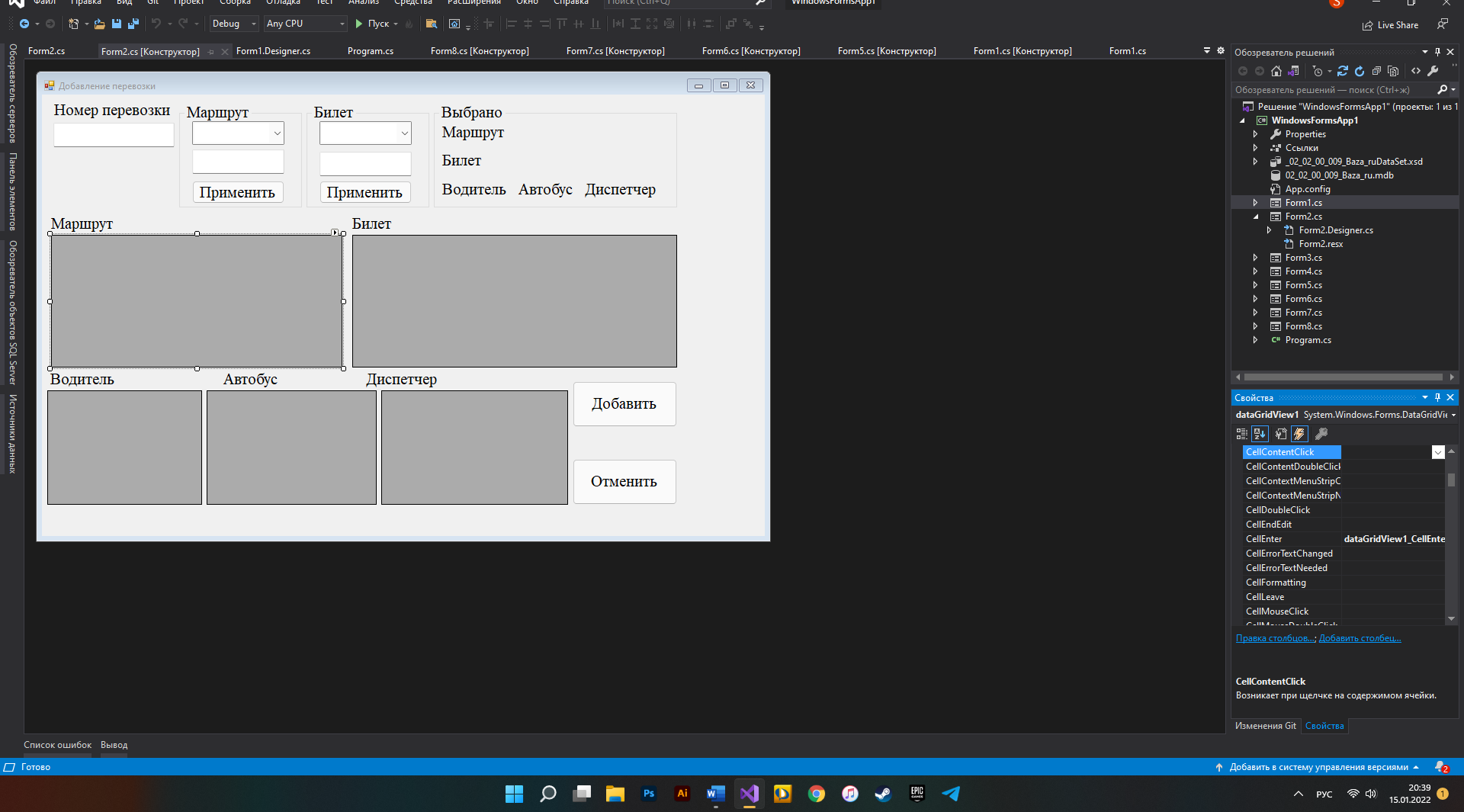
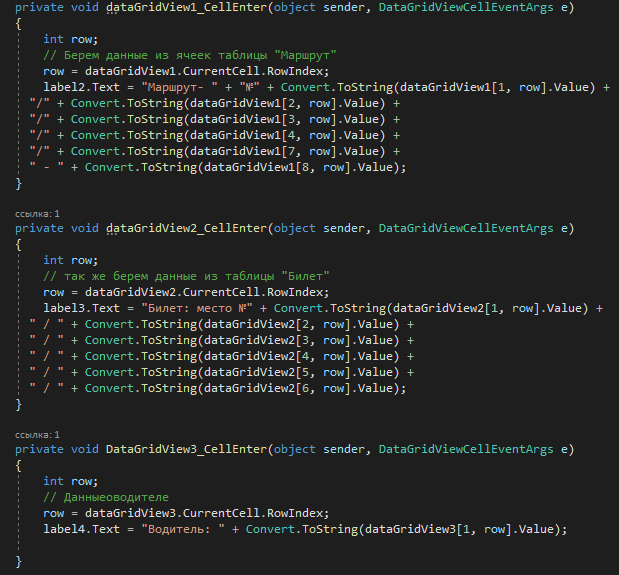
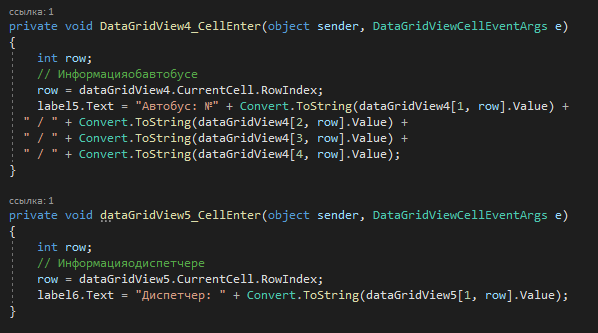


Рис. 22. Событие CellEnter элемента управления dataGridView1

Листинг обработчиков событий CellEnter для элементов управления dataGridView1, dataGridView2, dataGridView3, dataGridView4, dataGridView5 следующий.





**9.5. Построение строк полей для фильтра в элементах управления comboBox1 и comboBox2**

В компоненте comboBox1 пользователь имеет возможность выбирать поля из таблицы «Маршрут», которые будут использоваться при фильтрации данных.

Формируем такие строки в comboBox1:

* № маршрута;
* Пункт назначения;
* Район;
* Область.

В comboBox2 формируем такие строки:

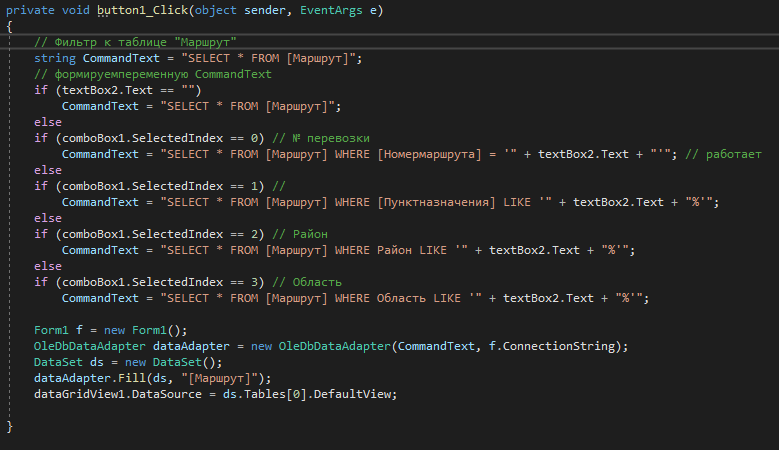
* Место;
* Ф. И. О. пассажира.

**9.6. Программирование фильтра к таблице «Маршрут»**

В форме Form2 можно применять фильтр к таблицам «Маршрут» и «Билет«.

Применение фильтра к таблице «Маршрут» реализовано с помощью компонент comboBox1, textBox2 и button1.

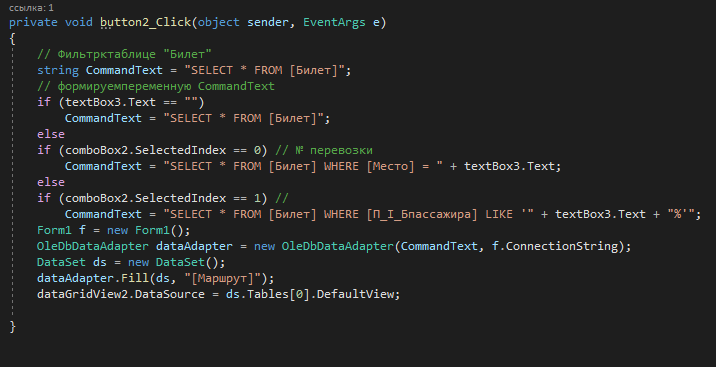
Листинг метода обработки события клика на кнопке «Применить» при выборе фильтра для таблицы «Маршрут»:



**9.7. Программирование фильтра к таблице «Билет»**

Фильтр к таблице «Билет» применяется при нажатии на кнопке «Применить» в группе «Билет». Элемент управления кнопки имеет название button2. Кроме того, для организации фильтра используются элементы управления comboBox2 и textBox3.

В поле textBox3 указывается значение фильтра. В comboBox2 указывается поле из таблицы «Билет«, к которому применяется фильтр.

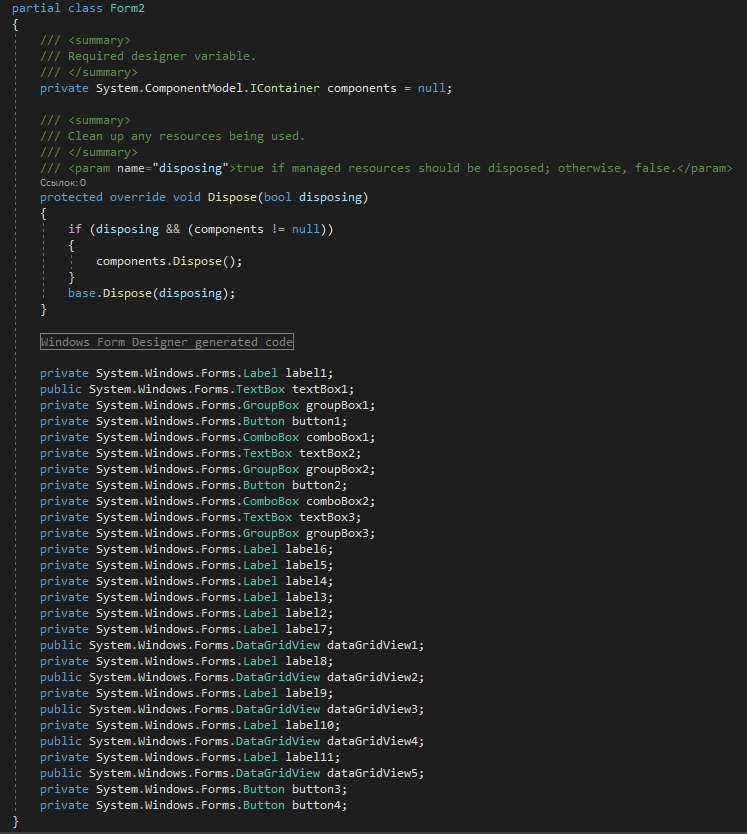


**9.8. Установление доступа к некоторым полям формы Form2**

Для того, чтобы из формы Form1 считывать значения некоторых элементов управления формы Form2, нужно изменить директиву видимости с private на public.

Чтобы осуществить это, вызовем файл «Form2.Designer.cs»

В результате, получим текст файла, который имеет приблизительно следующий вид:



Изменяем доступ с private на public для таких свойств:

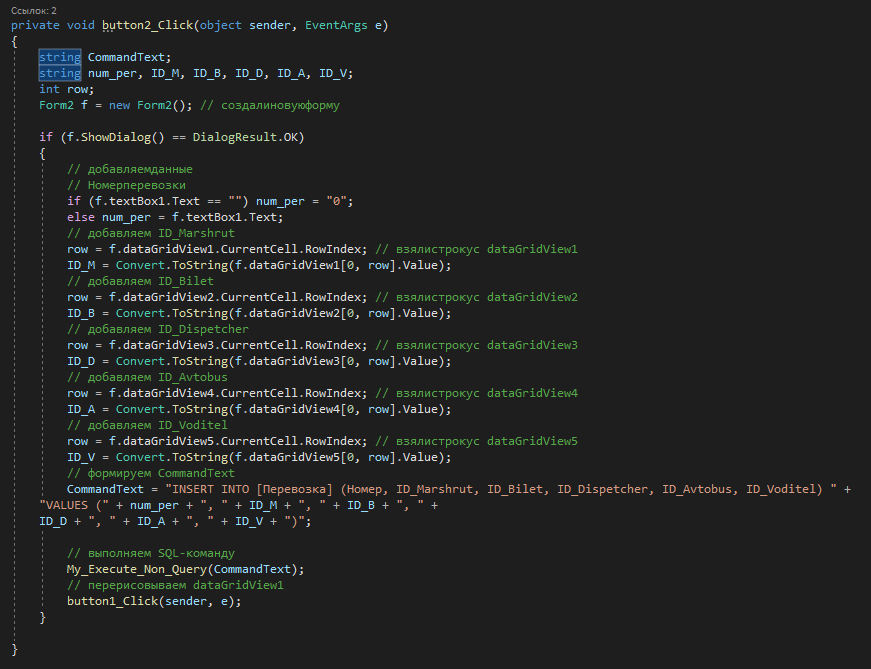
* textBox1 (№ перевозки);
* dataGridView1;
* dataGridView2;
* dataGridView3;
* dataGridView4;
* dataGridView5;

На этом создание формы «Form2.cs» завершено

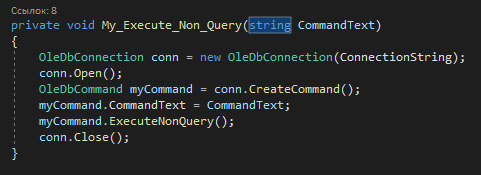
**10. Программирование события вызова формы Form2 из формы Form1**

Форма Form2 вызовется при выборе пользователем кнопки «Добавить…» из группы «Перевозка» (см. рис. 21). Элемент управления типа «кнопка», отвечающий команде «Добавить…», имеет название button2.

Листинг события вызова формы Form2 из формы Form1 имеет вид:



В методе button2\_Click() вызовется метод My\_Execute\_Non\_Query(), листинг которого приведен ниже:



В методе My\_Execute\_Non\_Query() создается объект, который осуществляет соединение с базой данных. Затем происходит выполнение команды, которая вносит изменения в базу данных. Для этого, в C# предназначен метод ExecuteNonQuery().

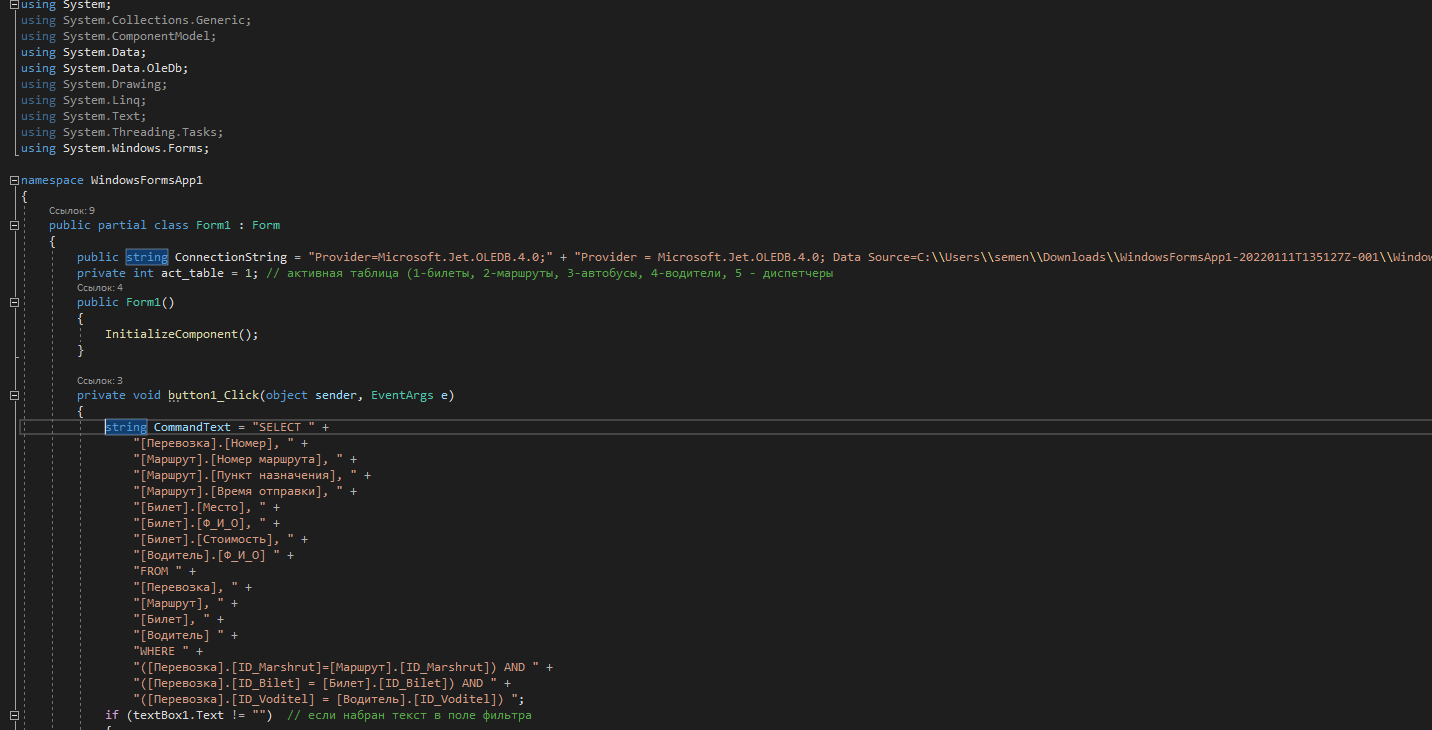
Команды на языке SQL делятся на два типа:

* те, которые читают данные из базы данных;
* те, что вносят изменения в данные базы данных.

После этого можно протестировать приложение. Можно вызвать команду «Добавить…» из группы «Перевозка» и добавить строку перевозки в таблицу, которая отображается в dataGridView1.

**11. Листинг модуля Form1.cs**

На данный момент, листинг файла «Form1.cs» в сокращенном виде следующий:





**12. Создание формы удаления записи в dataGridView1**

В случае, если пользователь выбирает кнопку «Удалить» (button3) из группы «Перевозка«, нужно запрограммировать метод обработки события клика на этой кнопке

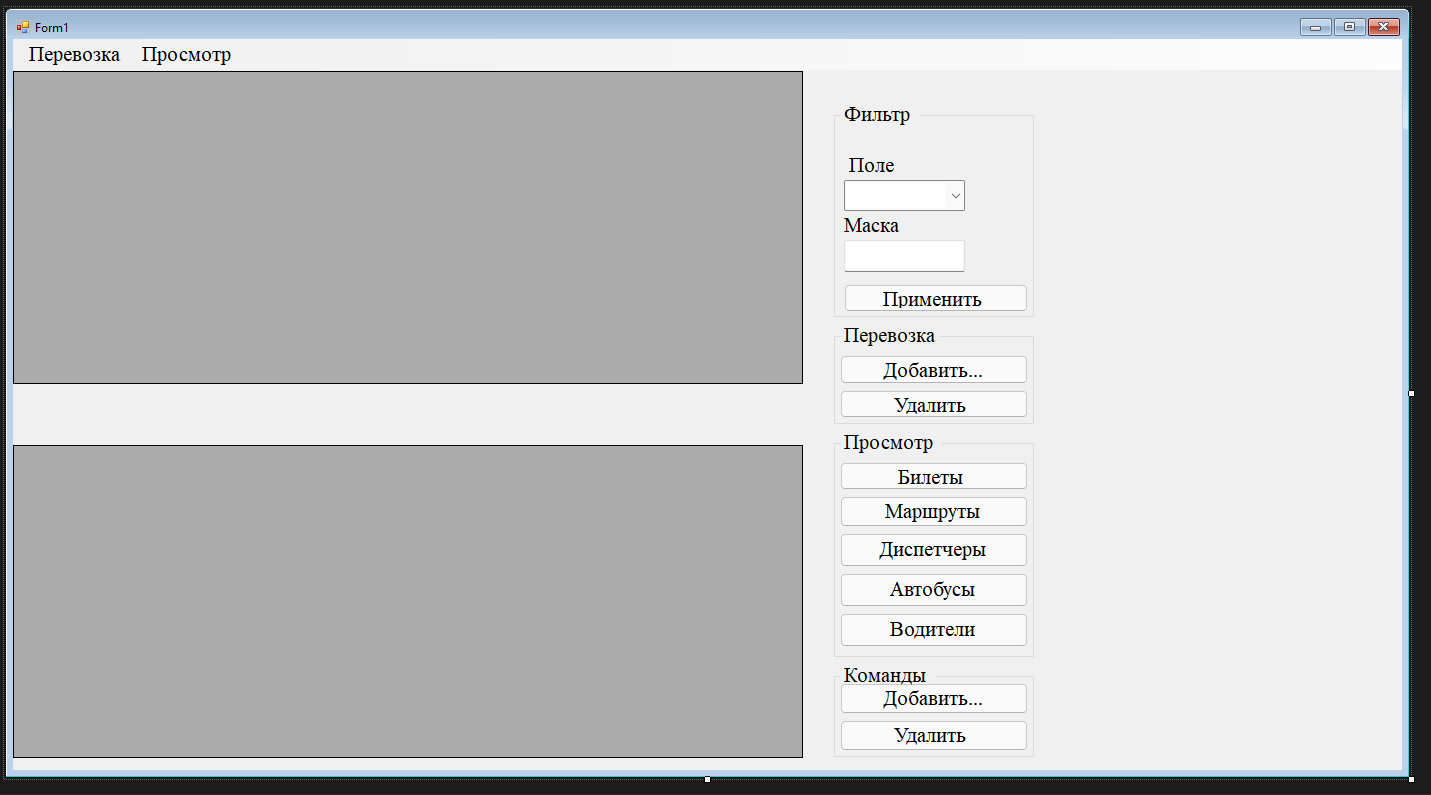


Рис. 23. Кнопка «Удалить» из основной формы «Form1.cs»

Создаем форму стандартным способом. Процесс создания формы в MS Visual Studio C# подробно описан [здесь](https://www.bestprog.net/ru/2016/01/09/006-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%b8%d1%8f-%d0%b8-%d0%b2%d1%8b%d0%b7%d0%be%d0%b2%d0%b0-%d0%bd%d0%be%d0%b2%d0%be%d0%b9-%d1%84%d0%be%d1%80%d0%bc%d1%8b/).

При [создании формы](https://www.bestprog.net/ru/2016/01/09/006-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%b8%d1%8f-%d0%b8-%d0%b2%d1%8b%d0%b7%d0%be%d0%b2%d0%b0-%d0%bd%d0%be%d0%b2%d0%be%d0%b9-%d1%84%d0%be%d1%80%d0%bc%d1%8b/) настраиваем такие свойства формы:

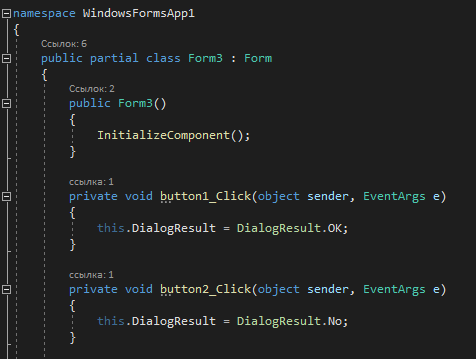
* свойство «Text» устанавливаем в значение «Удаление перевозки«;
* свойство FormBorderStyle = «Fixed3D»;
* свойство StartPosition = «CenterScreen»;
* свойство ControlBox = false.

Размещаем на форме следующие элементы управления:

* элемент управления типа «Label» (текст «Вы действительно желаете удалить данную запись?«);
* два элемента управления типа «Button» (кнопки «Да» и «Нет«).

В результате, в «Solution Explorer» получим файлы «Form3.cs» и «Form3.Designer.cs«.

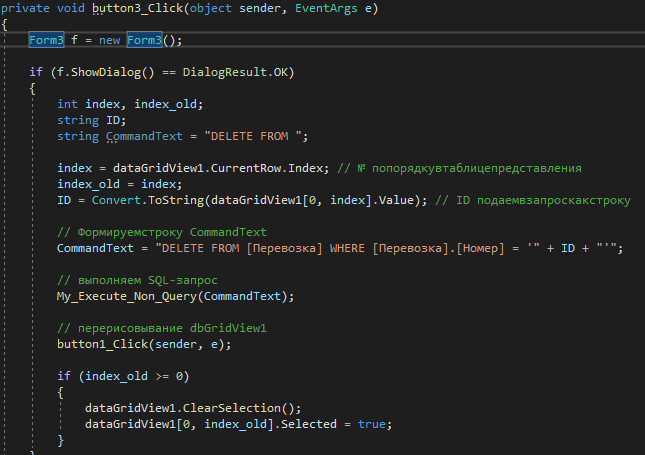
В форме «Form3.cs» программируем обработку события клика на кнопке «Да» стандартным способом:



**13. Программирование события вызова команды «Удалить» из группы «Перевозка»**

При вызове команды «Удалить» из группы «Перевозка» нужно вызвать форму «Form3.cs»

Листинг метода обработки события следующий:



**14. Программирование команд из группы «Просмотр»**

**14.1 Просмотр таблицы «Билет» в dataGridView2**

Чтобы вывести таблицу «Билет» в dataGridView2 пользователь должен сделать клик на кнопке «Билеты» (button4) из группы «Просмотр».

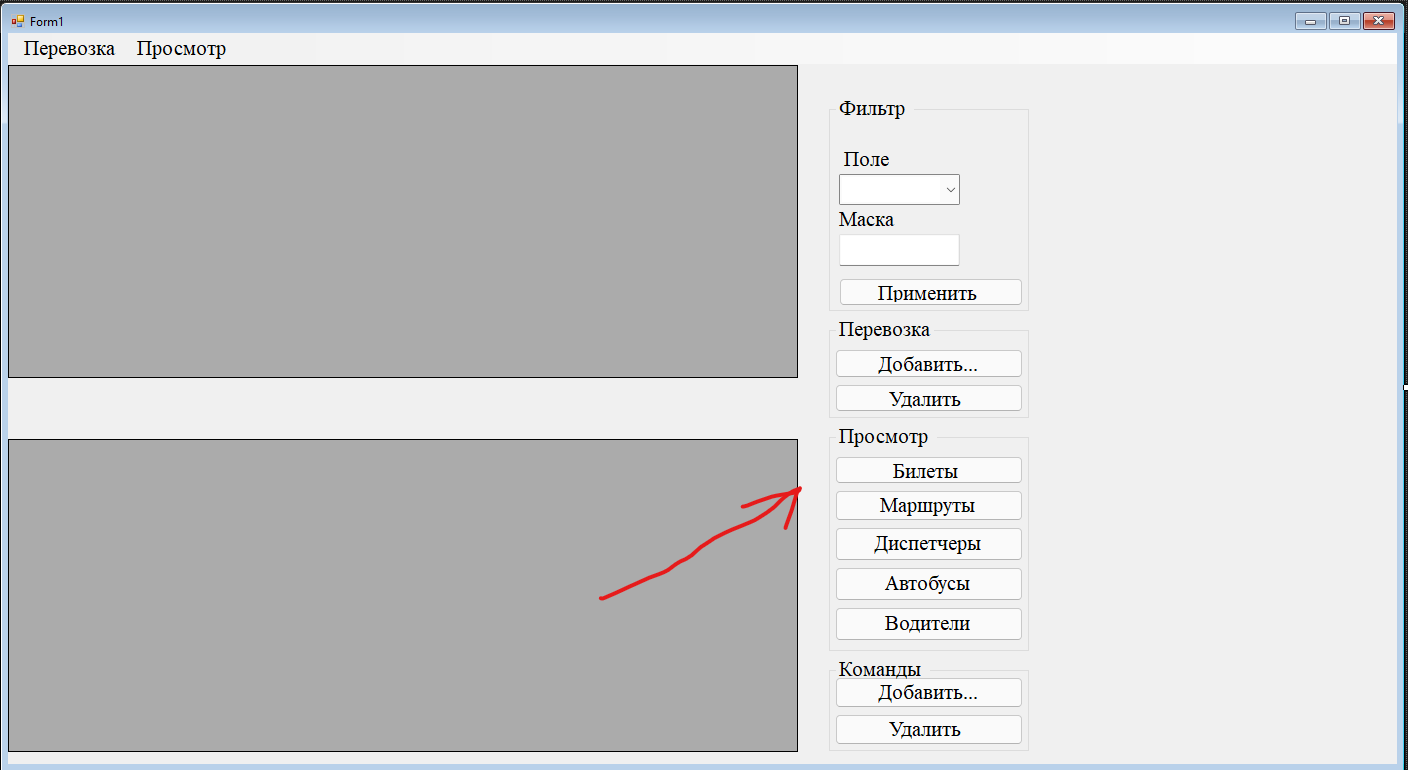
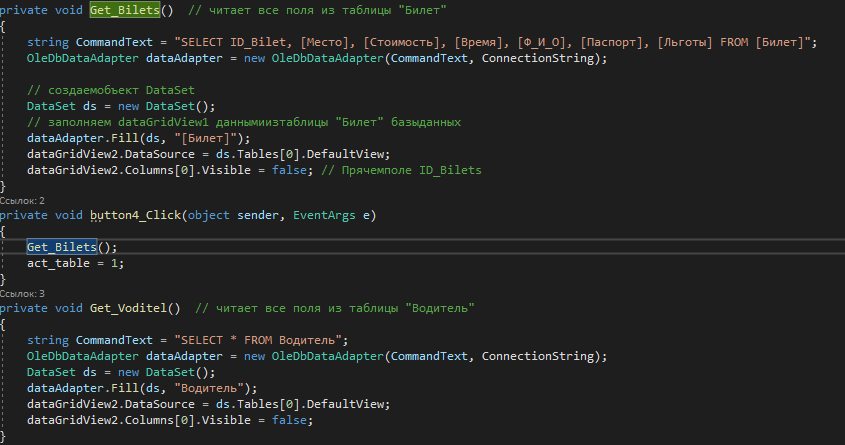
****

Рис. 24. Вызов команды просмотра списка купленных билетов

Листинг вспомогательной функции Get\_Bilets() и обработчика события button4\_Click() клика на кнопке button4 следующий.

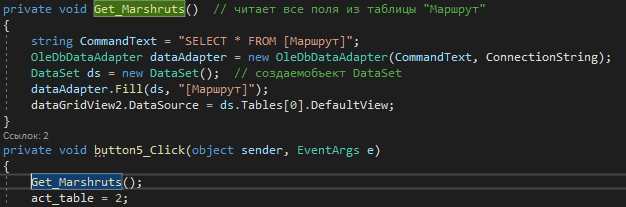
****

Теперь можно протестировать работу приложения.

**14.2. Просмотр таблицы «Маршрут» в dataGridView2**

Вывод таблицы «Маршрут» в dataGridView2 осуществляется с помощью вызова команды «Маршруты» из группы «Просмотр».

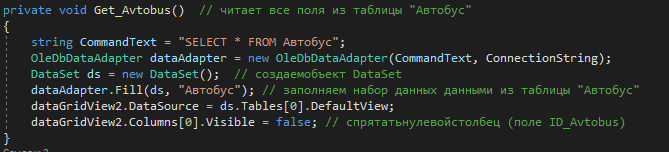
Листинг вспомогательного метода Get\_Marshruts() и обработчика события клика на кнопке «Маршруты» (button5) следующий:



**14.3. Просмотр таблицы «Автобус» вdataGridView2**

Чтобы вывести таблицу «Автобус» в dataGridView2 вызвать команду «Автобусы» (button6) из группы «Просмотр«.

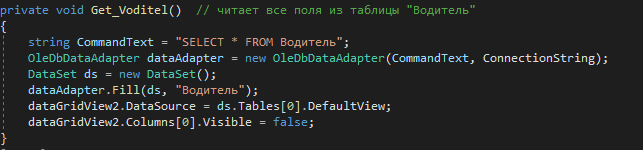
Листинг дополнительного метода Get\_Bus() и обработчика события клика на кнопке «Автобусы» следующий.



**14.4. Просмотр таблицы «Водитель» в dataGridView2**

Таблица «Водитель» выводится в dataGridView2 после нажатия на кнопке «Водители» из группы «Просмотр».

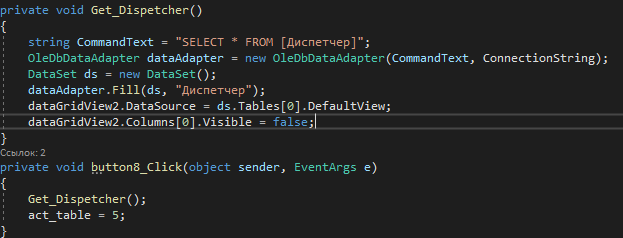
Листинг вспомогательного метода Get\_Voditel() и метода обработки события клика на кнопке button7 следующий:



**14.5. Просмотр таблицы «Диспетчер» в dataGridView2**

Для вывода таблицы «Диспетчер» в элементе управления dataGridView2 нужно выбрать команду «Диспетчеры» из группы «Просмотр«.

Листинг вспомогательного метода Get\_Dispetcher() и обработчика события клика на кнопке button8 следующий.

****

**15. Разработка форм добавления данных в таблицы**

**15.1. Разработка формы добавления нового билета**

Разработка новой формы в MS Visual Studio C# подробно описывается [здесь](https://www.bestprog.net/ru/2016/01/09/006-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%b8%d1%8f-%d0%b8-%d0%b2%d1%8b%d0%b7%d0%be%d0%b2%d0%b0-%d0%bd%d0%be%d0%b2%d0%be%d0%b9-%d1%84%d0%be%d1%80%d0%bc%d1%8b/).

Добавляем новую форму. Создаем форму добавления нового билета.

Файлы, которые соответствуют форме: «Form4.cs» и «Form4.Designer.cs«.

Настраиваем следующие свойства формы:

* свойство Text = «Билеты»;
* свойство FormBorderStyle = «Fixed3D»;
* свойство ControlBox = «false» (спрятать системное меню);
* свойство StartPosition = «CenterScreen» (окно появляется по центру экрана).

Размещаем элементы управления на форме. Значимые элементы управления имеют такие имена:

* textBox1 – поле «Место» в таблице «Билет«;
* textBox2 – поле «Стоимость«;
* textBox3 – «Время«;
* textBox4 – «Ф\_И\_О«;
* textBox5 – «Паспорт«;
* checkBox1 – «Льготы«;
* button1 – кнопка «Добавить«;
* button2 – кнопка «Отменить«.

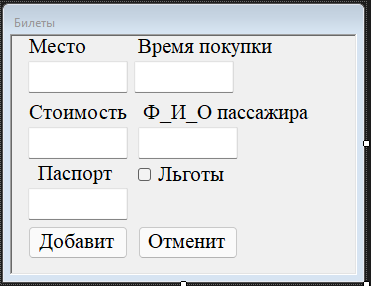
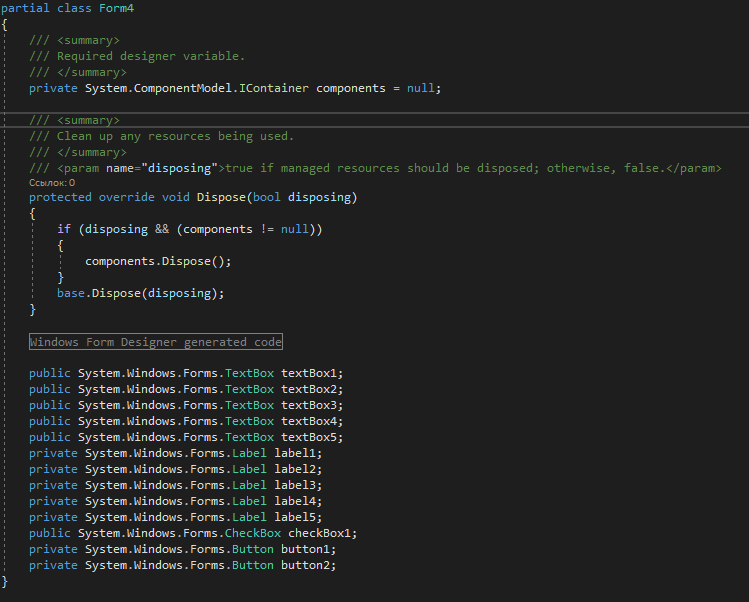


Рис. 25. Форма добавления нового билета

После создания формы нужно настроить видимость извне для некоторых полей класса Form4. Для этого переходим в файл «Form4.Designer.cs».

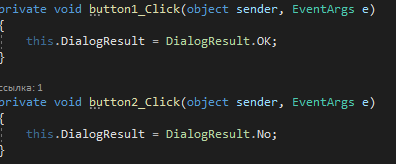
Устанавливаем доступ public для значимых полей формы «Form4.cs» – textBox1, textBox2, textBox3, textBox4, textBox5, checkBox1. Это можно сделать и другим путем изменив поле Modifiers в окне Properties.

Фрагмент соответствующего кода класса Form4 имеет вид:



Программируем событие клика на кнопках button1 и button2.

Листинг обработчиков событий, следующий:



**15.2. Разработка формы добавления данных в таблицу «Маршрут«**

По образцу создания формы добавления данных в таблицу «Билет» создаем форму добавления данных в таблицу «Маршрут». Построение новой формы подробно описывается [здесь](https://www.bestprog.net/ru/2016/01/09/006-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%b8%d1%8f-%d0%b8-%d0%b2%d1%8b%d0%b7%d0%be%d0%b2%d0%b0-%d0%bd%d0%be%d0%b2%d0%be%d0%b9-%d1%84%d0%be%d1%80%d0%bc%d1%8b/).

В результате, в Solution Explorer, получим файлы с именами «Form5.cs» и «Form5.Designer.cs».

Настраиваем следующие свойства формы:

* свойство Text = «Добавление маршрута»;
* свойство FormBorderStyle = «Fixed3D»;
* свойство ControlBox = «false»;
* свойство StartPosition = «CenterScreen».

Значимые элементы управления имеют такие имена:

* textBox1 – «№ маршрута«;
* textBox2 – «Пункт назначения«;
* textBox3 – «Район«;
* textBox4 – «Область«;
* textBox5 – «Расстояние«;
* textBox6 – «Вес«.
* dateTimePicker1 – время отправки;
* dateTimePicker2 – время прибытия;
* button1 – кнопка «Добавить«;
* button2 – кнопка «Отменить«.

В элементах управления dateTimePicker1 и dateTimePicker2 свойство Format устанавливаем у значение «Time».

Фрагмент соответствующего кода класса Form5 имеет вид:

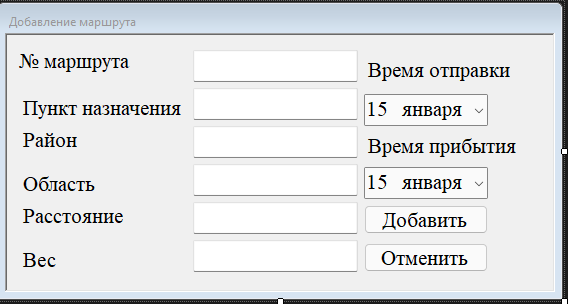
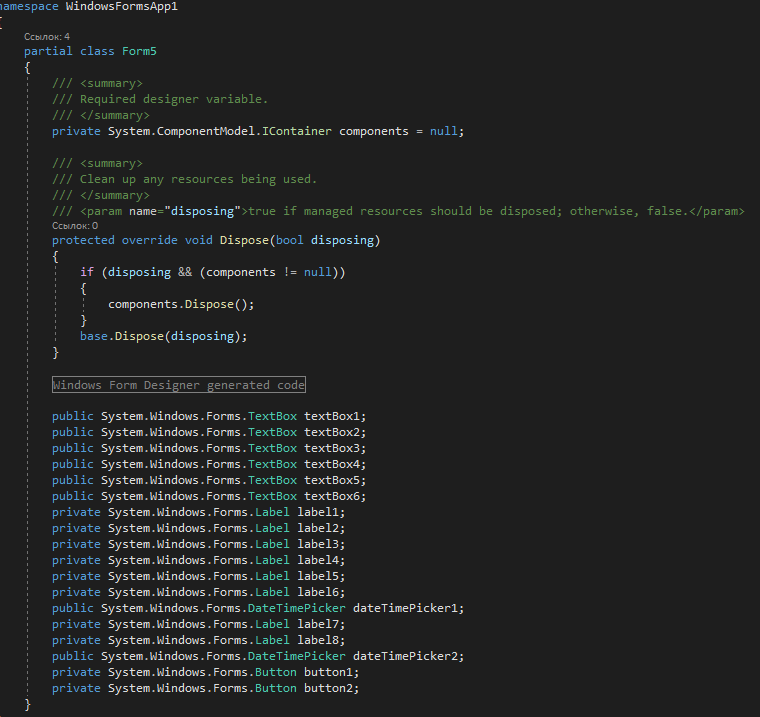


Рис. 26. Форма добавления нового маршрута

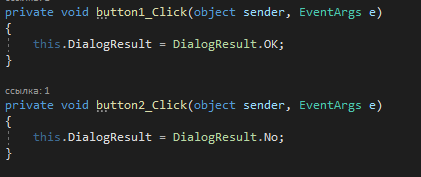
Устанавливаем доступ public для значимых полей формы «Form5.cs» – textBox1, textBox2, textBox3, textBox4, textBox5, textBox6, dateTimePicker1, dateTimePicker2.

Фрагмент соответствующего кода класса Form5 имеет вид:



Программируем события клика на кнопках button1 и button2.

Листинг обработчиков событий, следующий:



**15.3. Разработка формы добавления данных в таблицу «Автобус»**

Создание новой формы подробно описывается [здесь](https://www.bestprog.net/ru/2016/01/09/006-%d0%bf%d1%80%d0%b8%d0%bc%d0%b5%d1%80-%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%b8%d1%8f-%d0%b8-%d0%b2%d1%8b%d0%b7%d0%be%d0%b2%d0%b0-%d0%bd%d0%be%d0%b2%d0%be%d0%b9-%d1%84%d0%be%d1%80%d0%bc%d1%8b/).

Форма добавления нового транспортного средства изображена на рисунке 36. В приложении этой форме отвечают файлы «Form6.cs» и «Form6.Designer.cs»

Настраиваем следующие свойства формы:

* свойство Text = «Добавление автобуса«;
* свойство FormBorderStyle = «Fixed3D»;
* свойство ControlBox = «false»;
* свойство StartPosition = «CenterScreen».

Значимые элементы управления имеют такие имена:

* textBox1 (Номер);
* textBox2 (Модель);
* textBox3 (Номерной знак);
* textBox4 (Количество мест).

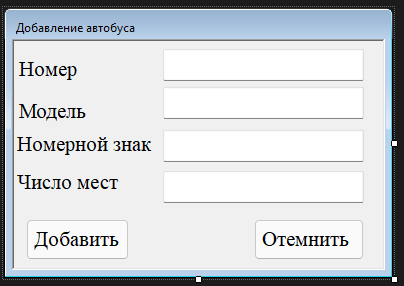
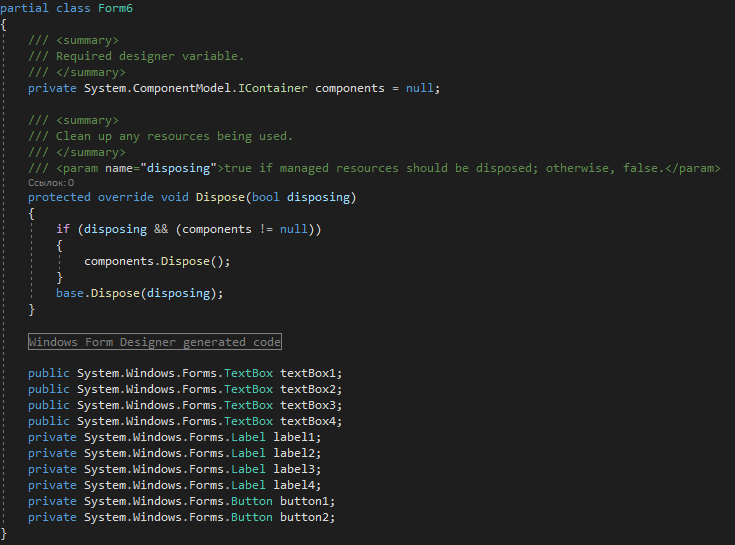
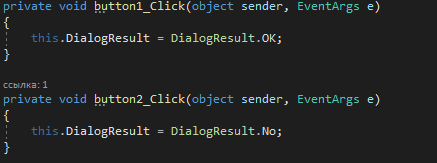


Рис. 27. Форма добавления нового автобуса

Устанавливаем доступ public для значимых полей формы «Form6.cs»: textBox1, textBox2, textBox3, textBox4.



Программируем событие клика на кнопках button1 и button2.



**15.4. Разработка формы добавления нового водителя**

Разработка формы добавления данных в таблицу «Водитель» осуществляется стандартным способом.

В Solution Explorer новосозданной форме соответствуют файлы «Form7.cs», «Form7.Designer.cs».

Свойства формы настраиваем так же, как и в предшествующих формах.

Значимые элементы управления имеют такие имена:

* textBox1 (Фамилия);
* textBox2 (Паспорт);
* dateTimePicker1 (Дата рождения);
* button1 – кнопка «Добавить«;
* button2 – кнопка «Отменить«.

В элементе управления dateTimePicker1 свойство Format = «Short».

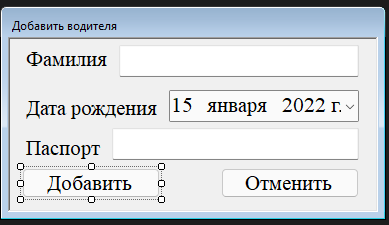


Рис. 28. Форма добавления данных в таблицу «Водитель«

В окне «Properties» устанавливаем доступ public в свойстве Modifiers для таких элементов управления:

* textBox1 (Фамилия);
* dateTimePicker1 (Дата рождения);
* textBox2 (Паспорт).

Автоматически будут внесены изменения в файле «Form7.Designer.cs»

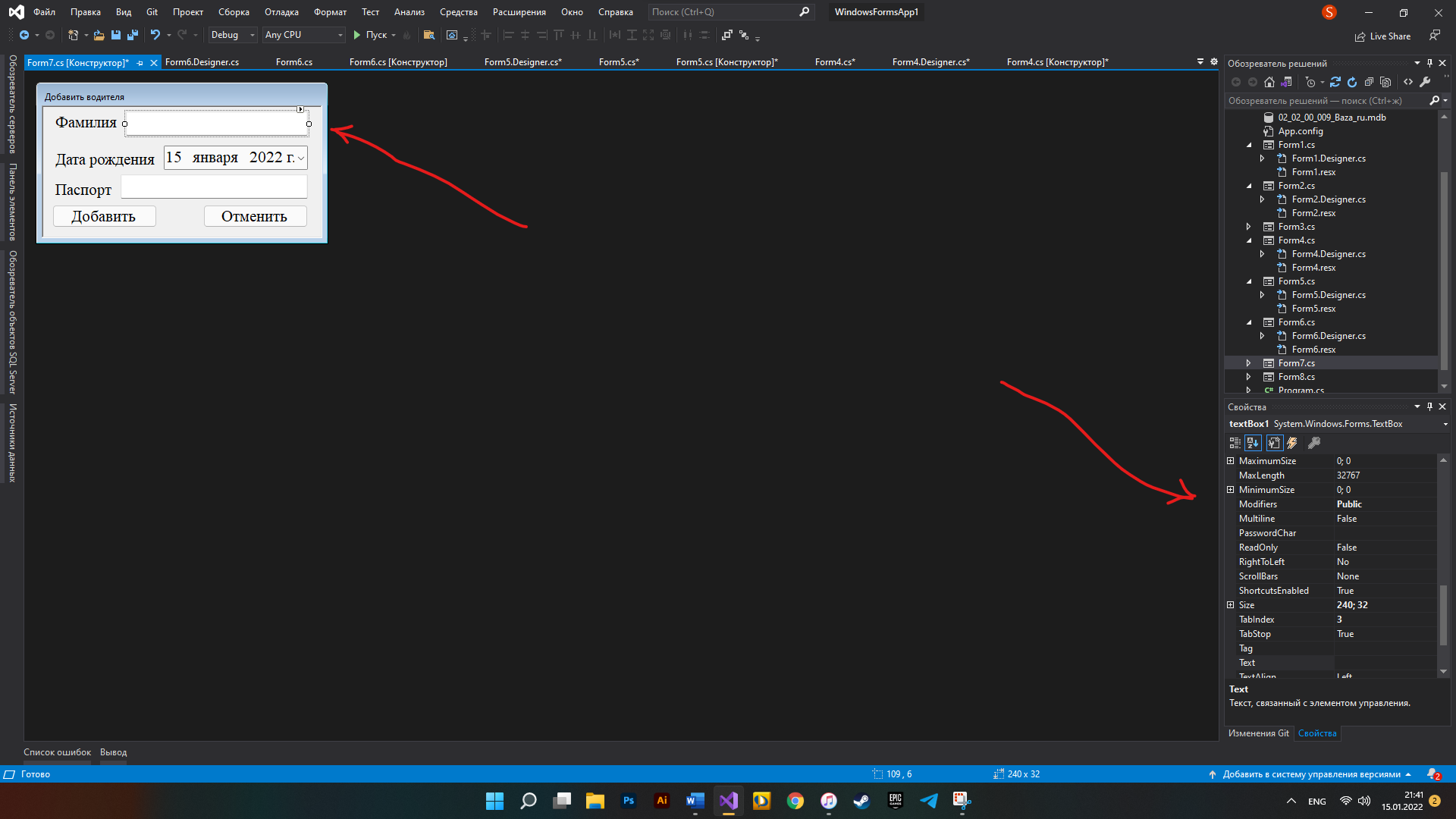
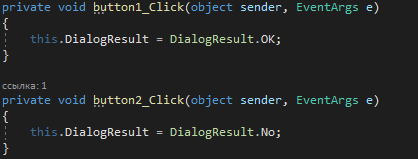


Рис. 29. Установление доступа public к элементу управления textBox1

Программируем событие клика на кнопках button1 и button2.

Листинг обработчиков событий, следующий:



**15.5. Разработка формы добавления нового диспетчера**

Форма добавления нового диспетчера. Файлы, которые соответствуют форме, имеют названия «Form8.cs» и «Form8.Designer.cs».

Свойства формы настраиваем так же как и в предшествующих формах.

Значимые элементы управления имеют такие имена:

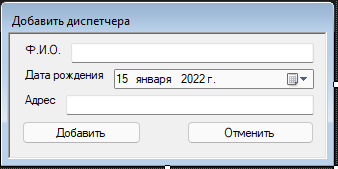


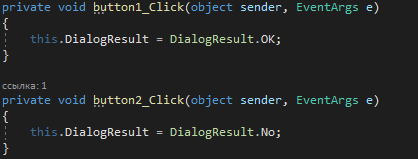
Рис. 30. Форма добавления данных в таблицу «Диспетчер»

В окне «Properties» устанавливаем доступ public в свойстве Modifiers для таких элементов управления:

* textBox1 (Ф\_И\_О.);
* dateTimePicker1 (Дата рождения);
* textBox2 (Адрес).

Автоматически будут внесены изменения в файл «Form8.Designer.cs».

Программируем события клика на кнопках button1 и button2 так же, как описано в предшествующих пунктах. Листинг обработчиков событий, следующий:



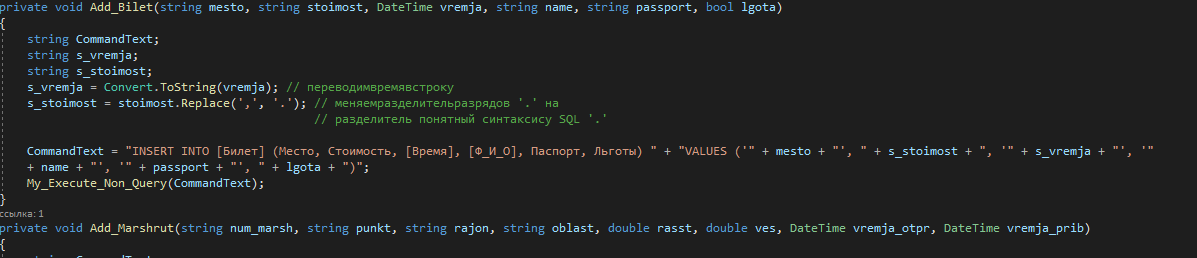
**16. Программирование методов добавления данных в таблицы «Билет», «Маршрут», «Автобус», «Водитель», «Диспетчер»**

Все следующие шаги будут выполнятся в главной форме приложения Form1.cs.

Поэтому, сначала активируем форму «Form1.cs»

**16.1. Добавление данных в таблицу «Билет»**

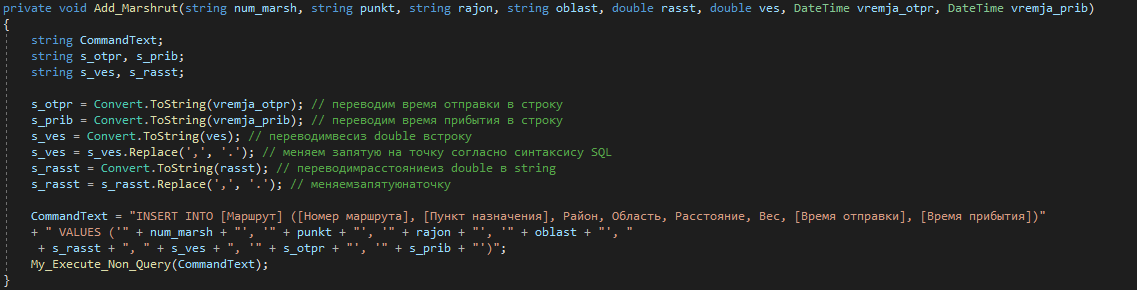
Листинг метода Add\_Bilet() добавления данных в таблицу «Билет» следующий:



В выше приведенном листинге метод Add\_Bilet() получает входным параметром переменные: mesto, stoimost, vremja, name, passport, lgota. Потом формируется запрос на языке SQL. В этом запросе, в переменной CommandText формируется команда INSERTINTO. Которая выполняется в методе My\_Execute\_Non\_Query().

**16.2. Добавление данных в таблицу «Маршрут»**

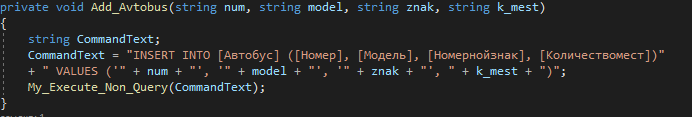
Листинг метода Add\_Marshrut() следующий.



Метод Add\_Marshrut() получает входными параметрами значения, которые отвечают полям таблицы «Маршрут» базы данных.

**16.3. Добавление данных в таблицу «Автобус»**

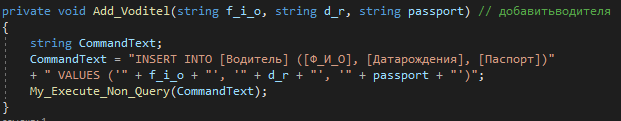
Листинг метода Add\_Avtobus() следующий.



Метод Add\_Avtobus() получает входные параметры, которые отвечают полям таблицы «Автобус» базы данных.

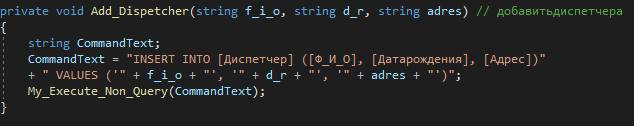
**16.4. Добавление данных в таблицу «Водитель»**

Листинг метода Add\_Voditel() добавления данных в таблицу «Водитель» следующий.



**16.5. Добавление данных в таблицу «Диспетчер»**

Листинг метода Add\_Dispetcher() следующий.



**17. Программирование события клика на кнопке «Добавить…» из группы «Команды»**

В основной форме «Form1.cs«, при выборе кнопки «Добавить…» из группы «Команды» нужно выводить одно из окон форм, которые соответствуют добавлению данных в таблицы «Билет», «Маршрут», «Автобус», «Водитель», «Диспетчер»

Для этого, средствами языка C# должна вызываться (создаваться) соответствующая форма. После внесения данных в форму и выбора подтверждающего запроса, данные из формы должны добавляться в соответствующую таблицу базы данных.

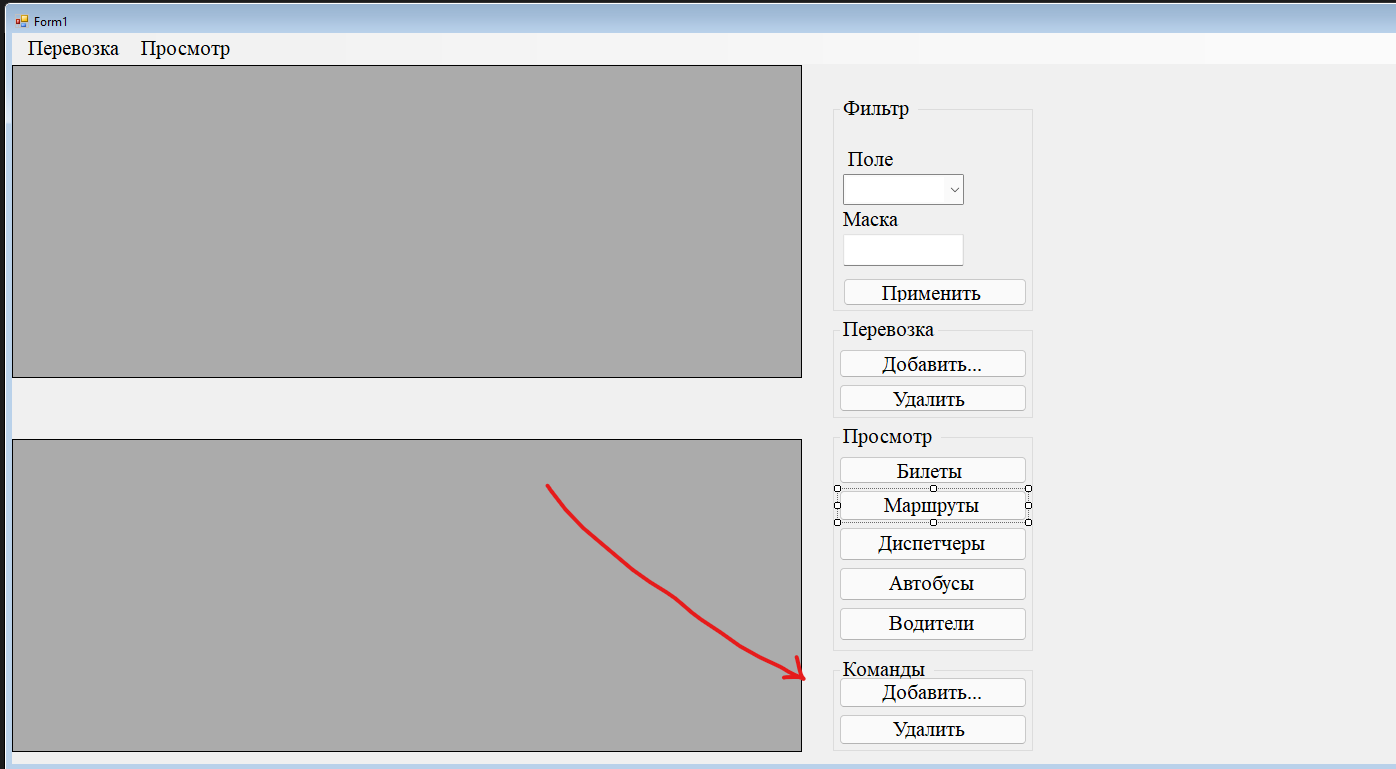
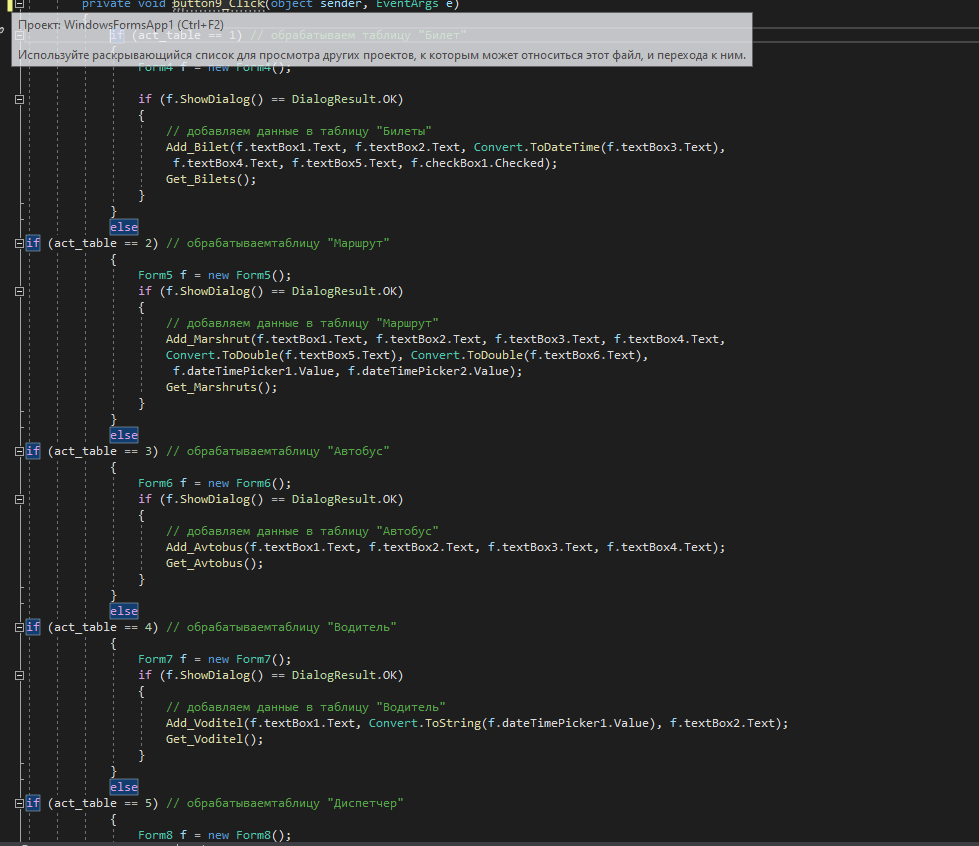


Рис. 31. Кнопка «Добавить…» главной формы

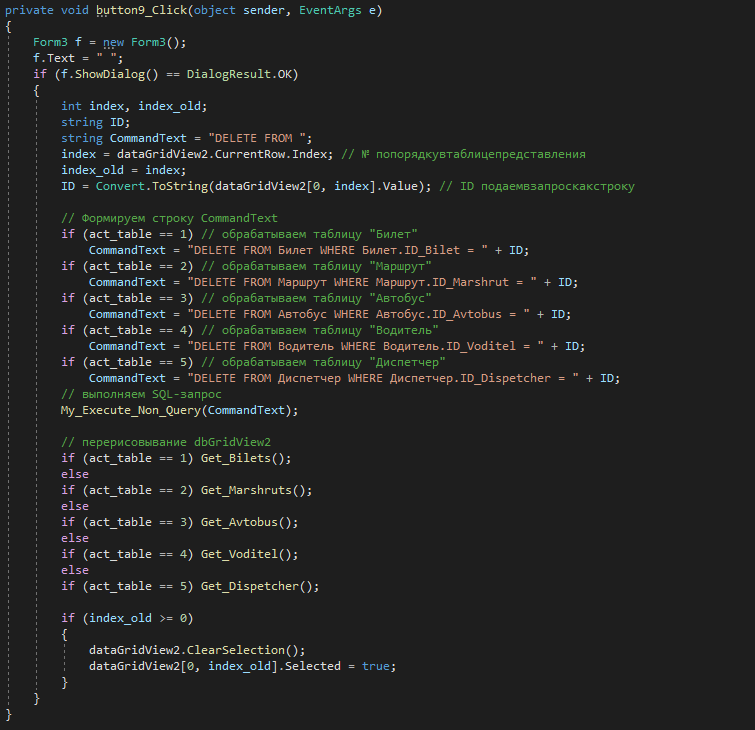
Листинг обработчика события клика на кнопке «Добавить…» следующий:



Как видно из листинга, на основе переменной act\_table (активная таблица) вызовется соответствующий метод добавления данных в базу данных.

**18. Программирование события клика на кнопке «Удалить» из группы «Команды»**

Листинг обработчика события клика на кнопке «Удалить» из группы «Команды» следующий.



**19. Программирование клика по командах меню**

Последним шагом осталось назначить командам меню выполнение соответствующих обработчиков событий клика на кнопках, которые размещаются на главной форме приложения.

Так, например, команда «Добавить…» из меню «Перевозка» должна выполнять ту же работу, что и команда «Добавить…» из группы «Перевозка» главной формы. Поэтому, нецелесообразно два раза писать код обработчика события добавления перевозки. Система Microsoft Visual Studio позволяет удобно назначать метод выполнения (если он уже запрограммирован) другой команде.

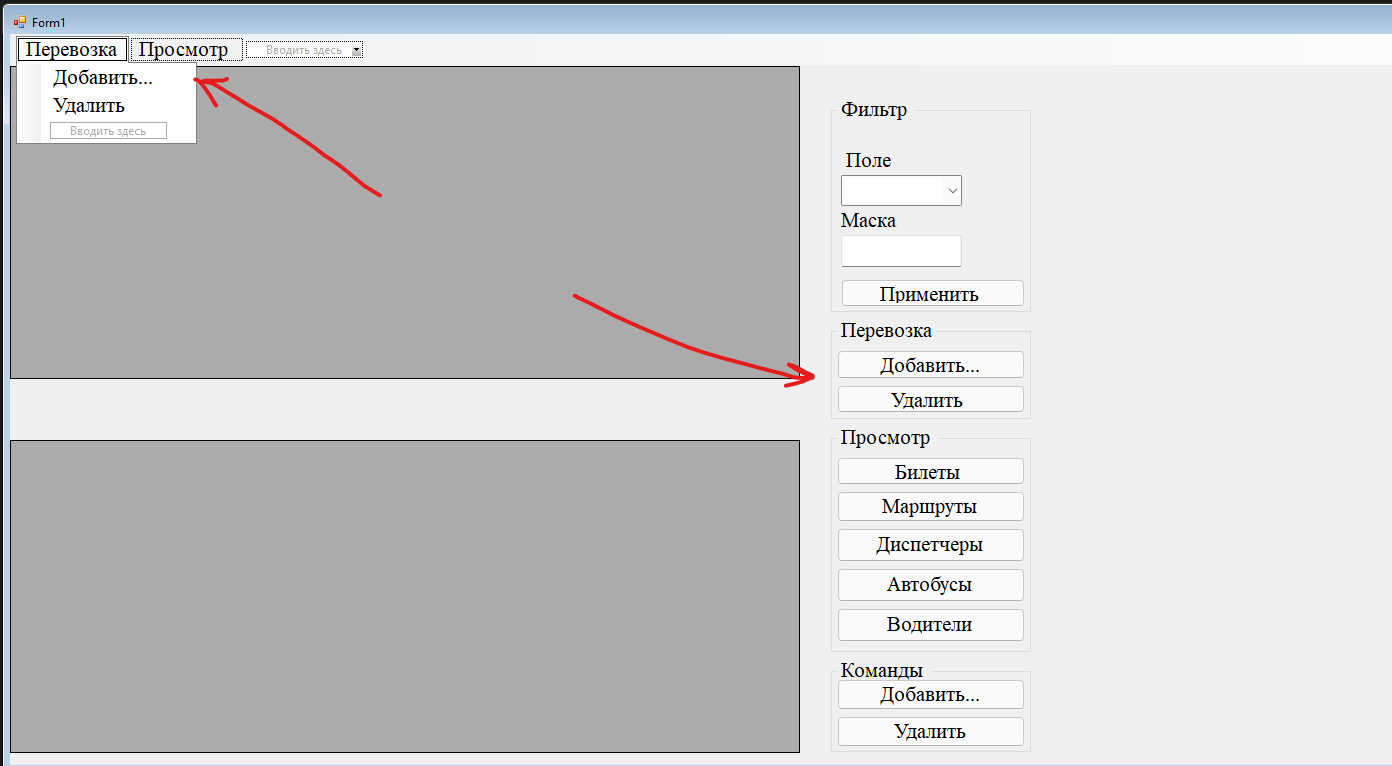
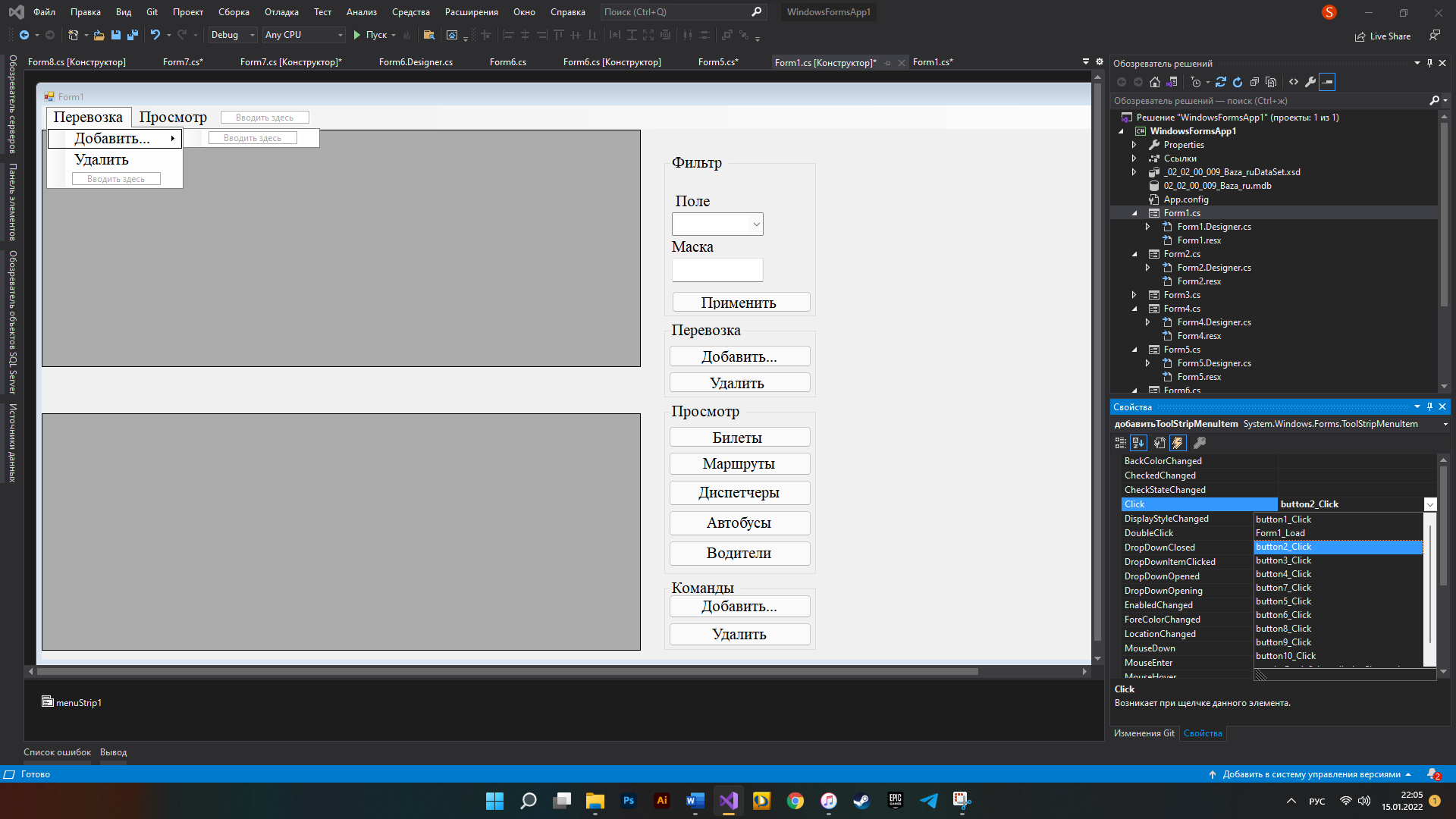


Рис. 32. Соответствие команды «Добавить…» из меню «Перевозка» одноименной команде из группы «Перевозка»

Команда добавления перевозки из основной формы вызовется на кнопке «Добавить…» из группы «Перевозка». Этой кнопке соответствует объект с именем button2. Метод обработки события клика на кнопке button2 в приложении имеет название button2\_Click() .

Для назначения метода обработки события команде «Добавить…» из меню «Перевозка» нужно:

* активировать (выделить мышкой) команду «Добавить…» из меню «Перевозка«;
* в окне «Properties» активировать вкладку «Events»;
* в поле «Click» выбрать метод button2\_Click (из выпадающего списка).

 Рис. 33. Назначение метода обработки события команде «Добавить…» из меню «Перевозка»

Точно также назначаем другим командам меню следующие обработчики событий:

* команде «Удалить» из меню «Перевозка» назначаем обработчик события button3\_Click;
* команде «Билеты» из меню «Просмотр» назначаем обработчик события button4\_Click;
* команде «Маршруты» из меню «Просмотр» назначаем обработчик события button5\_Click;
* команде «Автобусы» из меню «Просмотр» назначаем обработчик события button6\_Click;
* команде «Водители» из меню «Просмотр» назначаем обработчик события button7\_Click;
* команде «Диспетчеры» из меню «Просмотр» назначаем обработчик события button8\_Click.

**20. Запуск приложения на выполнение**

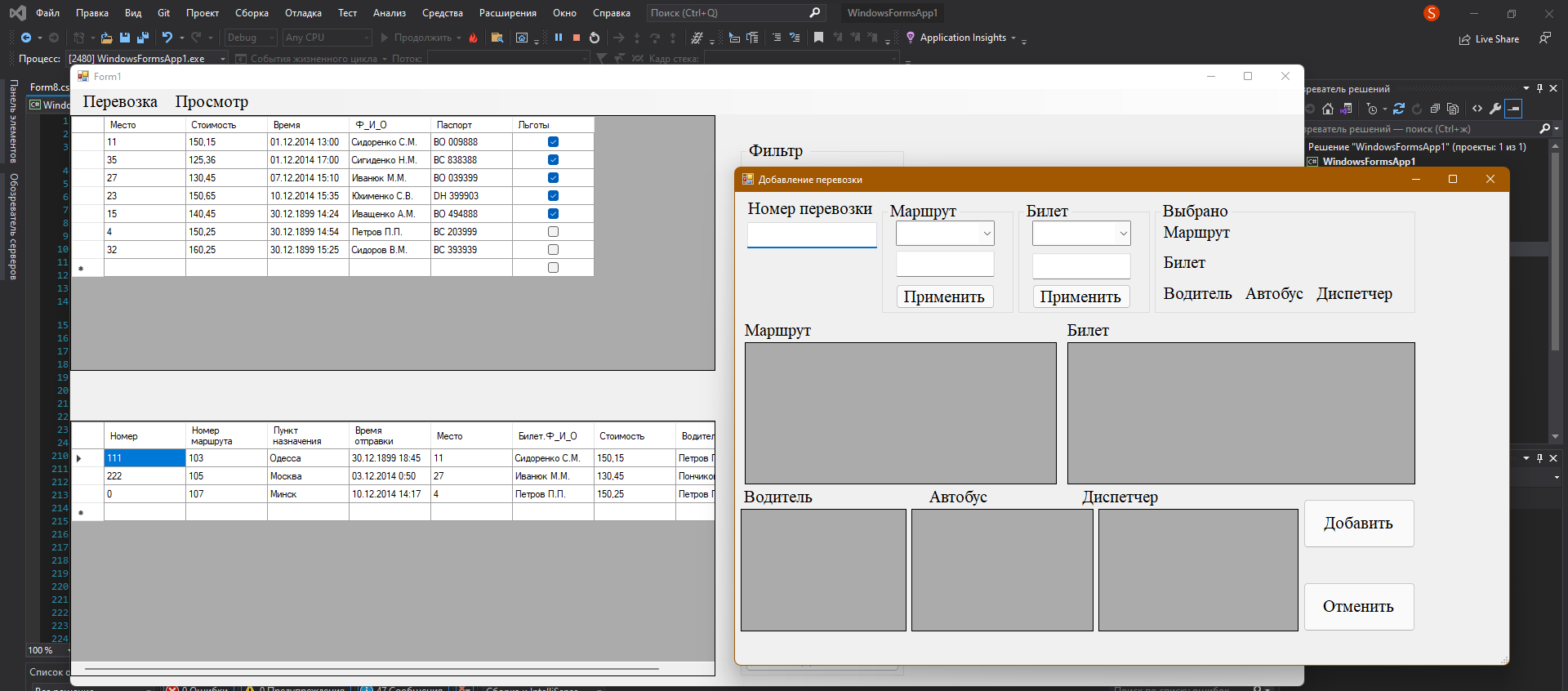


Рис. 34. (Скриншот работающей программы)

***Вывод:*** разработал приложение автоматизации работы диспетчера касс автовокзала, понял как устроена программа, повысил навык работы с формами, так же возникали некоторые трудности, но я так же нашёл для них решение.