

## АВТОНОМНАЯ МОДЕЛЬ ДОСТУПА ADO.NET И КЛАСС DATASET

Технология **ADO.NET** предоставляет в распоряжение разработчика класс **DataSet**, который спроектирован как автономный (отключенный) контейнер данных. В нем отсутствует соединение с базой данных. Фактически данные, хранящиеся в *DataSet*, даже не обязательно должны быть взяты из базы данных, с тем же успехом они могут представлять записи из файла CSV (Comma-Separated Values — значения, разделённые запятыми — файл текстового формата, предназначенный для представления табличных данных), файла XML, либо быть данными, полученными от измерительных устройств.

### Объекты DataSet и DataAdapter

Класс **DataAdapter** играет роль связующего звена между двумя компонентами ADO.NET: подключенным к источнику данных компонентом провайдеров данных .NET и неподключенным к источнику данных компонентом *DataSet*. Основным назначением объекта *DataAdapter* является управление процессом передачи данных от источника данных к объекту *DataSet* и от объекта *DataSet* к источнику данных.

Метод **Fill()** наполняет объект *DataSet* данными, полученными от источника данных, а метод **Update()** обновляет источник данных в соответствии с изменениями данных в таблицах объекта *DataSet*.

Для изучения основных принципов технологии программирования *DataSet* добавьте windows-форму и web-форму в ваши проекты и разместите в них элементы DataGridView и GridView соответственно. На рисунке 1 приведен код загрузки данных таблицы в DataGridView в windows-форме. На рисунке 2 демонстрируется win-форма в режиме пользователя.

```

16  SqlDataAdapter da;
17  DataSet ds = new DataSet();
18  SqlCommandBuilder bild;
19
20  private void Form3_Load(object sender, EventArgs e)
21  {
22      string str = ConfigurationManager.ConnectionStrings
23          ["Employees"].ConnectionString;
24      cnn = new SqlConnection(str);
25      da = new SqlDataAdapter("select * from Courses", cnn);
26      da.Fill(ds, "Courses");
27      dataGridView1.DataSource = ds.Tables["Courses"];
28      bild = new SqlCommandBuilder(da);
29  }
30  //При закрытии формы обновление записей базы данных
31  private void Form3_FormClosing(object sender, FormClosingEventArgs e)
32  {
33      da.Update(ds.Tables["Courses"]);
34  }

```

Рисунок 1 – Загрузка данных в объект DataSet в windows-форме

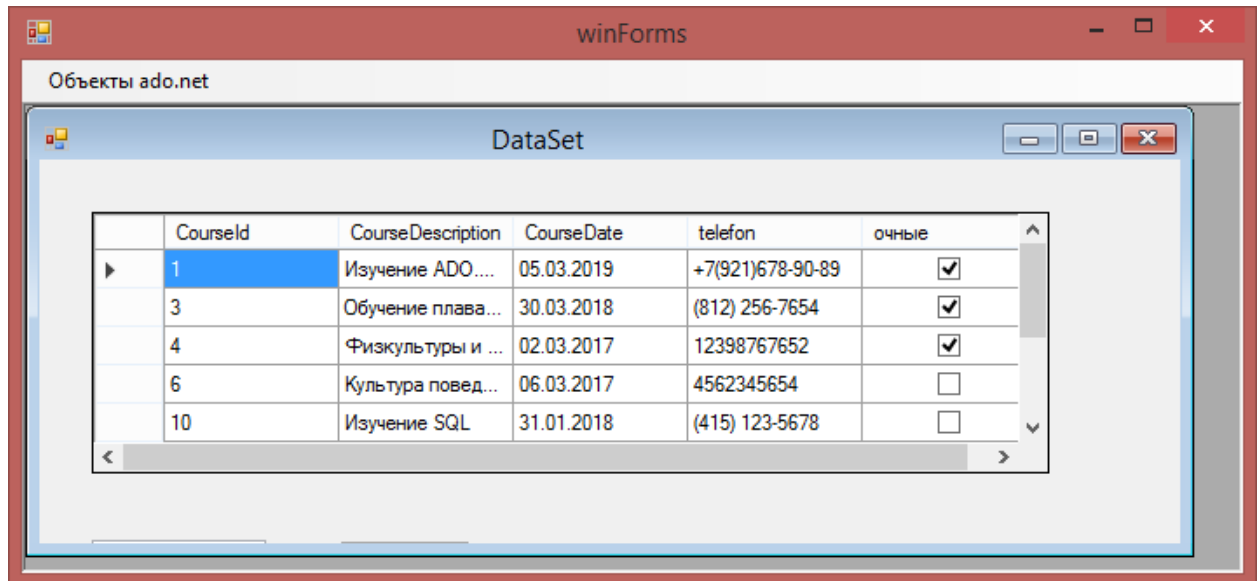


Рисунок 2 – Отображение DataSet в DataGridView windows-формы

Загрузить данные для локального хранения в объекте *DataSet* можно с помощью метода *Fill()* адаптера данных. Сохранить все изменения в БД можно с помощью метода *Update()* адаптера данных. Организовать отображение полученных данных удобно в элементе **DataGridView**, передать измененные данные в таблицу базы данных можно с помощью объектов **CommandBuilder** и *DataAdapter*.

Для web-формы программный код загрузки приведен на рисунке 3, а внешний вид приложения показан на рисунке 4.

```
DataSet ds = new DataSet();
SqlConnection cnn;
SqlDataAdapter da;
SqlCommandBuilder bild;

//Загрузка страницы
protected void Page_Load(object sender, EventArgs e)
{
    string str = WebConfigurationManager.ConnectionStrings
        ["Employees"].ConnectionString;
    cnn = new SqlConnection(str);
    da = new SqlDataAdapter("select * from Courses", cnn);
    da.Fill(ds, "Courses");
    //логическая связь серверного элемента управления с объектом данных
    GridView1.DataSource = ds.Tables["Courses"];
    //метод DataBind() проходит в цикле по DataSource, извлекает данные
    GridView1.DataBind();
    bild = new SqlCommandBuilder(da);
}
```

Рисунок 3 – Web-форма для работы с отсоединенными данными

CourseId	CourseDescription	CourseDate	telefon	очные
1	Изучение ADO.NET	05.03.2019 0:00:00	+7(921)678-90-89	<input checked="" type="checkbox"/>
3	Обучение плаванию	30.03.2018 0:00:00	(812) 256-7654	<input checked="" type="checkbox"/>
4	Физкультуры и спорта	02.03.2017 0:00:00	12398767652	<input checked="" type="checkbox"/>
6	Культура поведения	06.03.2017 0:00:00	4562345654	<input type="checkbox"/>
10	Изучение SQL	31.01.2018 0:00:00	(415) 123-5678	<input type="checkbox"/>
24	Изучение UML	19.02.2019 0:00:00	(123) 456-7123	<input checked="" type="checkbox"/>
25	Программирование на C#	06.02.2018 0:00:00	(123) 456-7890	<input checked="" type="checkbox"/>
26	Кулинария - искусство	19.03.2018 0:00:00	(234) 543-5678	<input type="checkbox"/>

Рисунок 4 - Внешний вид web-приложения для исследования DataSet

### Программирование вставки и удаления данных в объекте DataTable

Данные в объекте *DataSet* организованы в одной или нескольких объектах ***DataTable***. Каждая Запись в *DataTable* представлена объектом *DataRow*. Каждый объект *DataRow* представляет одиночную запись в таблице, которая была извлечена из источника данных. *DataRow* является контейнером для действительных значений полей.

Рассмотрим возможности объекта *DataTable* для работы со строками. Начнем с добавления и удаления строк в объекте. На рисунке 5 показана windows-форма с новыми элементами для ввода значений в поля таблицы, отображаемой в *DataGridView*.

На рисунке 6 показан программный код формирования значений и добавление новой строки в *DataTable*. Технологию записи новой строки данных в объект ***DataTable*** можно описать следующими шагами:

- Создается новый экземпляр объекта ***DataRow*** методом ***NewRow()***.
- Присваиваются конкретные значения полям новой записи.
- Запись включается в коллекцию ***Rows*** таблицы с помощью метода ***Add()***.

	CourseId	CourseDescription	CourseDate	telefon	очные
▶	1	Изучение ADO....	05.03.2019	+7(921)678-90-89	<input checked="" type="checkbox"/>
	3	Обучение плава...	30.03.2018	(812) 256-7654	<input checked="" type="checkbox"/>
	4	Физкультуры и ...	02.03.2017	12398767652	<input checked="" type="checkbox"/>
	6	Культура повед...	06.03.2017	4562345654	<input type="checkbox"/>
	10	Изучение SQL	31.01.2018	(415) 123-5678	<input type="checkbox"/>

2     

Наименование  
Новые технологии

24 февраля 2019 г.

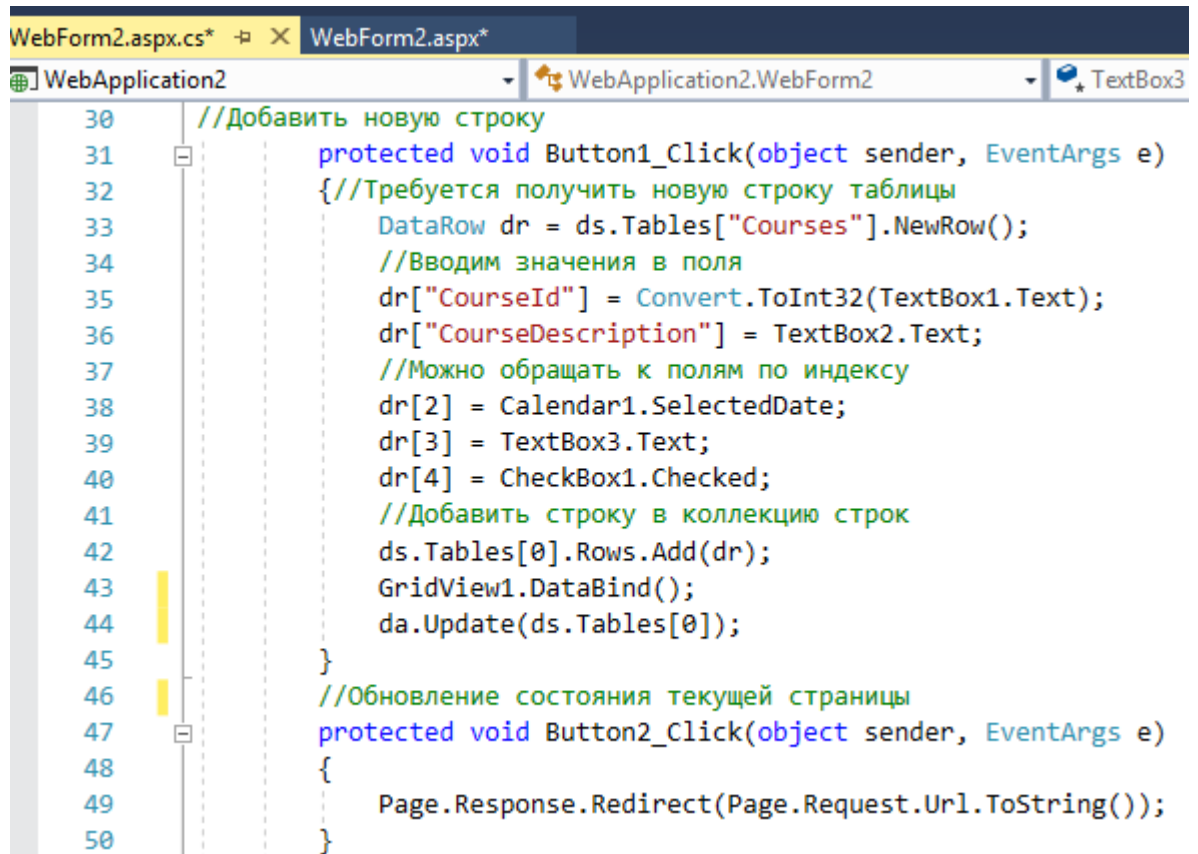
Телефон  
678-90-45

☒ Очно

Рисунок 5 – Изменение интерфейса для исследования объекта DataTable

```
//Добавить новую запись
private void button1_Click(object sender, EventArgs e)
{
    //Получить новую строку
    DataRow dr = ds.Tables["Courses"].NewRow();
    //Получить значения полей
    dr[0] = Convert.ToInt32(textBox1.Text);
    dr[1] = textBox2.Text;
    dr[2] = dateTimePicker1.Value.Date;
    dr[3] = textBox3.Text;
    dr[4] = checkBox1.Checked;
    //Добавить новую строку в коллекцию строк DataSet
    ds.Tables["Courses"].Rows.Add(dr);
}
```

Рисунок 6 – Добавление новой строки в DataTable в win-форме



```

30 //Добавить новую строку
31 protected void Button1_Click(object sender, EventArgs e)
32 {
33     //Требуется получить новую строку таблицы
34     DataRow dr = ds.Tables["Courses"].NewRow();
35     //Вводим значения в поля
36     dr["CourseId"] = Convert.ToInt32(TextBox1.Text);
37     dr["CourseDescription"] = TextBox2.Text;
38     //Можно обращаться к полям по индексу
39     dr[2] = Calendar1.SelectedDate;
40     dr[3] = TextBox3.Text;
41     dr[4] = CheckBox1.Checked;
42     //Добавить строку в коллекцию строк
43     ds.Tables[0].Rows.Add(dr);
44     GridView1.DataBind();
45     da.Update(ds.Tables[0]);
46 }
47 //Обновление состояния текущей страницы
48 protected void Button2_Click(object sender, EventArgs e)
49 {
50     Page.Response.Redirect(Page.Request.Url.ToString());
51 }

```

Рисунок 7 – Программирование веб-формы для добавления строки и обновления страницы

Строки можно не только вставлять, но и удалять целиком. Удаление записи реализовано методом **Delete()** объекта *DataRow*. Этот метод отмечает указанную строку на удаление, которое на самом деле произойдет только после вызова метода **AcceptChanges()**.

Метод **AcceptChanges()** фиксируют, а метод **RejectChanges()** наоборот, отменяет последние изменения в классах *DataTable*, *DataSet* и *DataRow*.

На рисунках 8 и 9 программный код и интерфейс для изучения действий методов **AcceptChanges()**, **RejectChanges()**.

```

//Пометка указанной строки на удаление
private void button2_Click(object sender, EventArgs e)
{
    //Вызов метода AcceptChange() для фиксирования всех изменений
    ds.Tables["Courses"].AcceptChanges();
    //Пометка на удаление строки с номером из textBox1
    ds.Tables["Courses"].Rows[Convert.ToInt32(textBox1.Text)].Delete();
}
//Снять пометку удаления с указанной строки
private void button3_Click(object sender, EventArgs e)
{
    ds.Tables[0].RejectChanges();
}

```

Рисунок 8 – Методы **AcceptChanges()** и **RejectChanges()**

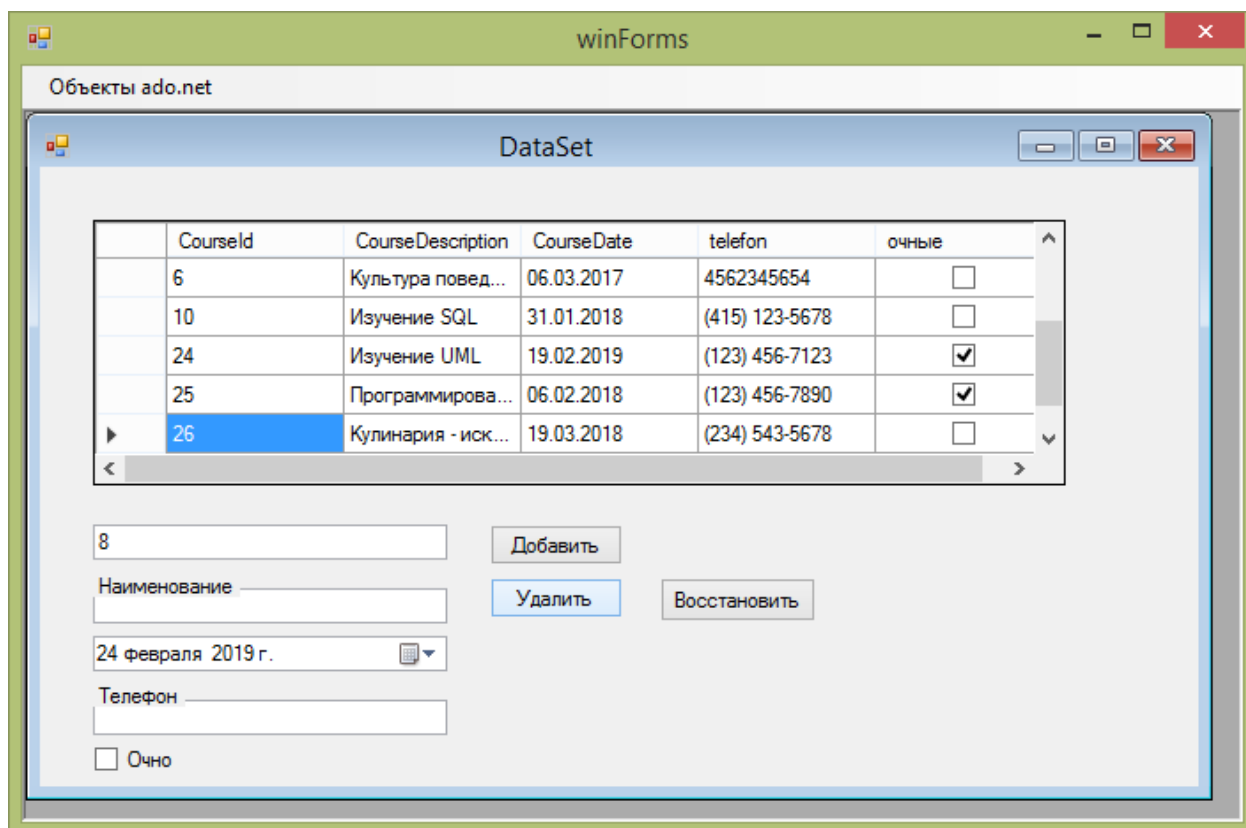


Рисунок 9 – Удаление и восстановление указанных строк в DataTable

### Задание для самостоятельной работы

Реализуйте все изученные методы в собственных приложениях. Отображение данных в элементах *DataGridView* и *GridView* надо сделать как в windows, так и в web-приложениях. В отчете опишите функциональные отличия этих элементов. Остальные методы: добавление новой строки с чтением данных из отдельных полей, пометку строки на удаление, отмену удаления, можно выполнить в одном из приложений.