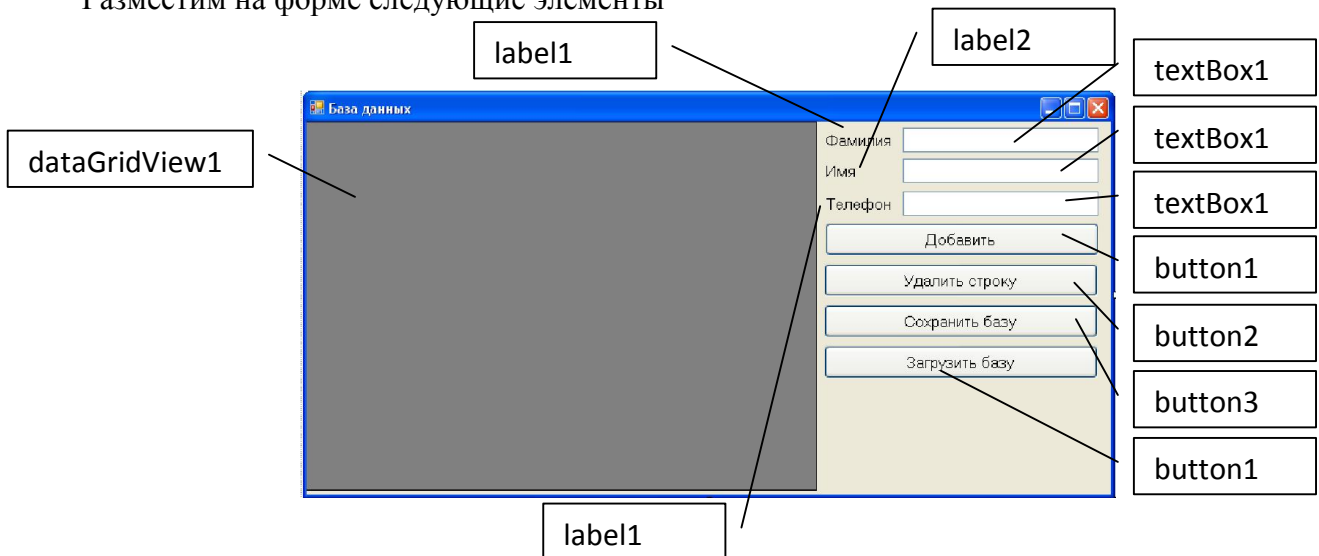


Создание простейшей базы данных при помощи dataGridView1 в C# и платформе .NET

Разместим на форме следующие элементы



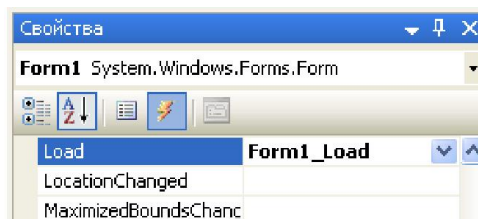
Так как мы будем сохранять данные на диске, подключим пространство имен, отвечающее за эти операции:

```
using System.Text;
using System.Windows.Forms;
using System.IO;
```

Опишем необходимые глобальные переменные:

```
public partial class Form1 : Form
{
    int n, m, i, j;
    public Form1()
```

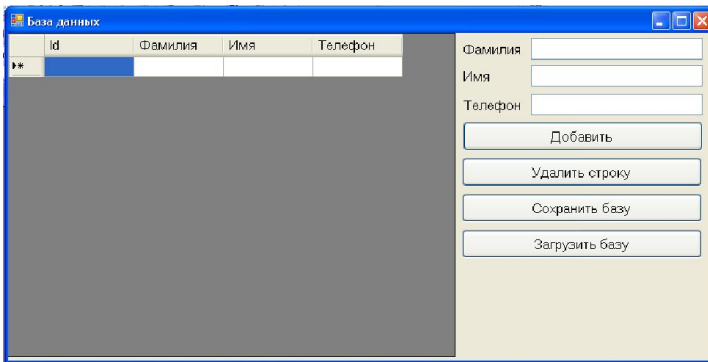
При старте программы у нас будет отображаться пустая таблица. Создадим событие, которое будет выполняться при загрузке формы. Выберем основную форму и перейдем в инспекторе объектов во вкладку события. Напротив слова Load дважды щелкнем мышкой и создадим это событие.



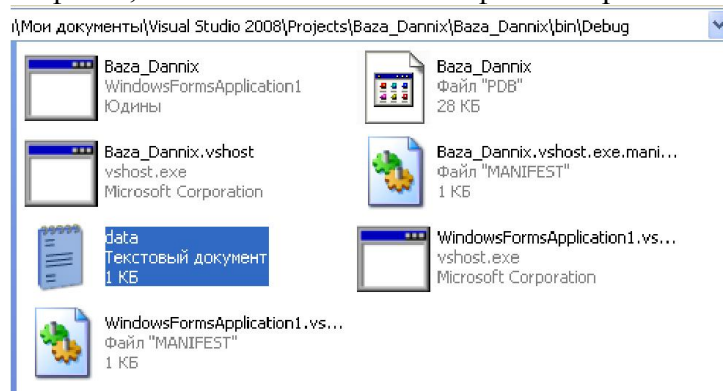
Запишем в него следующий код. В котором в пустую таблицу добавляются столбцы и подписываются их названия.

```
private void Form1_Load(object sender, EventArgs e)
{
    // загрузка формы
    // создаём и добавляем столбцы
    dataGridView1.Columns.Add("ID", "Id");
    dataGridView1.Columns.Add("Fa", "Фамилия");
    dataGridView1.Columns.Add("Name", "Имя");
    dataGridView1.Columns.Add("Tel", "Телефон");
}
```

При старте форма должна выглядеть следующим образом. Пользователь может добавлять



записи. Для этого он должен ввести данные в окна ввода и нажать кнопку «Добавить». Введенные строки можно будет удалять. Для этого нужно выделить строку целиком и нажать «Удалить строку». Базу можно будет сохранять и читать из файла. Правда сохранение и чтение происходит из строго определенного файла. Выбор файла тоже можно подключить, используя диалоговые окна. В нашем проекте данные будут храниться в текстовом файле, в папке с исполняемым файлом проекта.



Создадим событие «Добавить»

```
private void button1_Click(object sender, EventArgs e)
{
    //Добавляем строку с номером rowNumber
    int rowNumber = dataGridView1.Rows.Add();
    //Заполняем ячейки
    dataGridView1.Rows[rowNumber].Cells["ID"].Value = rowNumber;
    dataGridView1.Rows[rowNumber].Cells[1].Value = textBox1.Text;
    dataGridView1.Rows[rowNumber].Cells[2].Value = textBox2.Text;
    // можно по номеру столбца а можно по идентификатору
    dataGridView1.Rows[rowNumber].Cells["Tel"].Value=textBox3.Text;
    //Стираем поля ввода для новой информации
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
}

```

Создадим событие удалить строку. Хочу еще раз обратить внимание, что бы удалить строку, ее нужно выделить. Выделение всей строки в DataGridView происходит только при нажатии на самый первый (левый) столбец. Но, многие пользователи не знают о выше приведенном способе и часто выделяют строку, просто нажимая на любую ячейку.

```
private void button2_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        dataGridView1.Rows.Remove(row);
    }
}

```

Поэтому такой способ может не подойти, нам теперь нужно отловить не выделенную строку, а выделенную ячейку. Это мы можем сделать, например следующим образом:

```
foreach (DataGridViewCell cell in dataGridView1.SelectedCells)
{
    dataGridView1.Rows.RemoveAt (cell.RowIndex) ;
}
```

Запись в файл и StreamWriter

Для записи в текстовый файл используется класс StreamWriter. Свою функциональность он реализует через следующие методы:

- **Close**: закрывает записываемый файл и освобождает все ресурсы
- **Flush**: записывает в файл оставшиеся в буфере данные и очищает буфер.
- **Write**: записывает в файл данные простейших типов, как int, double, char, string и т.д.
- **WriteLine**: также записывает данные, только после записи добавляет в файл символ окончания строки

Создадим событие «Сохранить базу»

```
private void button3_Click(object sender, EventArgs e)
{
    n = dataGridView1.RowCount; // определяем число строк
    m = dataGridView1.ColumnCount; // определяем число столбцов
    //Создаём или перезаписываем существующий файл
    StreamWriter sw = File.CreateText("data.txt");
    //Записываем текст в поток файла
    sw.WriteLine(Convert.ToString(n));
    sw.WriteLine(Convert.ToString(m));
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            sw.WriteLine(Convert.ToString(dataGridView1[j,i].Value));
        }
    }
    //Закрываем файл
    sw.Close();
}
```

Чтение из файла и StreamReader

Класс StreamReader позволяет нам легко считывать весь текст или отдельные строки из текстового файла. Среди его методов можно выделить следующие:

- **Close**: закрывает считываемый файл и освобождает все ресурсы
- **Peek**: возвращает следующий доступный символ, если символов больше нет, то возвращает -1
- **Read**: считывает и возвращает следующий символ в численном представлении. Имеет перегруженную версию: Read(char[] array, int index, int count), где array - массив, куда считываются символы, index - индекс в массиве array, начиная с которого записываются считываемые символы, и count - максимальное количество считываемых символов
- **ReadLine**: считывает одну строку в файле
- **ReadToEnd**: считывает весь текст из файла

Создадим событие «Загрузить базу»

```
private void button4_Click(object sender, EventArgs e)
{
    //Создаем файловую переменную для чтения данных
    StreamReader f = new StreamReader("data.txt");
    //читаем первые две строки из файла
    n = Convert.ToInt32(f.ReadLine());
    m = Convert.ToInt32(f.ReadLine());
    //перебираем строки
    for (i = 0; i < n-1; i++)
    {
        int rowNumber = dataGridView1.Rows.Add();
        dataGridView1.Rows[rowNumber].Cells["ID"].Value=f.ReadLine();
        dataGridView1.Rows[rowNumber].Cells[1].Value = f.ReadLine();
        dataGridView1.Rows[rowNumber].Cells[2].Value = f.ReadLine();
        dataGridView1.Rows[rowNumber].Cells["Tel"].Value=f.ReadLine();
    }
    //Закрываем файл
    f.Close();
}
```

Программа в рабочем состоянии должна выглядеть так:

	Id	Фамилия	Имя	Телефон
	0	Иванов	Иван	1-22-33
	1	Петров	Петр	1-33-22
	2	Сидоров	Сидор	2-33-22
	3	Иванов	Петр	3-33-44
	4	Петров	Иван	4-33-44
	5	Сидоров	Иван	4-33-22
	6	Петров	Сидор	2-11-11
	7	Иванов	Сидор	4-33-21
	8	Сидоров	Петр	2-33-24
**				

Фамилия

Имя

Телефон

Добавить

Удалить строку

Сохранить базу

Загрузить базу