

insights/perevernutyi-klass-tekhnologiya-obucheniya-21-veka/ (дата обращения: 12.07.2017).

4. Поднебесова, Г. Б. Система профессиональной подготовки будущих учителей информатики / Г. Б. Поднебесова // Современная высшая школа: инновационный аспект. – 2012. – № 2. – С. 14-19.

5. Симонова, А. Л. U-Learning – повсеместное электронное обучение в XXI веке: на пути к коннективизму и СМАРТ-образованию / А. Л. Симонова, П. С. Ломаско // I Международная научная конференция в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и перспективы развития» «Информатизация образования и методика электронного обучения» : сборник материалов [Электронный ресурс]. – Красноярск: Сибирский федеральный ун-т, 2016. – Режим доступа: <http://conf.sfu-kras.ru/it-edu/participant/15011> (дата обращения 12.01.2018).

УДК 378.016:004.4

**Рудаков С. А.**

## **УПРАВЛЕНИЕ НАЧАЛОМ ПРОСМОТРА ОБЪЕКТОВ КЛАССА**

### **DATAGRIDVIEW НА ЯЗЫКАХ C++ И C#**

*Сергей Аркадьевич Рудаков*

*rudakov@csu.ru*

*ФГБОУ ВО «Челябинский государственный университет», Россия,*

*г. Челябинск,*

## **CONTROLLING THE BEGINNING OF VIEWING THE DATAGRIDVIEW OBJECTS IN C++ AND C# LANGUAGES**

*Sergey Arkadievich Rudakov*

*Chelyabinsk State University, Russia, Chelyabinsk*

***Аннотация.** В программу обучения студентов по направлениям, связанным с информатикой, входит курс «Объектно-ориентированное программирование». Изучение этого курса предполагает разработку оконных приложений с использованием шаблона *Windows Forms Application* и элемента управления *DataGridView*. Описаны некоторые приемы работы с *DataGridView*.*

***Abstract.** The IT-students' teaching program includes the "Object-oriented programming" training course. This course studying involves the development of window applications using the *Windows Forms Application* template and the *DataGridView* control element. Some approaches of working with *DataGridView* are described.*

***Ключевые слова:** *Windows Forms Application, DataGridView, C++, C#, класс, объект.**

***Keywords:** *Windows Forms Application, DataGridView, C++, C#, class, object.**

В программу курса «Объектно-ориентированное программирование» для студентов математического факультета ЧелГУ входит раздел разработки оконных приложений с использованием Microsoft Visual Studio версий 2013, 2015.

Microsoft Visual Studio предоставляет программистам удобный в использовании шаблон *Windows Forms Application (WFA)*, реализующий графический интерфейс пользователя и входящий в *Microsoft.NET Framework*. В состав проекта, использующего шаблон *WFA*, входят файлы управления проектом, файлы с графическим изображением формы и элементов управления, файлы с «неприкасаемым кодом», созданным Visual Studio, файлы, предназначенные для кода пользователя. Все эти файлы отображаются во вкладке «Обозреватель решения» (*Solution Explorer*) и могут быть открыты в окнах редакторов. Шаблон *WFA* выводит в окне конструктора Visual Studio форму в виде привычного окна *Windows*. При активном окне конструктора Visual

Studio предоставляет панель элементов управления разрабатываемого проекта. С помощью способа Drag-and-drop (тащи-и-бросай) элементы управления переносятся на форму или в окно конструктора. Практически все создаваемые проекты используют структурированную информацию, для работы с которой наиболее пригодны таблицы. Для создания таблиц среди элементов управления имеется элемент DataGridView, с помощью которого можно легко создать объект класса DataGridView.

Объект класса DataGridView отображается на форме в виде таблицы. Вид и содержание таблицы регулируются с помощью задания значений более 160 свойств и 400 методов класса DataGridView. Содержание таблицы может быть привязано средствами Visual Studio к базе данных (включая электронные таблицы) или формироваться функцией, разработанной программистом, из текстовых или других файлов.

При загрузке формы загружаем таблицу типа DataGridView. Просмотр таблицы в форме начинается с первой строки. Желательно увидеть таблицу с группой строк необходимой для просмотра (например, все строки, связанные со строкой, соответствующей текущей дате). В литературе (например, [1-4]) по языкам программирования C++ и C# такая информация отсутствует. На интернет-форумах вопросов о регулировании просмотра таблиц типа DataGridView много, но ответов мало и не все реализуемы на практике. Один из вариантов ответов предлагает ввести в программу следующий код, предусмотренный в Visual Studio, язык C#:

```
dataGridView1.ClearSelection(); //отмена текущего выделения
dataGridView1.Rows[j].Selected = true; //выделение строки
//Задание индекса первой отображаемой строки
dataGridView1.FirstDisplayedScrollingRowIndex = j;
```

При этом первая отображаемая строка с номером  $j$  будет также и выделенной.

**Пример 1.** Организация начала просмотра с заданной строки.

Создаем объект dataGridView1 с тремя полями: №пп, Дата, Содержание.

Заполняем таблицу 100000 записей. При выполнении программы получаем следующий вид таблицы

	№пп	Дата	Содержание	^
▶	0	16.01.1918 21:48:53	что-нибудь	
	1	20.01.1918 2:48:53	что-нибудь	
	2	23.01.1918 7:48:53	что-нибудь	
	3	26.01.1918 12:48:53	что-нибудь	
	4	29.01.1918 17:48:53	что-нибудь	
	5	01.02.1918 22:48:53	что-нибудь	
	6	05.02.1918 3:48:53	что-нибудь	
	7	08.02.1918 8:48:53	что-нибудь	▼

Рисунок 1. Вид таблицы класса dataGridView с 100000 строк

Поиск необходимой записи с использованием прокрутки – нудное и длительное занятие. Предлагаемый прием можно использовать при загрузке формы или при обработке какого-либо события (например, нажатия кнопки).

Код на языке C#:

```
j = 0;
do { d = Convert.ToDateTime(dataGridView1.Rows[j].Cells[1].Value);
    j++;
} //поиск строки
while (j < nr && d.Year < 2017 );
dataGridView1.Rows[j].Selected = true; //выделение строки
//j – начало просмотра
dataGridView1.FirstDisplayedScrollingRowIndex = j;
```

	№пп	Дата	Содержание	^
	11267	05.01.2017 4:52:00	что-нибудь	
	11268	08.01.2017 9:52:00	что-нибудь	
	11269	11.01.2017 14:52:00	что-нибудь	
	11270	14.01.2017 19:52:00	что-нибудь	
	11271	18.01.2017 0:52:00	что-нибудь	
	11272	21.01.2017 5:52:00	что-нибудь	
	11273	24.01.2017 10:52:00	что-нибудь	
	11274	27.01.2017 15:52:00	что-нибудь	∨

Рисунок 2. Вид таблицы с построенным началом просмотра

Рассмотренный прием предполагал сохранение порядка в таблице.

**Пример 2.** Организация начала просмотра с созданной группировкой.

В таблице расположены записи с такой же структурой как в примере 1, но поле содержания содержит 10-символьные слова, которые в совокупности содержат зашифрованное сообщение.

	№пп	Дата	Содержание	^
▶	0	29.01.1918 18:47:41	ПдèγН0ЗпÖ'	
	1	01.02.1918 23:47:41	.*Нl'Д7"	
	2	05.02.1918 4:47:41	jЗùТòКтèZll	
	3	08.02.1918 9:47:41	Fñ" 'èªèMè	
	4	11.02.1918 14:47:41	*RcÖзÛз	
	5	14.02.1918 19:47:41	}ΛэÇьIbÿ	
	6	18.02.1918 0:47:41	AjçÄЭF	
	7	21.02.1918 5:47:41	г"Υ#Dlrl	
	8	24.02.1918 10:47:41	è. 4-1l%+çQ.	∨

Рисунок 3. Вид таблицы с зашифрованным сообщением

Известно, что сообщение использует следующий алфавит:

'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'.

Для расшифровки сообщения необходимо начать просмотр таблицы со слов, содержащих эти символы алфавита.

Добавляем в таблицу поле Sort. Задаем функцию  $f: \{0, \dots, 99999\} \rightarrow N$

$f(k) = \begin{cases} \text{номер первого обнаруженного символа} & \text{если символ обнаружен} \\ \text{алфавита в строке поля «Содержание»,} & \end{cases}$

k+10,

если символ не обнаружен

В каждой строке с номером k полю Sort присваиваем значение f(k).

Упорядочиваем строки таблицы по возрастанию значения поля Sort.

	№пп	Дата	Содержание	Sort
	53816	21.10.2390 14:10:57	cшlljEú@μñ	0
	72957	10.12.2558 7:10:57	a <sup>l</sup> "r.obpγv	0
	7609	01.12.1984 3:10:57	f->fAāYŪĤ	0
	38431	30.08.2255 9:10:57	b9l8JqĤĚ	0
	73241	08.06.2561 11:10:57	aŪ>"EŞ'ĤčĚ	0
	81707	21.10.2635 5:10:57	cŌžqL?Nj	0
	81712	06.11.2635 6:10:57	g <sup>v</sup> *Λ?Ťč-	0
	81721	05.12.2635 3:10:57	hKllFč.đΩ	0

Рисунок 4. Вид таблицы с началом просмотра по возрастанию значений поля Sort

Код на языке C#:

```
j = 0; // алфавит
char[] mas = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
while (j < nr) // цикл поиска символов алфавита
{
    sns = dataGridView1.Rows[j].Cells[2].Value.ToString();
    ns = sns.IndexOfAny(mas);
    // вычисление функции f(j)
    if (ns >= 0) dataGridView1.Rows[j].Cells[3].Value = ns;
    else dataGridView1.Rows[j].Cells[3].Value = j + 10;
    j++;
}
// упорядочивание строк таблицы по возрастанию значений поля Sort
dataGridView1.Sort(dataGridView1.Columns[3], ListSortDirection.Ascending);
```

Обмен информацией с таблицей типа DataGridView можно производить в интерактивном режиме или с помощью записи в файл, чтения из файла. Рассмотрим эти способы.

Заносить информацию можно непосредственно в таблицу, если присвоить заранее свойству ReadOnly значение False. Такой метод не очень удобен в силу специфики вводимой информации (например, дата-время). Поэтому лучше подготовить необходимые поля на форме, которые отражают осо-

бенности вводимой информации. Рассмотрим программу «Календарь памятных дат», написанную на языке C++, которая при выполнении выглядит так (рисунок 5):



Рисунок 5. Окно выполнения программы «Календарь памятных дат»

Помимо текстовых полей здесь используются элементы управления классов `dateTimePicker` и `comboBox`.

Чтение текстовых полей производим с помощью кода  
`dGV1->Rows[nr]->Cells[0]->Value = tB1->Text; //Фамилия`

Чтение поля `comboBox` производим с помощью кода  
`dGV1->Rows[nr]->Cells[3]->Value = cB1->SelectedItem-> ToString();`  
`//вид события`

Чтение поля `dateTimePicker` производим в два шага: считываем значение с помощью кода в переменную типа `String`, а затем из полученной строки с учетом формата выбираем подстроку с необходимой информацией. Пишем код

```
d1 = dTP1->Value.Date.ToString();
d = d1->Substring(0, 10) // Выбираем только дату
```

При необходимости корректировки строки из таблицы производим обработку события **Click** при установленном фокусе на редактируемой строке. Чтение в текстовое поле производим с помощью кода

```
tB1->Text = dGV1->Rows[nr]->Cells[0]->Value->ToString();//Фамилия
```

Чтение в поле comboBox производим с помощью кода

```
cB1->Text = dGV1->Rows[nr]->Cells[3]->Value->ToString();//Вид события
```

Чтение в поле datePicker производим с помощью кода

```
dTP1->Value = Convert.ToDateTime(dGV1->Rows[nr]->Cells[4]->Value);//Дата
```

В рассматриваемой программе для хранения информации используем текстовый файл. Текстовый файл хранит информацию, структура которой отображена в заголовках таблицы на рисунке 5. Удобство хранения информации в текстовом файле состоит в том, что этот файл может редактироваться с помощью простых редакторов. В качестве разделителей полей информации в файле могут быть использованы символы псевдографики «||», «¶» и другие редко используемые в текстах символы. Признаком окончания строки обычный «\n». Код чтения файла в таблицу состоит из следующих этапов: чтение файла в строку; подсчет числа строк для формирования таблицы; цикл выбора подстроки, соответствующей строке таблицы; цикл выбора из полученной строки подстрок, соответствующих полям таблицы и присвоение значений этих подстрок полям таблицы. Чтение таблицы в файл происходит в обратном порядке. В цикле формируем строку из значений полей строки таблицы, расставляя разделители полей (например, «||») и разделители строк (например, «¶»). Из этих строк формируем строку, которую копируем в текстовый файл.

Описанные приемы работы с DataGridView апробированы на лабораторных занятиях со студентами 2-го курса математического факультета ЧелГУ в 2017 году.



### *Список литературы*

1. Албахари, Дж. Албахари, Б. С# 5.0. Справочник. Полное описание языка.: Пер. с англ. — Москва: ООО И.Д. Вильямс, 2014. — 1008 с.: ил.
2. Стиллмен Э., Грин, Дж. Изучаем С#. 2-е изд. — Санкт-Петербург: Питер, 2012. — 696 с.: ил.
3. Троелсен. Э. С# и платформа .NET. Библиотека программиста. — Санкт-Петербург: Питер, 2004. — 796 с.: ил.
4. Шилдт, Г. С# 4.0: полное руководство. : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2011. — 1056 с.: ил.

УДК [378.016:802/809]:[378.147.33:004]

**Сергиевская И. Л., Анисимова И. В.**

**ОБУЧЕНИЕ ПОСТРОЕНИЮ МОНОЛОГИЧЕСКОГО  
ВЫСКАЗЫВАНИЯ НА ИНОСТРАННОМ ЯЗЫКЕ С ПОМОЩЬЮ  
ИНТЕРАКТИВНЫХ ТЕХНОЛОГИЙ (НА БАЗЕ ВОЕННОГО ВУЗА)**

*Ирина Леонидовна Сергиевская*

*Кандидат педагогических наук, доцент*

*Ирина Владимировна Анисимова*

*alexiz-17@yandex.ru*

*Филиал Военной академии материально-технического обеспечения*

*им. Генерала армии А.В. Хрулева (г. Пенза)*

**LEARNING TO BUILD A MONOLOGIC UTTERANCE IN A FOREIGN  
LANGUAGE USING INTERACTIVE TECHNOLOGIES (ON THE BASIS  
OF THE MILITARY UNIVERSITY)**

*Irina Leonidovna Sergievskaya*

*Irina Vladimirovna Anisimova*

*The branch of the Military Academy of logistics*