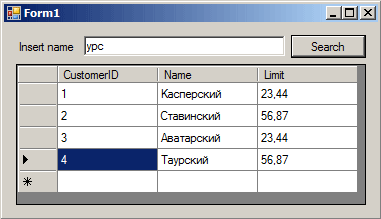
**c# - Поиск ячеек/строк по DataGridView**

Достаточно часто необходимо произвести поиск по значению определенной ячейки в DataGridView. Обычно это дело сводится к поиску по связанному объекту DataTable. Но сегодня мы поговорим именно о поиске в самом контроле DataGridView.

Имеется следующая форма:



На ней размещены myDataGridView с привязкой к myDataTable, myTextBox и myButton.

Текстовое поле принимает искомое значение для столбца Name, кнопка выполняет поиск и выделяет первую найденную строку, которая соответствует критерию поиска.

Обработчик события следующий:

[view plaincopy to clipboardprint?](http://skillcoding.com/Default.aspx?id=151)

1. **private** **void** myButton\_Click(**object** sender, EventArgs e)
2. {
3. **string** selectString =
4. "Name Like '%" + myTextBox.Text.Trim() + "%'";
6. DataRowCollection allRows =
7. ((DataTable)myDataGridView.DataSource).Rows;
9. DataRow[] searchedRows =
10. ((DataTable)myDataGridView.DataSource).
11. Select(selectString);
13. **int** rowIndex = allRows.IndexOf(searchedRows[0]);
15. myDataGridView.CurrentCell =
16. myDataGridView[0,rowIndex];
17. }

Сначала определяем фильтр selectString. Затем получаем все строки из источника данных. Потом, с помощью метода Select, среди них находим те, что удовлетворяют условию. Затем, в полученном масиве выбираем первый элемент-строку и получаем для неё индекс, под которым данная строка размещена в myDataTable. Последняя операция назначает текущую ячейку в myDataGridView используя полученный индекс.

Почти вся работа велась на контроле DataTable. Вы скажете, а как же поиск по самому DataGridView? Да, поиск выполнялся по источнику привязки, но результат этой работы отображается в виде динамического назначения текущей ячейки в DataGridView.

Если все таки необходимо «бегать» по гриду, тогда вот приемлемый вариант:

[view plaincopy to clipboardprint?](http://skillcoding.com/Default.aspx?id=151)

1. **private** **void** myButton\_Click(**object** sender, EventArgs e)
2. {
3. **for** (**int** i=0; i < myDataGridView.RowCount; i++)
4. **if** (myDataGridView[1, i].FormattedValue.ToString().
5. Contains(myTextBox.Text.Trim()))
6. {
7. myDataGridView.CurrentCell = myDataGridView[0, i];
8. **return**;
9. }
10. }

Но в этом случае также будет выводится только первая найденная строка.

Что бы дать возможность перемещаться по найденным строкам, введите переменную уровня класса, к примеру i, а в цикле выполните маленькие изменения:

[view plaincopy to clipboardprint?](http://skillcoding.com/Default.aspx?id=151)

1. **int** i;
3. **private** **void** myButton\_Click(**object** sender, EventArgs e)
4. {
5. **for** (; i < myDataGridView.RowCount;)
6. **if** (myDataGridView [1, i].FormattedValue.ToString().
7. Contains(myTextBox.Text.Trim()))
8. {
9. myDataGridView.CurrentCell = myDataGridView[0, i];
10. **if** (i < myDataGridView.RowCount-1)
11. i++;
12. **else**
13. i = 0;
14. **return**;
15. }
17. }

Обращаю ваше внимание, что если для элемента DataGridView установлена возможность добавлять строки, то в условии if (i < myDataGridView.RowCount-1) следует единичку заменить на двойку, что б не учитывать последнюю пустую строку вставки новых данных: if (i < myDataGridView.RowCount-2)