Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Коллекции в с#**»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

Изменить приложение, описанное в порядке выполнения практического задания из задания 2 в соответствии с вариантом задания. Приложение должно обеспечивать добавление элементов в коллекцию, удаление элементов из коллекции и отображение элементов коллекции, также добавить дополнительный метод. Варианты заданий, с которых необходимо выбрать объект предметной области и коллекцию, приведены в таблице 1.2. Для объекта предметной области необходимо придумать не менее 5-ти разнотипных полей, содержащих информацию. Интерфейс не должен быть переполнен элементами управления, но в тоже время должен предоставлять наибольшую функциональность.

**Теоретические сведения**

При выполнение этой работы в своей программе я использовал несколько основных элементов таких как класс DataGridView, класс SortedList, а также класс Device, который был создан мной.

**Класс DataGridView.**

Пространство имен: [System.Windows.Forms](https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.forms?view=windowsdesktop-8.0)

Сборка: System.Windows.Forms.dll

Данный класс отображает дату в настраиваемой сетке, в которой я создал шесть колонок: ID, Name, Limits, Units, Scale, Type.Заполнение сетки было выполнено при помощи взятие информации из объекта класса SortedList.

Так же к этому объекту был прикреплен объект windowsforms contexMenuStip с помощью которого были сделаны функции Изменение данных, их удаление, а также поиск.

**Класс SortedList**

Пространство имен:[System.Collections](https://learn.microsoft.com/ru-ru/dotnet/api/system.collections?view=net-7.0)

Сборка: System.Collections.NonGeneric.dll

Предоставляет коллекцию пар "ключ-значение", упорядоченных по ключам.

Доступ к парам можно получить по ключу и индексу.

В коллекции, которую я использовал в виде ключа использовалась переменную id созданная в самом классе Form1: Form, а для значения я брал объект класса Device.

**Класс Device**

У класса есть шесть основных полей name, limits, units, scale, type, divisions.

Все поля этого класса имеют тип доступа private.

В классе также есть конструктор, в котором заполняются поля, методы для вывода информации из полей и метод для расчёта поля scale.

У двух методов есть перегрузка, которая применяется в том случае, когда пользователь захочет изменить информацию в DataGridView.

**Блок-схема**

**Листинг программы**

Form1: Form

private DataGridViewColumn dataGridViewColumn1 = null;

private DataGridViewColumn dataGridViewColumn2 = null;

private DataGridViewColumn dataGridViewColumn3 = null;

private DataGridViewColumn dataGridViewColumn4 = null;

private DataGridViewColumn dataGridViewColumn5 = null;

private DataGridViewColumn dataGridViewColumn6 = null;

SortedList<int,Device> deivices = new SortedList<int,Device>();

int id = 1;

public Form1()

{

InitializeComponent();

initDataGridView();

}

private void initDataGridView()

{

dataGridView1.DataSource = null;

dataGridView1.Columns.Add(getdatecol1());

dataGridView1.Columns.Add(getdatecol2());

dataGridView1.Columns.Add(getdatecol3());

dataGridView1.Columns.Add(getdatecol4());

dataGridView1.Columns.Add(getdatecol5());

dataGridView1.Columns.Add(getdatecol6());

dataGridView1.AutoResizeColumns();

}

private DataGridViewColumn getdatecol1()

{

if (dataGridViewColumn1 == null)

{

dataGridViewColumn1 = new DataGridViewTextBoxColumn();

dataGridViewColumn1.Name = "";

dataGridViewColumn1.HeaderText = "ID";

dataGridViewColumn1.ValueType = typeof(string);

dataGridViewColumn1.Width = dataGridView1.Width / 12;

}

return dataGridViewColumn1;

}

private DataGridViewColumn getdatecol2()

{

if (dataGridViewColumn2 == null)

{

dataGridViewColumn2 = new DataGridViewTextBoxColumn();

dataGridViewColumn2.Name = "";

dataGridViewColumn2.HeaderText = "name";

dataGridViewColumn2.ValueType = typeof(string);

dataGridViewColumn2.Width = dataGridView1.Width / 6;

}

return dataGridViewColumn2;

}

private DataGridViewColumn getdatecol3()

{

if (dataGridViewColumn3 == null)

{

dataGridViewColumn3 = new DataGridViewTextBoxColumn();

dataGridViewColumn3.Name = "";

dataGridViewColumn3.HeaderText = "limits";

dataGridViewColumn3.ValueType = typeof(int);

dataGridViewColumn3.Width = dataGridView1.Width / 6;

}

return dataGridViewColumn3;

}

private DataGridViewColumn getdatecol4()

{

if (dataGridViewColumn4 == null)

{

dataGridViewColumn4 = new DataGridViewTextBoxColumn();

dataGridViewColumn4 .Name = "";

dataGridViewColumn4.HeaderText = "units";

dataGridViewColumn4.ValueType = typeof(string);

dataGridViewColumn4.Width = dataGridView1.Width / 6;

}

return dataGridViewColumn4;

}

private DataGridViewColumn getdatecol5()

{

if (dataGridViewColumn5 == null)

{

dataGridViewColumn5 = new DataGridViewTextBoxColumn();

dataGridViewColumn5.Name = "";

dataGridViewColumn5.HeaderText = "scale";

dataGridViewColumn5.ValueType = typeof(double);

dataGridViewColumn5.Width = dataGridView1.Width / 6;

}

return dataGridViewColumn5;

}

private DataGridViewColumn getdatecol6()

{

if (dataGridViewColumn6 == null)

{

dataGridViewColumn6 = new DataGridViewTextBoxColumn();

dataGridViewColumn6.Name = "";

dataGridViewColumn6.HeaderText = "type";

dataGridViewColumn6.ValueType = typeof(string);

dataGridViewColumn6.Width = dataGridView1.Width / 5;

}

return dataGridViewColumn6;

}

private void addDevice(string name,int limits, string units,string type,int divisions)

{

Device dev = new Device(name, limits, units, type,divisions);

deivices.Add(id, dev);

id++;

ShowListGrid();

}

private void DeletStudent(int elementindex)

{

deivices.RemoveAt(elementindex);

ShowListGrid();

}

private void ShowListGrid()

{

dataGridView1.Rows.Clear();

foreach (KeyValuePair<int, Device> s in deivices)

{

DataGridViewRow row = new DataGridViewRow();

DataGridViewTextBoxCell cell1 = new DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell2 = new DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell3 = new DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell4 = new DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell5 = new DataGridViewTextBoxCell();

DataGridViewTextBoxCell cell6 = new DataGridViewTextBoxCell();

Device d = s.Value;

cell1.ValueType = typeof(int);

cell1.Value = s.Key;

cell2.ValueType = typeof(string);

cell2.Value = d.Get\_name();

cell3.ValueType = typeof(int);

cell3.Value = d.Get\_limits();

cell4.ValueType = typeof(string);

cell4.Value = d.Get\_units();

cell5.ValueType = typeof(string);

cell5.Value=d.Get\_scale();

cell6.ValueType = typeof(string);

cell6.Value=d.Get\_type();

row.Cells.Add(cell1);

row.Cells.Add(cell2);

row.Cells.Add(cell3);

row.Cells.Add(cell4);

row.Cells.Add(cell5);

row.Cells.Add(cell6);

dataGridView1.Rows.Add(row);

}

}

private void button1\_Click(object sender, EventArgs e)

{

bool pr = true;

if (button1.Text == "Добавить")

{

if (textBox1.Text != "" && textBox2.Text != "" && textBox3.Text != "")

{

for (int i = 0; i < textBox1.Text.Length; i++)

{

if (char.IsDigit(textBox1.Text[i])) pr = false;

}

for (int i = 0; i < textBox2.Text.Length; i++)

{

if (char.IsDigit(textBox2.Text[i])) pr = false;

}

for (int i = 0; i < textBox3.Text.Length; i++)

{

if (char.IsDigit(textBox3.Text[i])) pr = false;

}

if (pr ==true)

{

addDevice(textBox1.Text, Convert.ToInt32(numericUpDown1.Value), textBox2.Text, textBox3.Text, Convert.ToInt32(numericUpDown2.Value));

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

}

}

else

{

if (textBox4.Text != "")

{

PerformSearch(textBox4.Text);

}

else

{

MessageBox.Show("Заполните поля");

}

textBox1.Visible = true;

textBox2.Visible = true;

textBox3.Visible = true;

numericUpDown1.Visible = true;

numericUpDown2.Visible = true;

label1.Visible = true;

label2.Visible = true;

label3.Visible = true;

label4.Visible = true;

label5.Visible = true;

textBox4.Visible = false;

button1.Text = "Добавить";

textBox4.Text = "";

}

}

private void deleteToolStripMenuItem\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Delete", " ", MessageBoxButtons.YesNo);

if (result == DialogResult.Yes)

{

int i = dataGridView1.CurrentRow.Index;

DeletStudent(i);

}

}

private void изменитьToolStripMenuItem\_Click(object sender, EventArgs e)

{

int i = dataGridView1.CurrentRow.Index;

dataGridView1.Rows[i].Cells[1].Value = textBox1.Text;

dataGridView1.Rows[i].Cells[2].Value = Convert.ToInt32(numericUpDown1.Value);

dataGridView1.Rows[i].Cells[3].Value = textBox2.Text;

Device s = deivices[i+1];

dataGridView1.Rows[i].Cells[4].Value = s.Get\_scale(Convert.ToInt32(numericUpDown1.Value),Convert.ToInt32(numericUpDown2.Value));

dataGridView1.Rows[i].Cells[5].Value = textBox3.Text;

}

private void поискToolStripMenuItem\_Click(object sender, EventArgs e)

{

textBox1.Visible = false;

textBox2.Visible = false;

textBox3.Visible = false;

numericUpDown1.Visible = false;

numericUpDown2.Visible = false;

label1.Visible = false;

label2.Visible = false;

label3.Visible = false;

label4.Visible = false;

label5.Visible = false;

textBox4.Visible = true;

button1.Text = "search";

}

private void PerformSearch(string searchText)

{

string searchValue = textBox4.Text;

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

try

{

foreach (DataGridViewRow row in dataGridView1.Rows)

{

foreach (DataGridViewCell cell in row.Cells)

{

if (cell.Value != null && cell.Value.ToString().Contains(searchValue))

{

dataGridView1.CurrentCell = cell;

row.Selected = true;

dataGridView1.CurrentRow.ToString();

break;

}

}

}

}

catch (Exception exc)

{

MessageBox.Show(exc.Message);

}

}

}

Device

private string name;

private int limits;

private string units;

private double scale ;

private string type;

private int divisions;

public Device(string name\_,int limits\_,string units\_,string tyep\_,int divisions\_)

{

name = name\_;

limits = limits\_;

units = units\_;

type = tyep\_;

divisions = divisions\_;

}

private double Get\_calculation()

{

scale = (limits - 0) / divisions;

return scale;

}

private double Get\_calculation(int a,int b)

{

scale = (a - 0) / b;

return scale;

}

public string Get\_name()

{

return name;

}

public int Get\_limits()

{

return limits;

}

public string Get\_units()

{

return units;

}

public double Get\_scale()

{

Get\_calculation();

return scale;

}

public double Get\_scale(int a,int b)

{

Get\_calculation(a,b);

return scale;

}

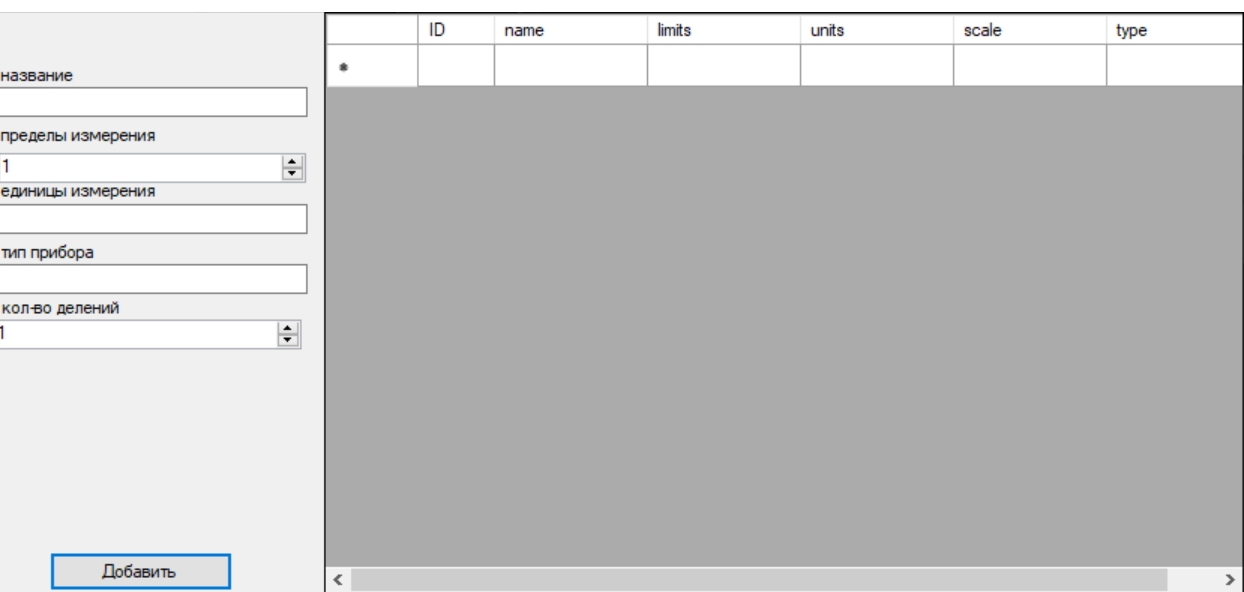
public string Get\_type()

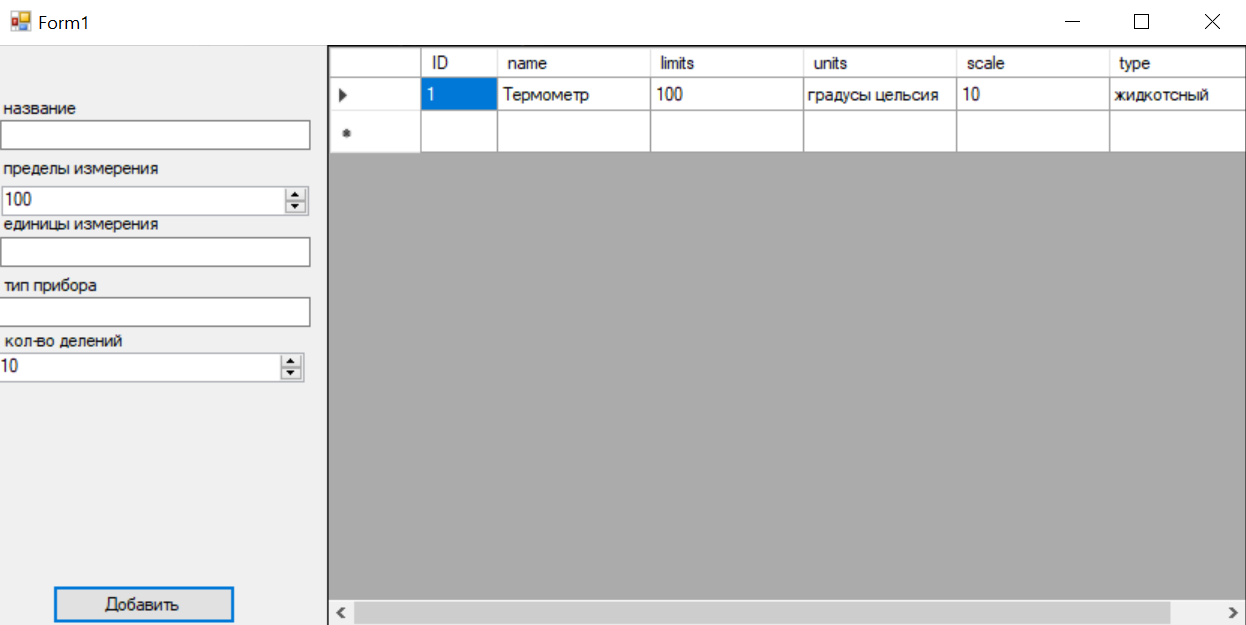
{

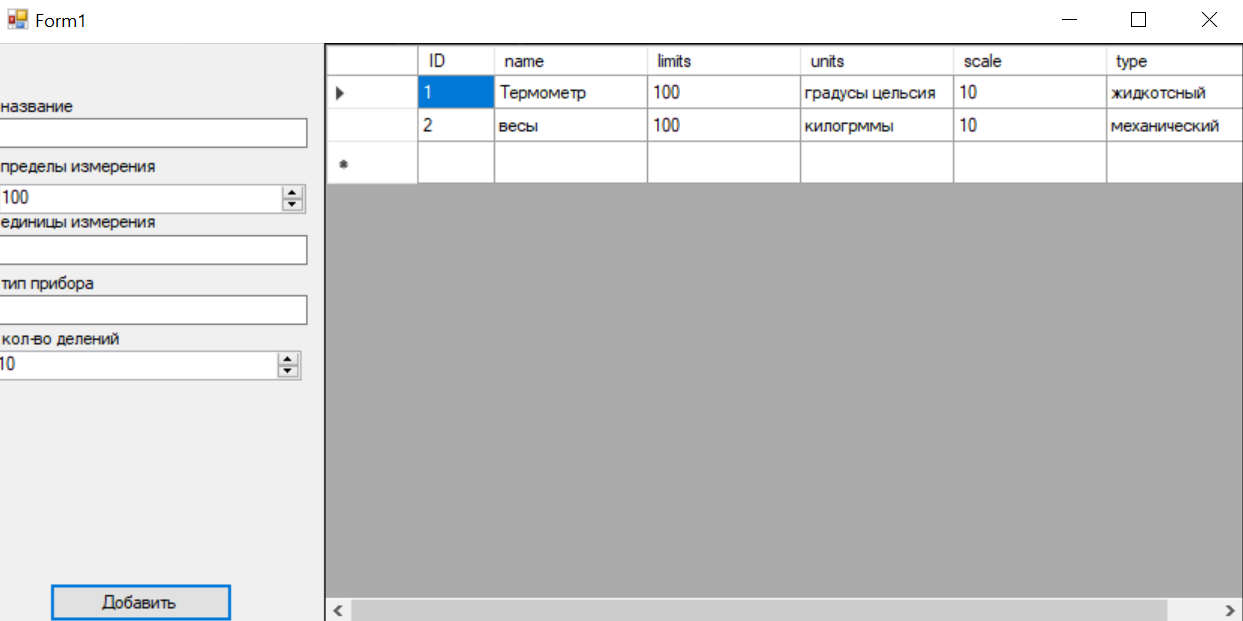
return type;

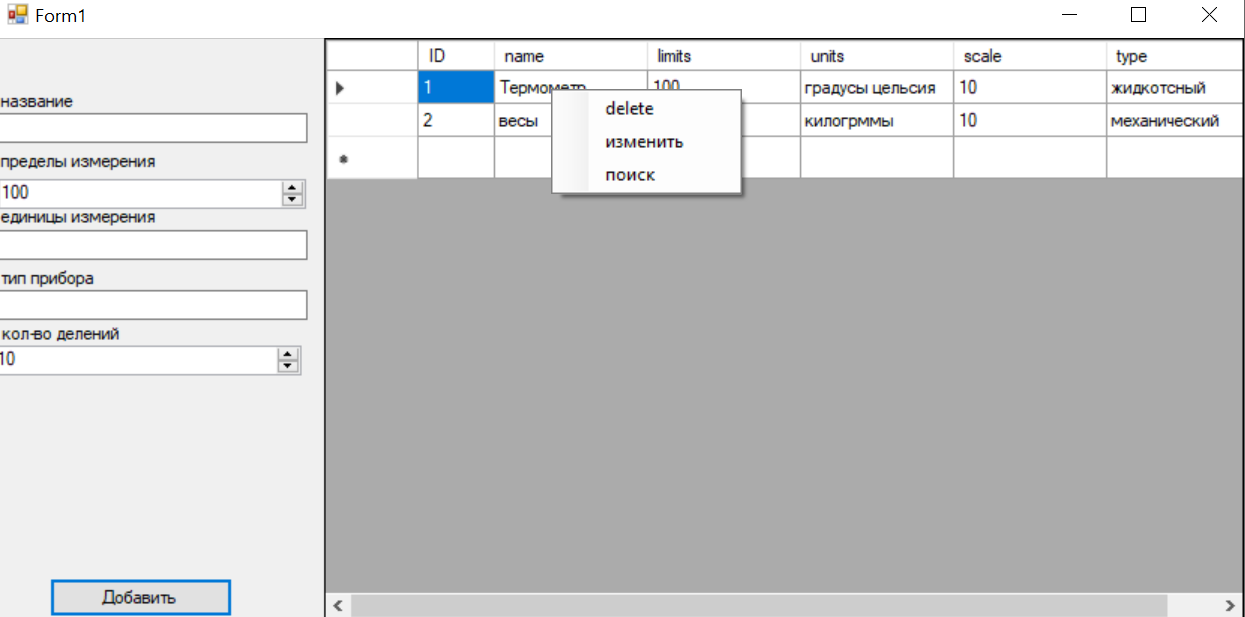
}

**Тестовые ситуации**

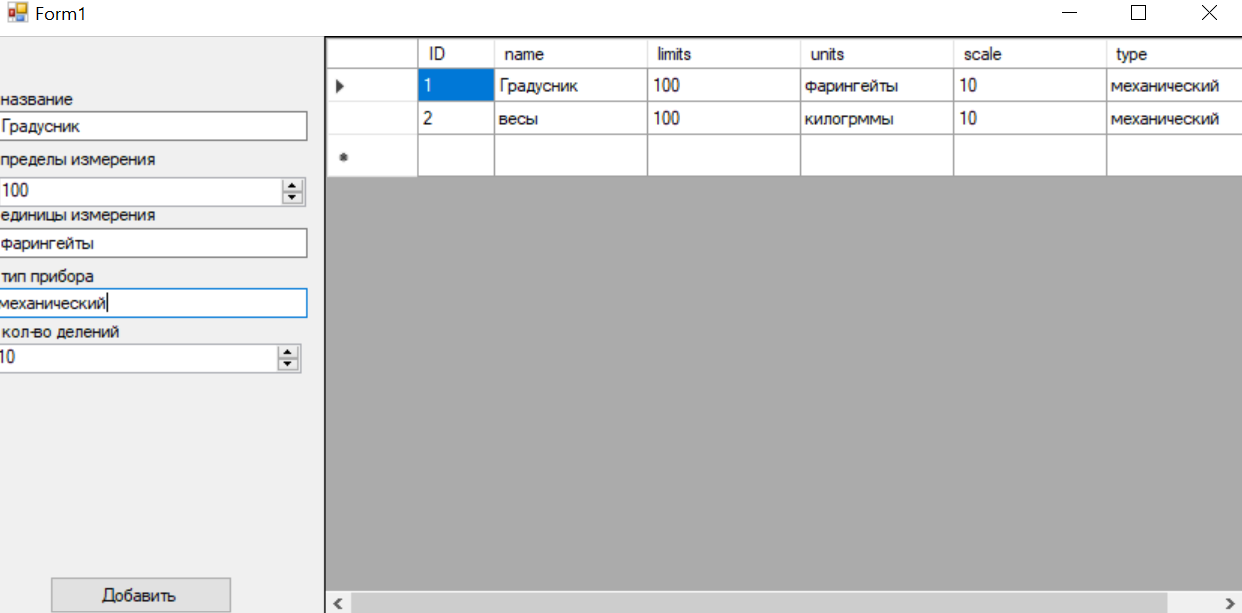
****

****

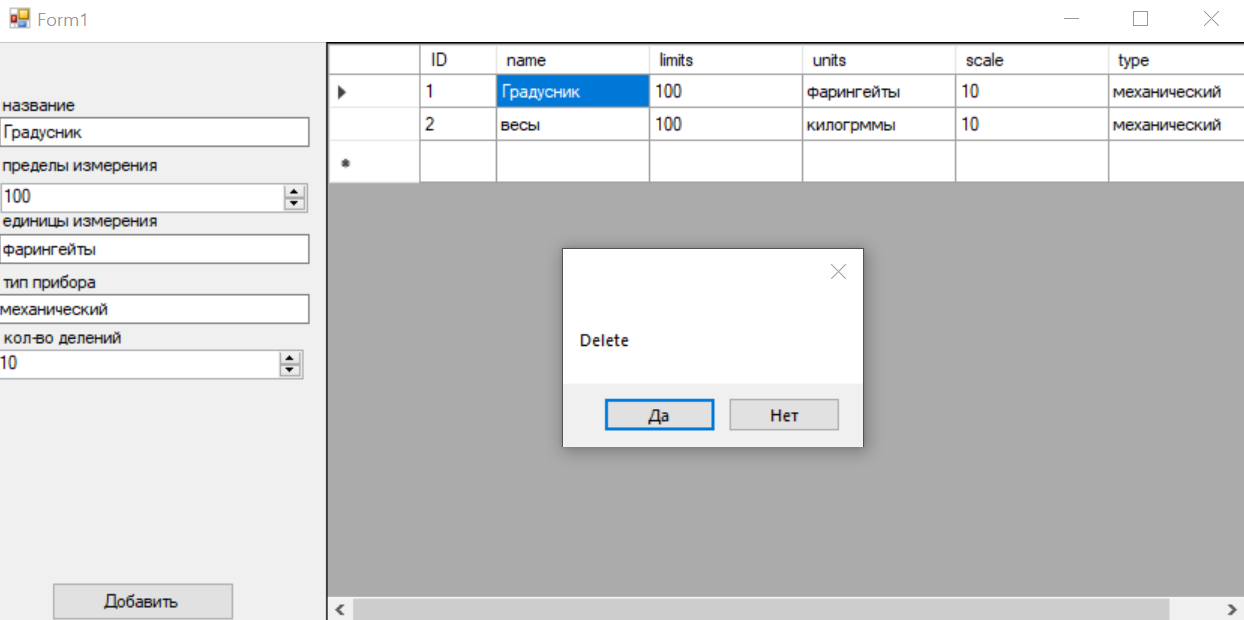
****

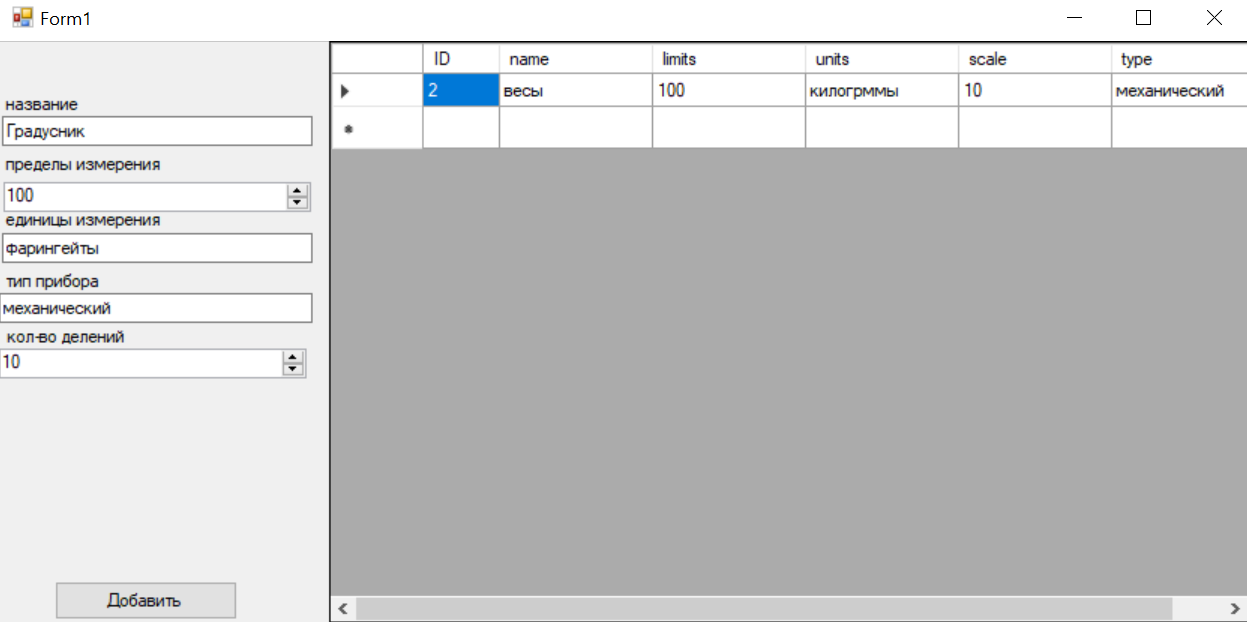
****

**Изменение**

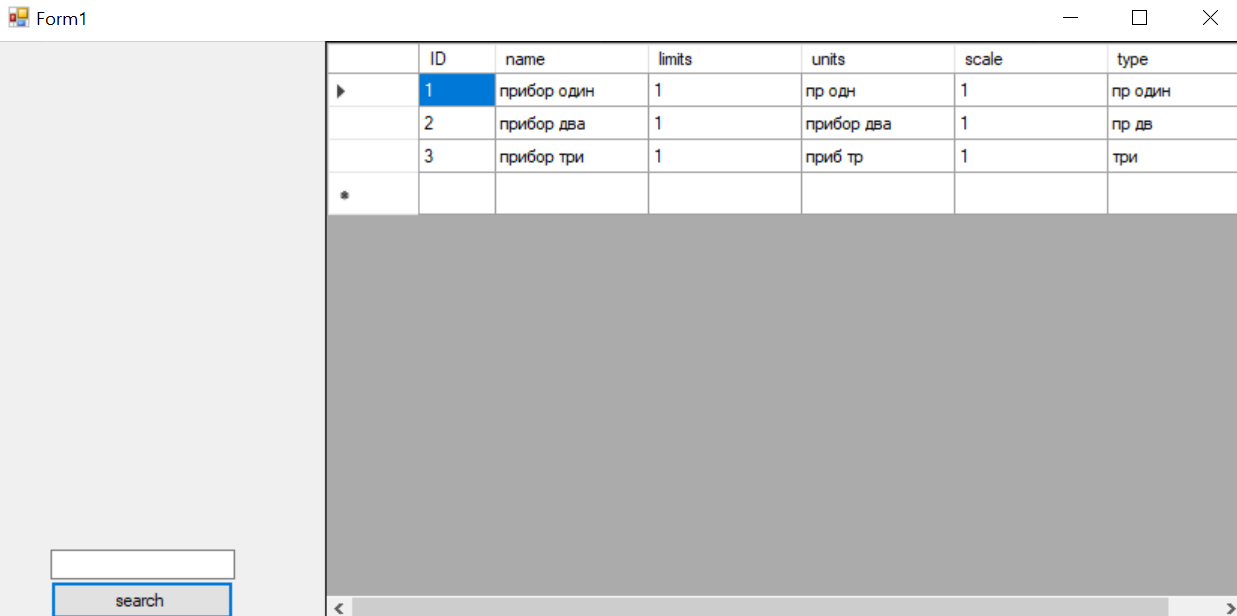
****

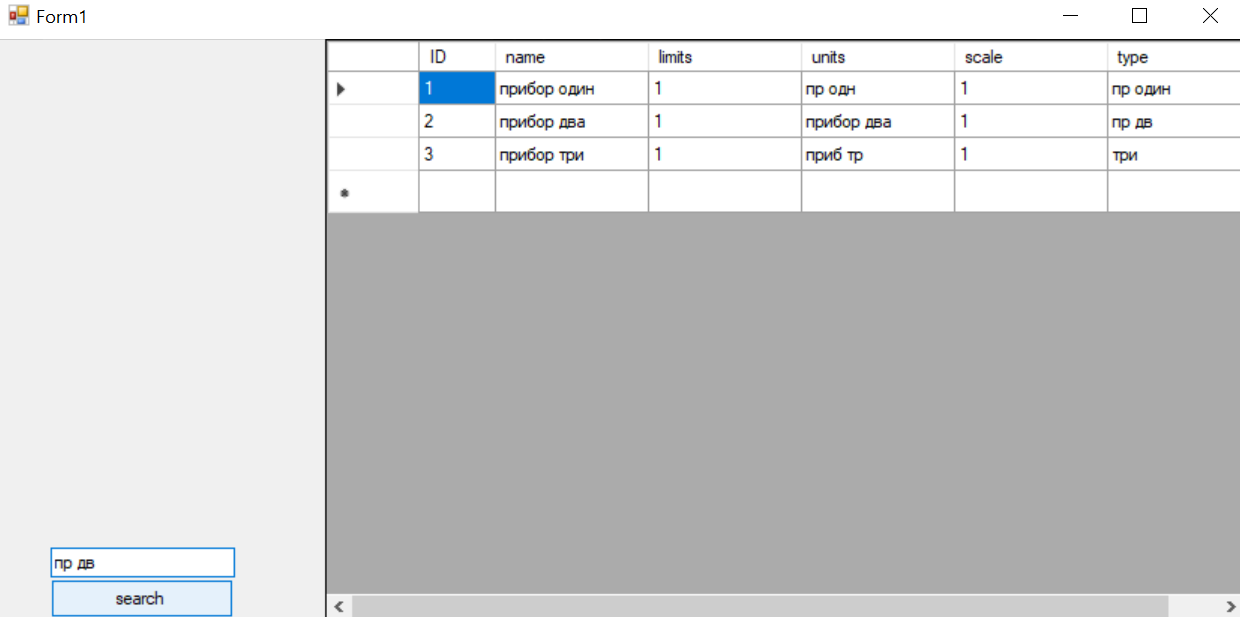
**Удаление**

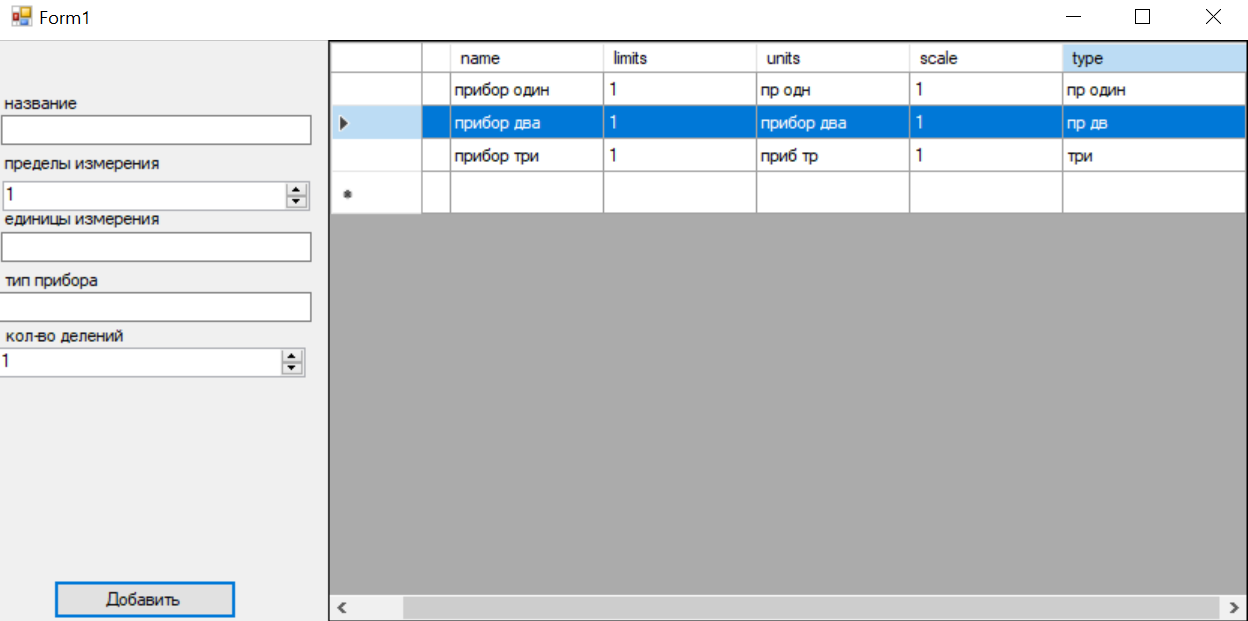
****

****

**Поиск**

****

****

****

при поиске информации выводится отдельное текстовое поле в которое мы вводим то что мы хотим найти если поле найдено то нас переносит к этой строке она также будет выделена и будет выделен столбец в котором нашлось то слово которое мы искали. Если же не будет найдено то словно которое мы вводили то тогда выдастся ошибка.

**Вывод**

Выполняя эту работу, я изучил новые способы работы с классом DataGridView, так же поработал с новой для меня коллекцией SortedList.

А также улучшил свои навыки работы с WindiwsForms