

**Национальный исследовательский университет**  
**«Высшая школа экономики»**

**Факультет компьютерных наук**

Департамент  
**Программной инженерии**

***Контрольное домашнее задание***  
***по дисциплине***  
***«Программирование»***

Тема работы: Программа для просмотра, обработки и сохранения результатов обработки данных из файла
---

Выполнил: студент группы 183 (2)

\_\_\_\_\_ Верко Р.А.

тел: \_\_\_\_\_ +79196160900

e-mail адрес: raverko@edu.hse.ru

Преподаватель: Чуйкин Николай Константинович

## Содержание

1. Условие задачи.....	3
2. Функции разрабатываемого приложения .....	4
2.1 Варианты использования .....	4
2.2 Описание интерфейса пользователя .....	4
3. Структура приложения .....	5
3.1. Диаграмма классов.....	5
3.2 Описание классов, их полей и методов .....	6
4. Распределение исходного кода по файлам проекта.....	8
5. Контрольный пример и описание результатов.....	9
6. Сообщения пользователю.....	14
7. Код программы .....	15
8. Список литературы.....	33

## 1. Условие задачи

Разработать оконное приложение Windows Forms Application, позволяющее:

1. Обработать данные из CSV-файла и проверять корректность данных в нем.
2. Отображать данные в оконном приложении.
3. Создавать новые элементы путем ввода новых данных пользователем.
4. Позволить удаление элементов пользователем, редактирование уже существующих.
5. Сохранять данные в указанном пользователем текстовом файле в режиме:
  - 5.1. Сохранения в текущем файле.
  - 5.2. Добавления в файл текущих данных.
  - 5.3. Сохранения в новом файле.
6. Фильтрацию данных по алфавиту и возрастанию полей Name, Owner.
7. Сортировку данных по количеству остановок у владельца.
8. Фильтрацию данных по полям Flow, Road.

## 2. Функции разрабатываемого приложения

### 2.1 Варианты использования

Данная программа может использоваться главным образом в целях обучения студентов, для наглядного примера работы с файлами и для практического применения работе с файлами стандарта CSV, их создания и редактирования. Также возможно использование в развлекательных целях.

### 2.2 Описание интерфейса пользователя

Интерфейс пользователя представляет собой окно (рис 1), большую часть которого занимает объект DataGridView (1.1), позволяющий представлять обрабатываемые данные в виде сводной таблицы. В верхней строке, представляющей собой меню программы находятся 3 вкладки: File, Edit, Info. (1.2). Во вкладке File находится панель элементов, позволяющих создать, сохранить и отсортировать выбранный файл. Пункт Edit позволяет пользователю с легкостью редактировать выбранный элемент таблицы, а так же добавить новый, с помощью нового диалогового окна. Последний пункт – Info позволяет вывести сообщение типа MessageBox с данными о разработчике данной программы. Так же, на панели представлены две кнопки типа Button с названиями Flow и Road (1.3) и поле для ввода TextBox (1.4), с помощью которых можно провести фильтрацию соответствующих колонок в таблице, вывести их в отдельное окно и сохранить в файл.

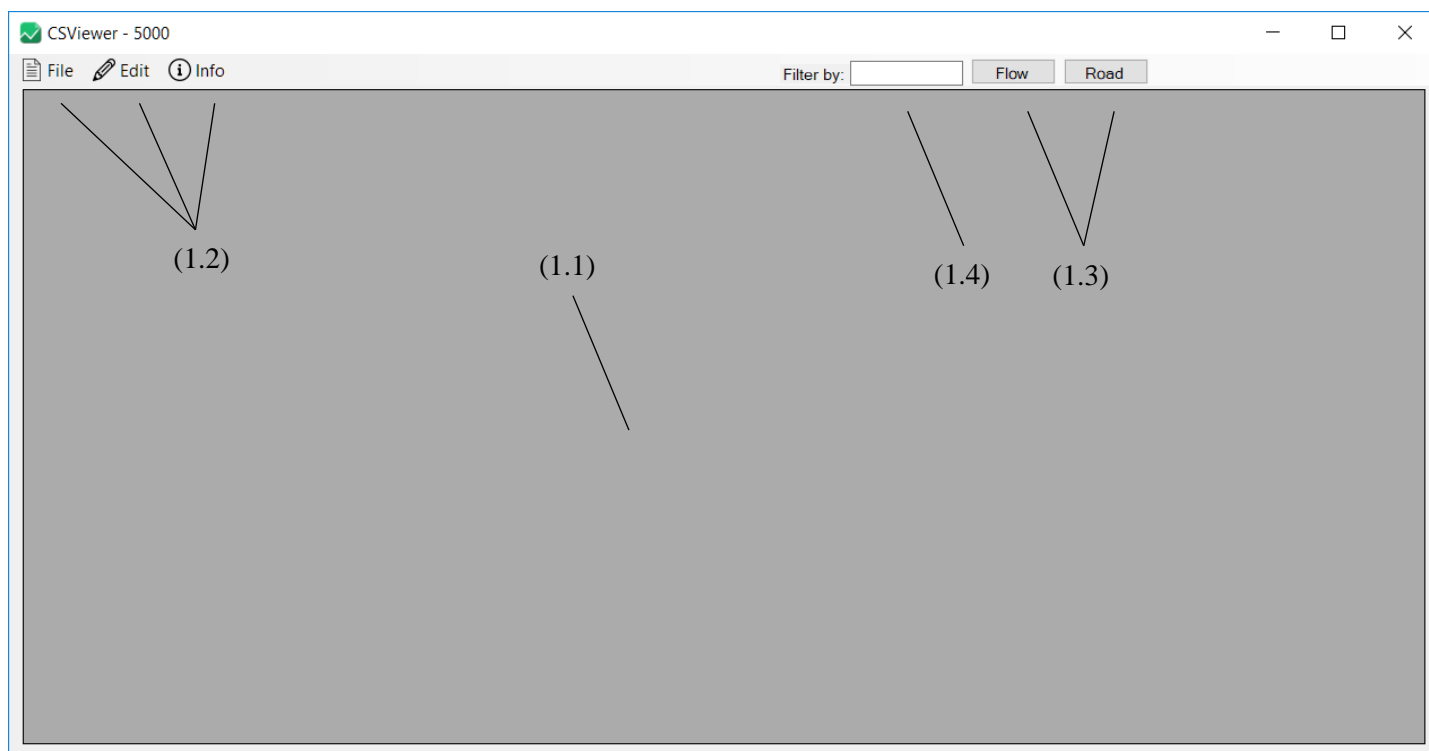


Рис. 1. Основное окно программы

### 3. Структура приложения

#### 3.1. Диаграмма классов

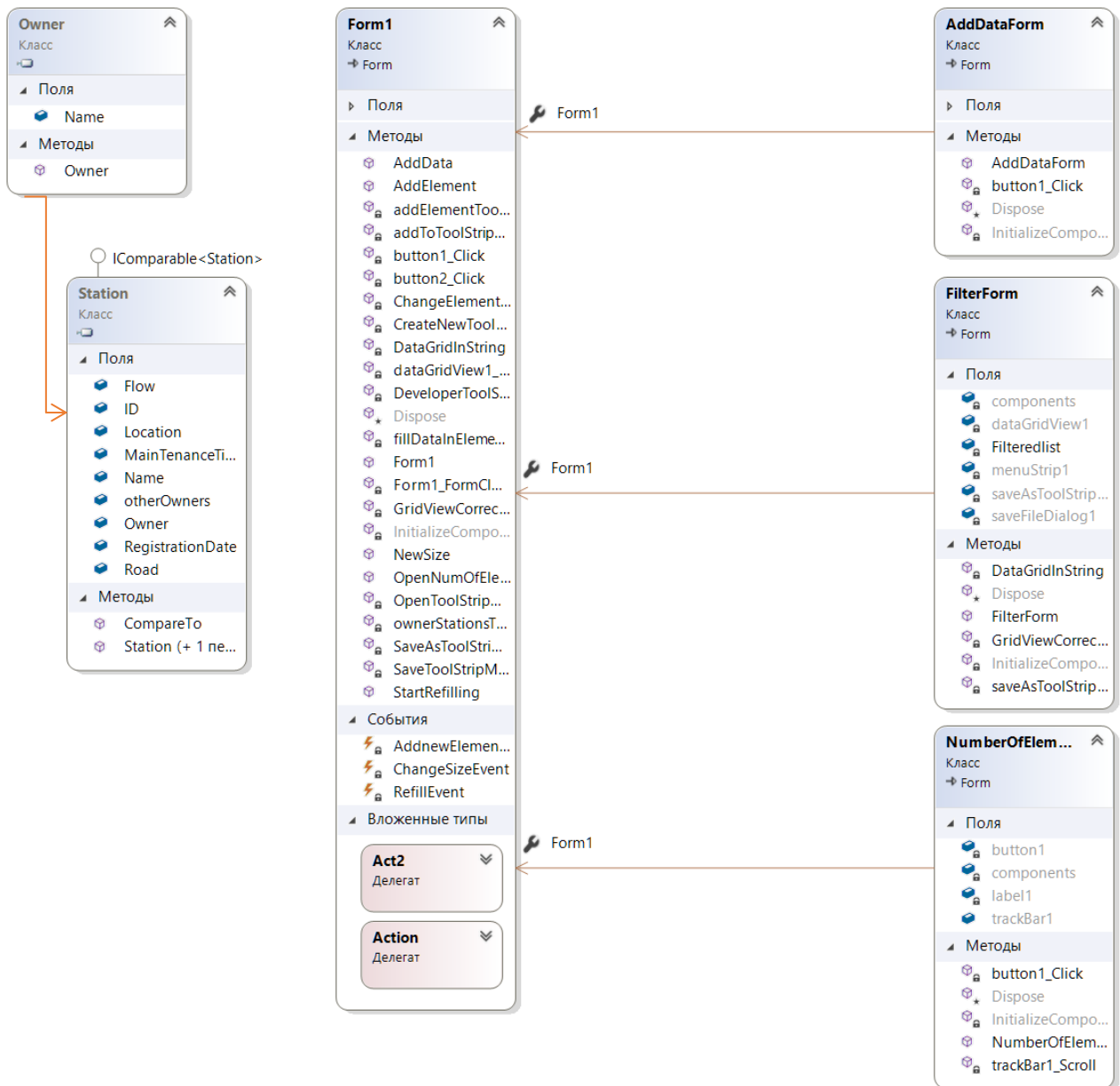


Рис. 2. Диаграмма классов

### 3.2 Описание классов, их полей и методов

- `public class Station` – Класс станция
  - `public string Name;` - название
  - `public int ID;` - пер. номер
  - `public string RegistrationDate;` - дата регистрации
  - `public string Location;` - расположение
  - `public Owner Owner;` - владелец (агрегация)
  - `public int Flow;` - поток(дорога)
  - `public int MainTenanceTime;` - время обслуживания
  - `public string Road;` - шоссе
  - `public int otherOwners;` - количество станций у этого же владельца
  - `public Station()` - конструктор
  - `string CorrectWord(string old)` – корректировка имени
  - `public int CompareTo(Station stat)` – реализация интерфейса сравнения
- `public class Owner` – класс владельца
  - `public string Name;` - имя
  - `public Owner(string a)` - конструктор
- `public partial class Form1: Form` – класс основной формы
  - `public Form1()` - конструктор
  - `public List<int> ElementsWithoutData = new List<int>();` - список номеров элементов с недостающей информацией
  - `public delegate void Action(int num);` - делегат для добавления информации
  - `private static event Action ChangeSizeEvent;` - событие для добавления информации, вызываемое из других классов
  - `public static int RefillItems;` - количество элементов с ошибками
  - `public delegate void Act2(int position, Station newstation);` - делегат для вставки новых элементов таблицы
  - `private static event Act2 RefillEvent;` - событие
  - `private static event Act2 AddnewElementEvent;`
  - `public static void AddElement(int nomatter, Station station) { AddnewElementEvent(nomatter, station); }`
  - `public static void StartRefilling(int position, Station newstation) { RefillEvent(position,newstation); }` – добавление всей недостающей информации
  - `public static void NewSize(int num) { ChangeSizeEvent(num); }` – изменение количества отображаемых элементов
  - `public List<Station> stations = new List<Station>();` - список станций
  - `public List<int> NumOfErrors = new List<int>();` - номера элементов с ошибками
  - `bool IsOpened = false;` - открыт ли файл
  - `string filename;` - путь к файлу
  - `public void OpenNumOfElemform()` – метод открытия файла

- `int minID = 0, maxID = 0, minFlow = 0, maxFlow = 0, minMT = 0, maxMT = 0;` переменные для сортировки
- `private void OpenToolStripMenuItem_Click(object sender, EventArgs e)` – открытие нового файла
- `private void fillDataInElementsToolStripMenuItem_Click(object sender, EventArgs e)` – метод заполнения недостающей информации
- `private void DeveloperToolStripMenuItem_Click(object sender, EventArgs e)` – вывод сообщения о разработчике
- `private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)` – редактирование элемента двойным нажатием
- `private void ChangeElementToolStripMenuItem_Click(object sender, EventArgs e)` – редактирование элемента из меню
- `private void ownerStationsToolStripMenuItem_Click(object sender, EventArgs e)` – сортировка по убыванию станций у владельца
- `private void button1_Click(object sender, EventArgs e)` – фильтрация по flow
- `private void button2_Click(object sender, EventArgs e)` – фильтрация по road
- `private void addElementToolStripMenuItem_Click(object sender, EventArgs e)` – добавление нового элемента из меню
- `private void SaveToolStripMenuItem_Click(object sender, EventArgs e)` – сохранить файл
- `private void addToToolStripMenuItem_Click(object sender, EventArgs e)` – добавить строки к имеющемуся файлу
- `private void Form1_FormClosing(object sender, FormClosingEventArgs e)` – вопрос о сохранении при выходе
- `private void SaveAsToolStripMenuItem_Click(object sender, EventArgs e)` – сохранение с изменением пути
- `private void CreateNewToolStripMenuItem_Click(object sender, EventArgs e)` – создание нового файла
- `private void GridViewCorrection()` – добавление оформления столбцов
- `public Station AddData(Station station, int n, int m)` – создание текстовой переменной с информацией об элементах в таблице

#### **4. Распределение исходного кода по файлам проекта**

1. Form1.cs – файл содержит основную форму, обеспечивающую интерфейс пользователя и позволяющую работать с данными.
2. AddDataForm.cs - представляет собой диалоговое окно, позволяющее ввести или изменить данные любой из ячеек.
3. FilterForm.cs – файл содержащий форму вывода отфильтрованных пользователем значений и возможностью их сохранения.
4. NumberOfElementsForm.cs – файл диалогового окна, дающий возможность изменить количество показываемых элементов.
5. Class1.cs – файл в библиотеке классов ClassLibrary1, в котором объявлены такие классы, как station, owner.



## 5. Контрольный пример и описание результатов

1. Запустим выполняемый файл .exe , чтобы запустить наше приложение.
2. Перед нами откроется пустое окно программы (рис. 1).
3. Для начала откроем вкладку File > Open и выберем нужный нам CSV-файл (рис. 2)

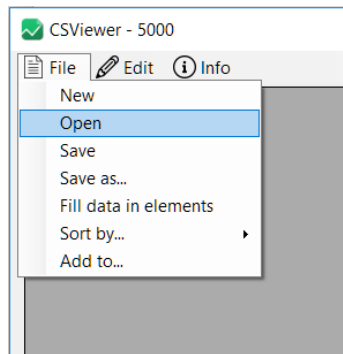


Рис.2

4. Перед открытием программа считывает файл, и сообщает пользователю о количестве элементов без некоторой необходимой информации, а так же о количестве строк с неправильным форматом данных, с просьбой заполнить их. (рис.3)

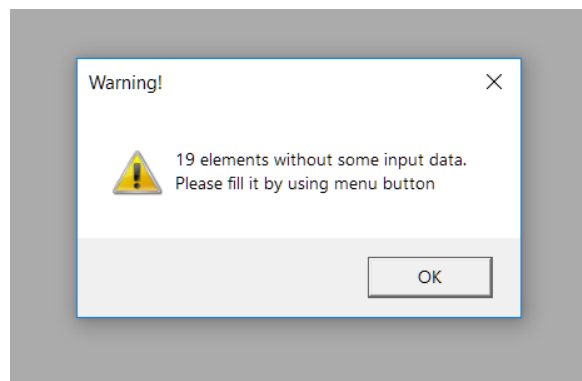


Рис.3

5. Далее возникает окно с ползунком, позволяющее изменять число отображаемых элементов (рис. 4)

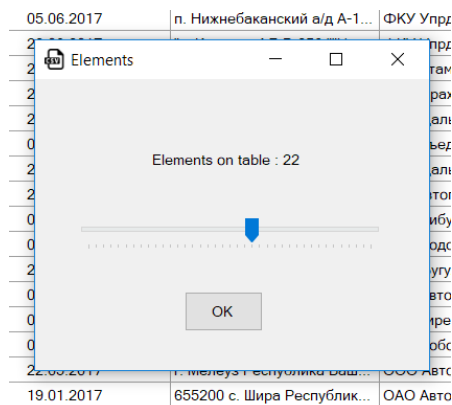


Рис. 4

6. Перед пользователем открывается таблица типа Data grid view, с возможностью редактирования существующих и добавления новых элементов. Для этого откроем вкладку Edit > Add element. (рис. 5) Откроется диалоговое окно с полями для ввода данных. (рис. 6) Поля Flow, ID, Maintenance time – числовые, и принимают только целочисленные значения. После ввода корректной информации нужная строка будет добавлена в конец списка. (рис. 7)

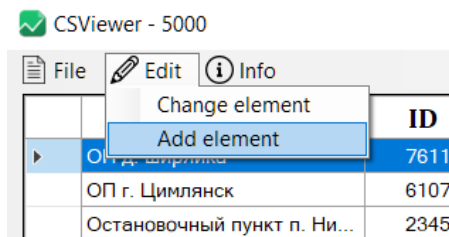


Рис. 5

Рис. 6

Name	ID	Registration date	Location	Owner	Flow	Maintenance time
ОП п. Каштан	24093	05.05.2017	"п. Каштан ФАД М-53 "Бел...	ФКУ Байкалуправтодор	24	5
ОП г. Белоусово	40086	20.05.2017	г. Белоусово в/д А-130 Мос...	ФКУ Упрдор Москва - Боб...	12	5
ОП с. Хлевное	48009	22.03.2017	399260 с. Хлевное Липецк...	ИП Ларин Валерий Иванов...	13	5
Кассовый пункт г. Болхов	57001	11.07.2016	г. Болхов Орловская обл., ...	АО Орелавтотранс	24	10
АС г. Кимовск	71005	04.10.2016	301721 г. Кимовск Тульская...	ООО Актив инвест	23	2
Благодарненская автоста...	26001	02.08.2016	356420 г. Благодарный Ст...	ОАО Автовокзал	12	5
АС г. Ставрополь (ЖДВ)	26102	18.07.2016	355008 г. Ставрополь г. Ст...	ИП Семенов А.В.	3	2
ОП п. Новостройка	27015	20.04.2017	п. Новостройка в/д А-370 ...	ФКУ Дальуправтодор	12	5
Кассовый пункт с. Верхние...	2033	20.08.2016	452500 с. Верхние Киги Ре...	ГУП Башавтотранс	6	2
ОП с. Пшеда 1475км	23394	05.05.2017	с. Пшеда в/д М-4 «Дон» М...	ГК Автодор	20	1
АВ ст. Зеленчукская	9004	22.05.2017	ст. Зеленчукская Карачае...	ИП Биджиев Осман Шагаб...	2	10
ОП с. Краснопартизанское	23181	03.04.2017	с. Краснопартизанское в/д...	ГК Автодор	20	1
ОП д. Мальцево	67064	20.04.2017	д. Мальцево в/д М-1 «Бела...	ГК Автодор	12	2
ДКП г. Маринский Посад	21005	19.01.2017	429570 г. Маринский Пос...	АО АвтоВАС	24	5
АВ Щербинки г. Нижний Но...	52111	20.10.2016	603137 г. Нижний Новгоро...	ООО Щербинки	49	5
АС с. Большое Игнатово	13014	29.09.2016	431670 с. Большое Игнато...	ОАО Объединение автово...	6	5
АС с. Романово	22017	21.11.2016	658640 с. Романово Алтай...	ООО Романовская автост...	2	10
Остановочный пункт с. Ма...	4030	05.06.2017	с. Майма в/д Р-256 «Чуйск...	Администрация муниципа...	30	1
Остановочный пункт п. Це...	23414	05.06.2017	п. Цемдолина А-290 Новор...	ФКУ Упрдор Черноморье	12	5
АВ г. Октябрьский	2004	20.08.2016	452620 г. Октябрьский Рес...	ГУП Башавтотранс	30	3
АВ г. Дюртюли	2003	20.08.2016	452320 г. Дюртюли Респуб...	ГУП Башавтотранс	24	4
ОП г. Москва	13	12.11.2019	Москва	Owner's name	13	10

Рис. 7

7. Для редактирования информации сделаем двойной щелчок по нужному элементу. Откроется уже знакомое пользователю окно ввода информации. (рис. 8) После нажатия кнопки «ОК» программа проверит введенную информацию и внесет ее в нужную позицию.

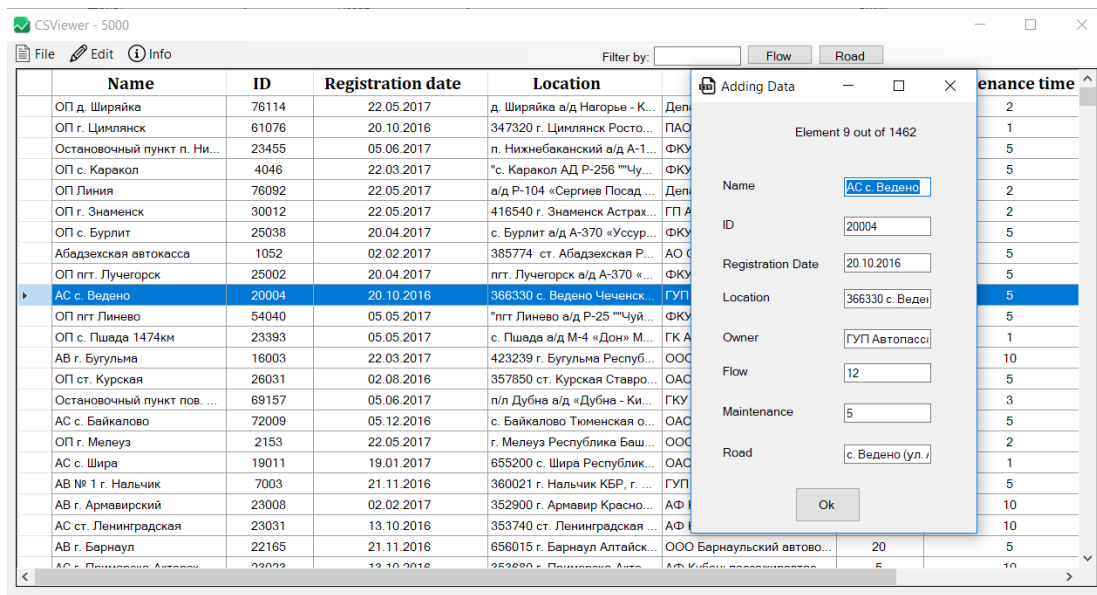


Рис. 8

8. Чтобы сохранить данные пройдем по вкладке File > Save или File > Save as... в зависимости от того, хотите ли вы сохранить данные в тот же файл либо изменить его местоположение в памяти устройства. (рис. 9-11).

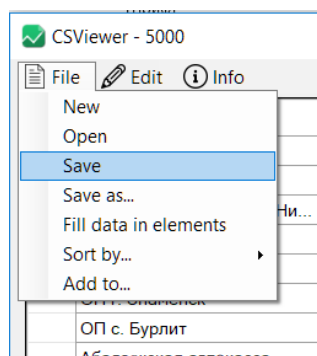


Рис. 9

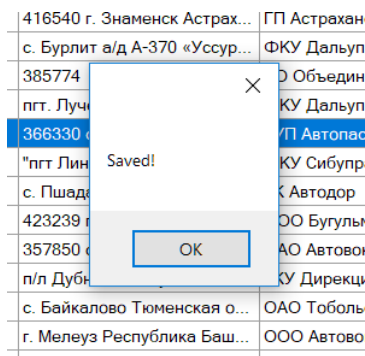


Рис.10

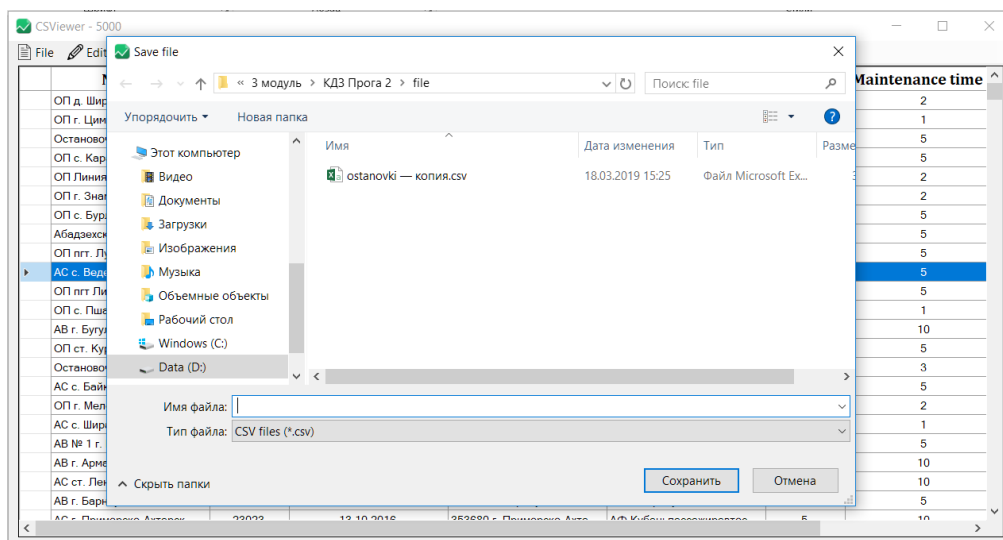


Рис. 11

9. Так же, любой столбец можно отсортировать, кликнув по заголовку. Текстовые поля программа будет сортировать по алфавиту, а числовые – по возрастанию или убыванию. (рис. 12-13)

CSViewer - 5000

File Edit Info

	Name ▲	ID
	Абадзехская автокасса	1050
	АВ «Центральный» г. Гроз...	2000
	АВ «Черемушки» г. Кизляр	5020
	АВ № 1 г. Владикавказ	1500
	АВ № 1 г. Нальчик	7000
	АВ № 2 г. Владикавказ	1500
	АВ Абинский	2300
	АВ Волоколамск	5000
	АВ г. Абакан	1901
	АВ г. Алексин	7100
	АВ г. Альметьевск	1600
	АВ г. Анапа	2300
	АВ г. Апшеронский	2300
	АВ г. Армавирский	2300
	АВ г. Асбест	6600
	АВ г. Астрахань	3000
	АВ г. Баймак	2000
	АВ г. Барнаул	2216
	АВ г. Белгород	3100
	АВ г. Белебей	2000
	АВ г. Белово	4200
	АВ г. Белогорск	2800
	АВ г. Боровицк	2000

Рис. 12

ID ▲	R
1001	
1002	
1003	
1004	
1005	
1008	
1009	
1038	
1045	
1052	
1059	
1065	
1072	
1082	
1096	
1099	
1114	
1142	
1146	
1153	
1154	
1155	
1156	

Рис. 13

10. Наконец выполним фильтрацию элементов. Введем в поле (рис. 1.4) некоторое значение и выберем столбец, по которому будет проводиться фильтрация. (рис 14) В новом окне появятся результаты сортировки, которые так же можно сохранить в новый файл.( рис. 15)

Filter by: 24 | Flow | Road

on	Owner	Flow	Maintenance time
хабль Респ...	АО Объединение автовокз...	24	5
п Республ...	АО Объединение автовокз...	25	2
жий Респу...	АО Объединение автовокз...	24	5
югвардейс...	АО Объединение автовокз...	24	5

Рис. 14

CSV Viewer - 5000

File Edit Info Filter by: 24 Flow Road

Filter Save as

Name	ID	Registration date	Location	Owner	Flow	Mainten
Кошехабльская автокасса	1001	02.02.2017	385400 а. Кошехабль Респ...	АО Объединение автовокз...	24	
Тульская автокасса	1003	02.02.2017	385730 п. Тульский Респу...	АО Объединение автовокз...	24	
Красногвардейская автока...	1004	02.02.2017	385300 с. Красногвардейс...	АО Объединение автовокз...	24	
АВ г. Дюртюли	2003	20.08.2016	452320 г. Дюртюли Респу...	ГУП Башавтотранс	24	
АВ № 1 г. Нальчик	7003	21.11.2016	360021 г. Нальчик КБР, г. ...	ГУП Каббалкавтотранс-14...	24	
ОП с. Кетченеры	8005	15.12.2016	с. Кетченеры Республика ...	ООО Транс-Юг	24	
ОП с. Малые Дербеты	8010	15.12.2016	с. Малые Дербеты Респу...	ООО Транс-Юг	24	
ОП с. Садовое	8012	15.12.2016	с. Садовое Республика Ка...	ООО Транс-Юг	24	
АВ г. Набережные Челны	16016	20.08.2016	423800 г. Набережные Че...	ООО Экспресс-Кама НЧ	24	
Кассовый пункт г. Абакан	19001	22.03.2017	"655017 г. Абакан Респу...	ОАО Автоколонна 1967	24	
ОП с. Троицкое	19018	05.05.2017	"с. Троицкое ФАД М-54 "	ФКУ Байкалуправтодор	24	
ОП с. Сарагаш	19049	05.05.2017	"с. Сарагаш ФАД М-54 "Е...	ФКУ Байкалуправтодор	24	
ОП пгт. Пригорск	19050	05.05.2017	"пгт. Пригорск ФАД М-54 "	ФКУ Байкалуправтодор	24	
ДКП г. Маринский Посад	21005	19.01.2017	429570 г. Маринский Пос...	АО АвтоВАС	24	
Кассовый пункт ст. Новоп...	23003	13.03.2017	353020 ст. Новопокровска...	ОАО «Кубаньпассажир-ав...	24	
Кассовый пункт с. Са...	1103	05.06.2017	с. Садовое а/д А-100 "Кам...	ФКУ Упрдор Черноморье	12	а/д А-160
Кассовый пункт с. Пр...	1154	05.06.2017	с. Преображенское а/д А-1...	ФКУ Упрдор Черноморье	12	а/д А-160
Кассовый пункт ст. К...	1155	05.06.2017	ст. Кирпильская а/д А-160 ...	ФКУ Упрдор Черноморье	12	а/д А-160
Кассовый пункт ст. К...	1156	05.06.2017	ст. Кирпильская а/д А-160 ...	ФКУ Упрдор Черноморье	12	а/д А-160

Рис.15

## 6. Сообщения пользователю

1. При попытке открыть файл по несуществующему пути пользователю выводится сообщение типа MessageBox с текстом «Файл не найден. Повторите попытку».
2. При попытке изменить элемент, не выбрав ни один, пользователю выводится сообщение типа MessageBox с текстом «Select any row»
3. При вводе недопустимых значений элемента пользователю выводится сообщение типа MessageBox с текстом «Wrong data in *название поля* »
4. При попытке сохранить данные в уже существующий файл, пользователю выводится сообщение типа MessageBox с текстом «Такой файл уже существует. Вы хотите изменить его?»
5. При открытии нового файла пользователю выводится сообщение типа MessageBox с текстом «Are you want to save it? Your data can be lost»
6. При создании нового файла пользователю выводится сообщение типа MessageBox с текстом «Are you want to save it? Your data can be lost»

## 7. Код программы

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using ClassLibrary1;

namespace CSVViewer_5000
{
    public partial class Form1: Form
    {

        public Form1()
        {
            InitializeComponent();
            openFileDialog1.Filter = "CSV files (*.csv)|*.csv";
            saveFileDialog1.Filter = "CSV files (*.csv)|*.csv";
            saveFileDialog1.RestoreDirectory = true;
            RefillEvent += (position, newstation) => {
                dataGridView1.Rows.RemoveAt(position);
                dataGridView1.Rows.Insert(position, newstation.Name,
newstation.ID, newstation.RegistrationDate, newstation.Location,
newstation.Owner.Name, newstation.Flow, newstation.MainTenanceTime,
newstation.Road);
            };
        }
    }
}
```

```

    };

    AddnewElementEvent += (position, newstation) => {
        dataGridView1.Rows.Add(newstation.Name, newstation.ID,
newstation.RegistrationDate, newstation.Location, newstation.Owner.Name,
newstation.Flow, newstation.MainTenanceTime, newstation.Road);
    };
}

public List<int> ElementsWithoutData = new List<int>();
public delegate void Action(int num);
private static event Action ChangeSizeEvent;
public static int RefillItems;
public delegate void Act2(int position, Station newstation);
private static event Act2 RefillEvent;
private static event Act2 AddnewElementEvent;
public static void AddElement(int nomatter, Station station) {
AddnewElementEvent(nomatter, station); }
public static void StartRefilling(int position, Station newstation)
{ RefillEvent(position,newstation); }

public static void NewSize(int num) { ChangeSizeEvent(num);

} //Resizing Form1 using data from NumberOfElementsForm

public List<Station> stations = new List<Station>();
public List<int> NumOfErrors = new List<int>();
public List<string> NameOfErrors = new List<string>();
bool IsOpened = false;
string filename;
public void OpenNumOfElemform()
{
    NumberOfElementsForm form2 = new NumberOfElementsForm();

```



```
form2.Show();
ChangeSizeEvent += (num) => { Height = 144 + 24 * num; };
} // Correcting number of visible elements

int minID = 0, maxID = 0, minFlow = 0, maxFlow = 0, minMT = 0, maxMT
= 0;

private void OpenToolStripMenuItem_Click(object sender, EventArgs e)
{

    if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
        return;
    // получаем выбранный файл

    if (IsOpened)
    {
        DialogResult dialogResult = MessageBox.Show("Do you want to
save this file?\nAll your changes can be lost.", "Save",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dialogResult == DialogResult.Yes)
        {
            SaveToolStripMenuItem_Click(sender, e);
        }
        IsOpened = false;
    }
    IsOpened = true;
    dataGridView1.Rows.Clear();
    dataGridView1.Columns.Clear();
    stations.Clear();
    GridViewCorrection();
    OpenNumOfElemform();
}
```

```
filename = openFileDialog1.FileName;
NumOfErrors.Clear();
NameOfErrors.Clear();
ElementsWithoutData.Clear();
try
{
    using (StreamReader sr = new StreamReader(filename,
Encoding.Default))
    {
        int ID, Flow, MT,num = -1;

        string Line ;
        string[] strarr;

        while ((Line = sr.ReadLine()) != null)
        {
            if
(Line.Equals("Name;ID;RegistrationDate;Location;Owner;Flow;MaintenanceTime;R
oad")) continue;

            num++;
            strarr = Line.Split(';');
            if (strarr.Length!=8)
            {
                NumOfErrors.Add(num);
                NameOfErrors.Add(strarr[0]);
                continue;
            }

            int.TryParse(strarr[1], out ID);
            int.TryParse(strarr[5], out Flow);
            int.TryParse(strarr[6], out MT);
```

```
stations.Add(new Station(strarr[0], ID, strarr[2],
strarr[3], new Owner(strarr[4]), Flow, MT, strarr[7]));

        if (strarr[0]=="" || ID==0 || strarr[2]==""||
strarr[3]=="" || strarr[4]=="" || Flow == 0 || MT == 0 || strarr[7] == "")
        {
            ElementsWithoutData.Add(num);
            NameOfErrors.Add(Line);
        }

        if (num == 0) { minID = ID; maxID = ID; minFlow =
Flow;maxFlow = Flow;minMT = MT;maxMT = MT; }

        if (ID > maxID) maxID = ID;
        if (ID < minID) minID = ID;
        if (Flow > maxFlow) maxFlow = Flow;
        if (Flow < minFlow) minFlow = Flow;
        if (MT > maxMT) maxMT = MT;
        if (MT < minMT) minMT = MT;
    } // creating new elements in stations list

    for (int i = 0; i < stations.Count; i++)
    {
        dataGridView1.Rows.Add(
            stations[i].Name,
            stations[i].ID,
            stations[i].RegistrationDate,
            stations[i].Location,
            stations[i].Owner.Name,
            stations[i].Flow,
            stations[i].MainTenanceTime,
            stations[i].Road);
    } // adding rows in table
```

```
    }

}

catch (Exception er) { MessageBox.Show(er.Message); }

if (NumOfErrors.Count + ElementsWithoutData.Count != 0)
{
    string warning = "";

    if (NumOfErrors.Count != 0) warning += $"{NumOfErrors.Count}
elements with incorrect input type.\n";

    if (ElementsWithoutData.Count != 0) warning +=
$"{ElementsWithoutData.Count} elements without some input data.\n";

    warning += "Please fill it by using menu button";

    MessageBox.Show(warning, "Warning!", MessageBoxButtons.OK,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button1,
MessageBoxOptions.DefaultDesktopOnly);

    } // Warning about missing data

} // Open & parse new file

private void fillDataInElementsToolStripMenuItem_Click(object
sender, EventArgs e)
{
    int id, flow, mt;
    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        int.TryParse(dataGridView1[1, i].Value.ToString(), out id);
        int.TryParse(dataGridView1[5, i].Value.ToString(), out
flow);

        int.TryParse(dataGridView1[6, i].Value.ToString(), out mt);

        if (dataGridView1[0, i].Value.ToString() == null ||
dataGridView1[0, i].Value.ToString().Equals("")) || dataGridView1[2,
```

```

i].Value.ToString() == null || dataGridView1[2,
i].Value.ToString().Equals("") || dataGridView1[3, i].Value.ToString() ==
null || dataGridView1[3, i].Value.ToString().Equals("") || dataGridView1[4,
i].Value.ToString() == null || dataGridView1[4,
i].Value.ToString().Equals("") || dataGridView1[7, i].Value.ToString() ==
null || dataGridView1[7, i].Value.ToString().Equals(""))

        {

            Station station = new Station(dataGridView1[0,
i].Value.ToString(), id, dataGridView1[2, i].Value.ToString(),
dataGridView1[3, i].Value.ToString(), new Owner(dataGridView1[4,
i].Value.ToString()), flow, mt, dataGridView1[7, i].Value.ToString());

            station = AddData(station, i, dataGridView1.Rows.Count);

        }

    }

```

```

} //Refill all empty data

```

```

private void DeveloperToolStripMenuItem_Click(object sender,
EventArgs e)

```

```

{

    MessageBox.Show("Verko Roman, 2019\n HSE CS 183");

} // Developer info in menu strip

```

```

private void dataGridView1_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)

```

```

{

    ChangeElementToolStripMenuItem_Click(sender, e);

} // Changing data in row by Double clicking

```

```

private void ChangeElementToolStripMenuItem_Click(object sender,
EventArgs e)

```

```

{

    try

```

```

    {
        Station newst = new Station();
        int CP = dataGridView1.SelectedRows[0].Index;
        int id, flow, mt;
        int.TryParse(dataGridView1[1, CP].Value.ToString(), out id);
        int.TryParse(dataGridView1[5, CP].Value.ToString(), out
flow);

        int.TryParse(dataGridView1[6, CP].Value.ToString(), out mt);

        newst = new Station(dataGridView1[0, CP].Value.ToString(),
id, dataGridView1[2, CP].Value.ToString(), dataGridView1[3,
CP].Value.ToString(), new Owner(dataGridView1[4, CP].Value.ToString()),
flow, mt, dataGridView1[7, CP].Value.ToString());

        newst = AddData(newst, CP, dataGridView1.Rows.Count);
    }

    catch (ArgumentOutOfRangeException) { MessageBox.Show("Select
any row", "Warning"); }

    catch (Exception) { MessageBox.Show("Exeption"); }

} // Changing data in row using menu strip

private void ownerStationsToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (stations.Count == 0) return;
    Station[] ownstat = new Station[dataGridView1.Rows.Count];
    int id=0, flow=0, mt=0;

    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        int.TryParse(dataGridView1[1,i].Value.ToString(), out id);
        int.TryParse(dataGridView1[5,i].Value.ToString(), out flow);
        int.TryParse(dataGridView1[6,i].Value.ToString(), out mt);

        ownstat[i] = new
Station(dataGridView1[0,i].Value.ToString(), id,

```

```

dataGridView1[2,i].Value.ToString(), dataGridView1[3,i].Value.ToString(),
new Owner(dataGridView1[4,i].Value.ToString()), flow, mt,
dataGridView1[7,i].Value.ToString());
    }
    for (int j = 0; j < ownstat.Length; j++)
    {
        for (int k = 0; k < ownstat.Length; k++)
        {
            if (ownstat[j].Owner.Name.Equals(ownstat[k].Owner.Name)
&& j!=k) ownstat[j].otherOwners++;
        }
    }
    Array.Sort(ownstat);
    List<Station> ownstatlist = new List<Station>();
    foreach (var item in ownstat)
    {
        ownstatlist.Add(item);
    }
    FilterForm Ff = new FilterForm(ownstatlist);
    Ff.Show();

} //todo fliter by owners

private void button1_Click(object sender, EventArgs e)
{
    int example;

    if(textBox1.Text.Trim()==null ||
textBox1.Text.Trim().Equals("") || !int.TryParse(textBox1.Text, out example))
{ MessageBox.Show("no or wrong data"); return; }

    if (stations.Count == 0) return;

    int.TryParse(textBox1.Text, out example);

```

```
stations.Clear();
int id, flow, mt;

for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    int.TryParse(dataGridView1[1, i].Value.ToString(), out id);
    int.TryParse(dataGridView1[5, i].Value.ToString(), out
flow);

    int.TryParse(dataGridView1[6, i].Value.ToString(), out mt);
    stations.Add(new Station(dataGridView1[0,
i].Value.ToString(), id, dataGridView1[2, i].Value.ToString(),
dataGridView1[3, i].Value.ToString(), new Owner(dataGridView1[4,
i].Value.ToString()), flow, mt, dataGridView1[7, i].Value.ToString()));
}
List<Station> FilteredList = new List<Station>();
foreach (var item in stations)
{
    if (item.Flow == example) FilteredList.Add(item);
}

FilterForm Ff = new FilterForm(FilteredList);
Ff.Show();

} //FilterByFlow

private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text.Trim() == null ||
textBox1.Text.Trim().Equals("")) { MessageBox.Show("no or wrong data");
return; }

    string example = textBox1.Text.Trim();
    if (stations.Count == 0) return;
```



```
stations.Clear();
int id, flow, mt;

for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    int.TryParse(dataGridView1[1, i].Value.ToString(), out id);
    int.TryParse(dataGridView1[5, i].Value.ToString(), out
flow);

    int.TryParse(dataGridView1[6, i].Value.ToString(), out mt);
    stations.Add(new Station(dataGridView1[0,
i].Value.ToString(), id, dataGridView1[2, i].Value.ToString(),
dataGridView1[3, i].Value.ToString(), new Owner(dataGridView1[4,
i].Value.ToString()), flow, mt, dataGridView1[7, i].Value.ToString()));
}
List<Station> FilteredList = new List<Station>();
foreach (var item in stations)
{
    if (item.Road.ToLower().Contains(example.ToLower()))
FilteredList.Add(item);
}

FilterForm Ff = new FilterForm(FilteredList);
Ff.Show();

} //FilterByRoad

private void addElementToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (!IsOpened) return;
    Station newStation = new Station();
    stations.Add(newStation);
}
```

```
stations.Add( AddData(newStation, dataGridView1.Rows.Count + 1,
dataGridView1.Rows.Count));

} //Adding new element

private void SaveToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.Rows.Count == 0) return;

    try
    {
        // сохраняем текст в файл
        if (File.Exists(filename))
        {
            File.WriteAllText(filename,
DataGridInString(dataGridView1),Encoding.Default);
        }
        else
        {
            MessageBox.Show("This file no longer exists");
        }

        MessageBox.Show("Saved!");
    }catch(Exception ex) { MessageBox.Show(ex.Message); }

} // Save in current file

private void addToToolStripMenuItem_Click(object sender, EventArgs
e)
{
    if (dataGridView1.Rows.Count == 0) return;
    if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
```

```
        return;

        // получаем выбранный файл
        string savefilename = saveFileDialog1.FileName;
        // сохраняем текст в файл
        saveFileDialog1.CheckPathExists = true;
        saveFileDialog1.CheckFileExists = false;
        File.AppendAllText(savefilename,
DataGridInString(dataGridView1),Encoding.Default);
        MessageBox.Show("Added!");
        saveFileDialog1.CheckFileExists = true;
    } // Add elements to another file

private void Form1_FormClosing(object sender, FormClosingEventArgs
e)
{
    if (IsOpened)
    {
        DialogResult dialogResult = MessageBox.Show("Do you want to
save this file?\nAll your changes can be lost.", "Save",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dialogResult == DialogResult.Yes)
        {
            SaveToolStripMenuItem_Click(sender, e);
        }
        IsOpened = false;
    }
} // Ask to save file before quitting

private void SaveAsToolStripMenuItem_Click(object sender, EventArgs
e)
{
    if (dataGridView1.Rows.Count == 0) return;
```

```
if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)
    return;
// получаем выбранный файл
string savefilename = saveFileDialog1.FileName;
// сохраняем текст в файл
if (File.Exists(savefilename))
{
    filename = savefilename;
    SaveToolStripMenuItem_Click(sender, e);
}
else
{
    File.WriteAllText(savefilename,
DataGridInString(dataGridView1), Encoding.Default);
    MessageBox.Show("Saved!");
}

} // Change directory of saving file

private void CreateNewToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (IsOpened)
    {
        DialogResult dialogResult = MessageBox.Show("Do you want to
save this file?\nAll your changes can be lost.", "Save",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dialogResult == DialogResult.Yes)
        {
            SaveToolStripMenuItem_Click(sender, e);
        }
        IsOpened = false;
    }
}
```

```
}
dataGridView1.Rows.Clear();
dataGridView1.Columns.Clear();
stations.Clear();
GridViewCorrection();
string newdir = Directory.GetCurrentDirectory();
File.WriteAllText(newdir + "new.csv",
DataGridView1.ToString(), Encoding.Default);
filename = newdir + "new.csv";
MessageBox.Show("Created new.csv");

IsOpened = true;
} // New file

private void GridViewCorrection()
{
    dataGridView1.Columns.Add("column 1", "Name");
    dataGridView1.Columns.Add("column 2", "ID");
    dataGridView1.Columns.Add("column 3", "Registration date");
    dataGridView1.Columns.Add("column 4", "Location");
    dataGridView1.Columns.Add("column 5", "Owner");
    dataGridView1.Columns.Add("column 6", "Flow");
    dataGridView1.Columns.Add("column 7", "Maintenance time");
    dataGridView1.Columns.Add("column 8", "Road");

    dataGridView1.Columns[1].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    dataGridView1.Columns[2].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    dataGridView1.Columns[5].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    dataGridView1.Columns[6].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
```

```

        dataGridView1.Columns[6].Width *= 2;
        dataGridView1.Columns[2].Width *= 2;
        dataGridView1.Columns[0].Width *= 2;
        dataGridView1.Columns[3].Width *= 2;
        dataGridView1.Columns[4].Width *= 2;
        dataGridView1.Columns[7].Width *= 2;

        dataGridView1.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

        Font F = new Font("Cambria", 12, FontStyle.Bold);
        dataGridView1.ColumnHeadersDefaultCellStyle.Font = F;

        dataGridView1.SelectionMode =
DataGridViewSelectionMode.FullRowSelect;
    } //Making new table in DataGridView1
    /// <summary>
    /// Adding data to the element in grid
    /// </summary>
    /// <param name="station">element</param>
    /// <param name="n">number of element</param>
    /// <param name="m">count of all elements</param>
    /// <returns></returns>
    public Station AddData(Station station, int n, int m)
    {
        AddDataForm form3 = new AddDataForm(station,n,m);
        form3.Show();
        return form3.newstation;
    } // Opening Form for adding new element
    /// <summary>
    /// String representation of all data (CSV)
    /// </summary>
    /// <param name="dtg">datagridview</param>
    /// <returns></returns>

```

```

private string DataGridInString(DataGridView dtg)
{
    string text = "";
    // text +=
    "Name;ID;RegistrationDate;Location;Owner;Flow;MaintenanceTime;Road\n";
    for (int i = 0; i < dtg.Rows.Count; i++)
    {
        text += $"{dtg[0,i].Value.ToString()};{dtg[1,
i].Value.ToString()};{dtg[2, i].Value.ToString()};{dtg[3,
i].Value.ToString()};{dtg[4, i].Value.ToString()};{dtg[5,
i].Value.ToString()};{dtg[6, i].Value.ToString()};{dtg[7,
i].Value.ToString()}\n";
    }
    return text;
} // Turn data in csv text file
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace ClassLibrary1
{
    public class Station: IComparable<Station>
    {
        public string Name;
        public int ID;
        public string RegistrationDate;
        public string Location;
        public Owner Owner;
        public int Flow;
        public int MainTenanceTime;
        public string Road;
        public int otherOwners;
        public Station(string name,int iD, string registration, string
location, Owner own, int flow, int maint, string road)
        {
            own.Name = CorrectWord(own.Name);

```

```
        name = CorrectWord(name);
        road = CorrectWord(road);

        Name = name;
        ID = iD;
        RegistrationDate = registration;
        Location = location;
        Owner = own;
        Flow = flow;
        MainTenanceTime = maint;
        Road = road;

    }
    public Station()
    {
        Owner = new Owner("Owner's name");
    }
    string CorrectWord(string old)
    {
        old = old.Replace("'", ' ');
        old = old.Replace(" ", " ");
        old = old.Trim();
        return old;
    }

    public int CompareTo(Station stat)
    {
        if (stat.otherOwners > otherOwners) return 1;
        else if (stat.otherOwners == otherOwners) return 0;
        else return -1;
    }
}

public class Owner
{
    public string Name;
    public Owner(string a)
    {
        Name = a;
    }
}
}
```



## 8. Список литературы

1. Герберт Шилдт: С# 4.0. Полное руководство // С# 4.0: The Complete Reference, «Вильямс» 2015.
2. Руководство по программированию на С# - MSDN – Microsoft [Электронный ресурс]: русский ресурс, содержащий информацию по документации .NET Framework 4.7.2 , URL: <https://docs.microsoft.com/ru-ru/dotnet/api/4.7.2> (дата обращения: 01.12.2018).
3. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ Р 7.0.5—2008. ЕСПД. БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления / Издание официальное, Москва, Стандартинформ, 2008.