МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ автономное ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

(ФГБОУ ВПО МПУ)



Кафедра СМАРТ-технологии

Практическая работа № 5

«Исследование возможностей работы с файлами и файловой системой на языке C#»

По дисциплине «Программирование и алгоритмизация на языках высокого уровня»

Группа \_\_\_\_241-324 \_\_\_\_\_\_

№ группы

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сальников Л.В.

Подпись студента

Дата \_\_\_27.05.2025\_\_\_\_

Дата сдачи

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крыжановская Т.Г.

Подпись преподавателя

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Клецкин А.А.

Подпись преподавателя

2025

Цель:

* Создание приложения для работы с файловой системой;
* Создать приложение для чтения и записи текстового файла.

Выполнение задания:

**Этап 1: Создание проекта**

* Создал новый проект типа **Windows Forms App (.NET 8.0)**.
* Назвал, CsvEditor.

### Этап 2: Дизайн интерфейса

Разместил на форме следующие элементы:

|  |  |
| --- | --- |
| **Элемент управления** | Назначение |
| DataGridView | Основная таблица для отображения и редактирования CSV |
| Button "Открыть" | Чтение CSV-файла и загрузка в таблицу |
| Button "Сохранить" | |  | | --- | |  |  |  | | --- | | Сохранение содержимого таблицы в файл | |
| Button "Новый" | Отчищает поле для работы с новым файлом |

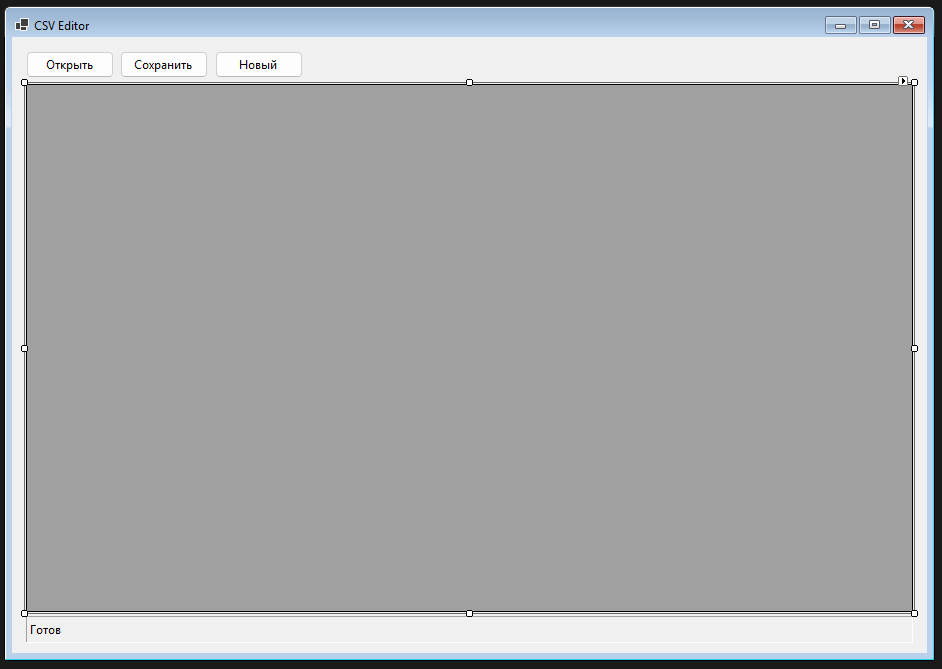


Рисунок 1

**Этап 3: Загрузка CSV в DataGridView**

1. При нажатии на кнопку "Открыть":
   * Открыть диалог выбора файла (OpenFileDialog) с фильтром \*.csv.
   * Прочитать строки из файла.
   * Разбить каждую строку на ячейки с помощью String.Split(',').
   * Добавить строки и столбцы в DataGridView.

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

{

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.Filter = "CSV files (\*.csv)|\*.csv|All files (\*.\*)|\*.\*";

openFileDialog.FilterIndex = 1;

openFileDialog.RestoreDirectory = true;

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

// Чтение всех строк из файла

string[] lines = File.ReadAllLines(openFileDialog.FileName);

if (lines.Length == 0) return;

// Очистка DataGridView

dataGridView.Rows.Clear();

dataGridView.Columns.Clear();

// Разделение первой строки для заголовков столбцов

string[] headers = lines[0].Split(',');

foreach (string header in headers)

{

dataGridView.Columns.Add(header, header);

}

// Добавление строк данных

for (int i = 1; i < lines.Length; i++)

{

string[] rowData = lines[i].Split(',');

dataGridView.Rows.Add(rowData);

}

lblStatus.Text = $"Файл загружен: {openFileDialog.FileName}";

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при открытии файла: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**Этап 4: Сохранение CSV из DataGridView**

1. При нажатии на кнопку "Сохранить CSV":
   * Пройти по строкам и ячейкам DataGridView.
   * Собрать значения строк, разделяя ячейки запятой.
   * Сохранить их в файл с помощью File.WriteAllLines.

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

{

if (dataGridView.Rows.Count == 0)

{

MessageBox.Show("Нет данных для сохранения", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

using (SaveFileDialog saveFileDialog = new SaveFileDialog())

{

saveFileDialog.Filter = "CSV files (\*.csv)|\*.csv|All files (\*.\*)|\*.\*";

saveFileDialog.FilterIndex = 1;

saveFileDialog.RestoreDirectory = true;

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

using (StreamWriter writer = new StreamWriter(saveFileDialog.FileName))

{

// Запись заголовков

var headers = dataGridView.Columns.Cast<DataGridViewColumn>()

.Select(column => column.HeaderText);

writer.WriteLine(string.Join(",", headers));

// Запись данных

foreach (DataGridViewRow row in dataGridView.Rows)

{

if (!row.IsNewRow)

{

var cells = row.Cells.Cast<DataGridViewCell>()

.Select(cell => cell.Value?.ToString() ?? "");

writer.WriteLine(string.Join(",", cells));

}

}

}

lblStatus.Text = $"Файл сохранен: {saveFileDialog.FileName}";

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при сохранении файла: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**Этап 5: Обработка ошибок**

* Обернул чтение и запись в try-catch.
* Выводятся сообщения об ошибках через MessageBox.

**Этап 6: Тестирование**

* Проверил работу с простыми и сложными CSV (в том числе с пустыми ячейками).
* Убедился, что данные корректно отображаются и сохраняются.

Ключевые моменты:

**Загрузка данных:**

* Открывается диалог выбора файла (OpenFileDialog).
* Считываются строки с помощью File.ReadAllLines.
* Первая строка обрабатывается через String.Split(',') для получения заголовков.
* Остальные строки обрабатываются аналогично и добавляются в таблицу.

**Сохранение данных:**

* Открывается диалог сохранения (SaveFileDialog).
* Заголовки и строки таблицы записываются в файл через StreamWriter.
* Используется string.Join(",", ...) для формирования строк.

Вывод:

В ходе выполнения задания разработано приложение, демонстрирующее базовые навыки работы с файловой системой, табличными данными и графическим интерфейсом пользователя в среде Windows Forms. Использование формата CSV делает приложение универсальным для обмена табличными данными.