Задача для соискателя на должность C# разработчика

# Описание

Необходимо разработать приложение, позволяющее загружать текстовые файлы с данными и отображать их в графическом интерфейсе программы.

# Входные данные

Входными данными является текстовый файл. В этом файле содержится информация о связях между двумя произвольными идентификаторами (в каждой строке одна связь). Строка содержит поля с информацией о связываемых идентификаторах и различные характеристики связи. Поля (столбцы) разделены символом-разделителем. Набор полей фиксированный и их порядок не изменяется. Первая строка содержит заголовки данных. Примером такого файла является обычный .csv файл. Файл может быть большого размера.

Пример входного файла:

Date; Object A; Type A; Object B; Type B; Direction; Color; Intensity; LatitudeA; LongitudeA; LatitudeB; LongitudeB; 01.01.2018;B;Type1;C;Type2;in;#00008B;5;60.023898;30.406769;60.015158;30.391641;

01.01.2018;A;Type1;B;Type1;out;red;2;60.012746;30.394991;60.018848;30.416448;

02.01.2018;A;Type1;B;Type1;out;black;3;60.018848;30.416448;60.023898;30.406769;

…

# Основные требования

Необходимо разработать приложение, позволяющее разбирать файлы с форматом, описанным выше и отображать пользователю данные в табличном виде. При отображении данных программа должна учитывать типы полей и позволять осуществлять сортировку в соответствии с форматом полей.

В приложении должен быть реализован пользовательский интерфейс для загрузки данных из выбранного файла. В процессе загрузки данных приложение должно сохранять способность обрабатывать команды пользователя (не должно «подвисать»). При загрузке данных должен отображаться прогресс, и кнопка отмены. При нажатии на кнопку отмены необходимо прерывать процесс загрузки.

Программа должна быть написана на языке C# (Visual Studio 2015 и выше) с помощью технологии

1. et WPF (4.5 и выше) и должна следовать принципам ООП и MVVM.

# Дополнительные требования

Дополнительные требования не являются обязательными. Но выполнение этих требований будет учтено при выборе кандидата.

* + Использование компонент DevExpress (например, GridControl). Более подробно можно ознакомится по ссылке [https://documentation.devexpress.com/WPF/6084/Controls-and-](https://documentation.devexpress.com/WPF/6084/Controls-and-Libraries/Data-Grid) [Libraries/Data-Grid.](https://documentation.devexpress.com/WPF/6084/Controls-and-Libraries/Data-Grid)
  + Визуализация загруженных данных в любых других компонентах (схема связей, графики, карта и пр.)
  + Загрузка файлов с измененной последовательностью полей (столбцов).
  + Фильтрация данных в пользовательском интерфейсе.
  + Загрузка исходных данных в БД произвольной структуры по команде пользователя. Экспорт из базы данных.