**Практическое занятие № 12. Методы работы с файлами. Пространство имен System.IO**

**1 Цель занятия**

Получить практические навыки использования свойств файлов и методов работы с файлами

**2 Перечень оборудования и программного обеспечения**

Персональный компьютер

Microsoft Office (Word)

Microsoft Visual Studio

**3 Краткие теоретические сведения**

Класс FileInfo пространства имен System.IO предоставляет статические методы для создания, копирования, удаления, перемещения и открытия файлов.

Для того чтобы работать с файлами, необходимо:

* **подключить пространство имен**: **using System.IO;**
* использовать класс FileInfo (файл) со свойствами и методами:

Exists – значение (true, false), показывающее существует ли файл;

Create() – создает файл;

Delete() – удаляет файл;

CreateText() – создает выходной текстовый поток StreamWriter, записывающий текст в файл;

AppendText() – создает выходной текстовый поток StreamWriter, дозаписывающий текст в файл;

* использовать класс StreamWriter (выходной текстовый поток) с методами:

Write() – записывает в выходной поток строку;

WriteLine() – записывает в выходной поток строку с признаком конца строки;

Close() – закрывает выходной поток;

* использовать класс StreamReader (входной текстовый поток) со свойствами и методами:

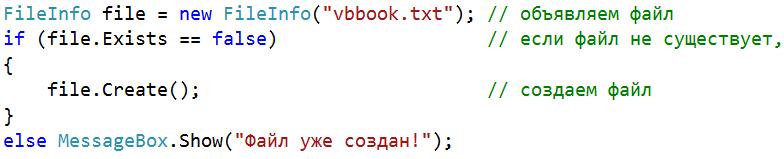
EndOfStream – значение (true, false), определяющее, находится ли текущая позиция входного потока в его конце;

Read() – чтение символа из входного потока и перемещение текущей позиции входного потока на одну позицию вперед;

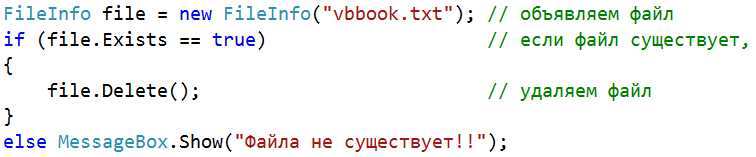
ReadLine() – чтение строки символов из входного потока в текстовую строку и перемещение текущей позиции входного потока к началу следующей строки символов;

Close() – закрывает входной поток.

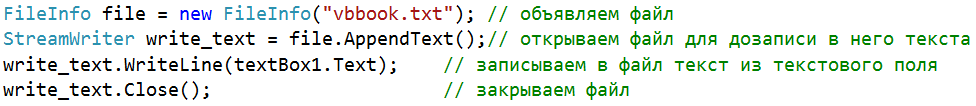
**Пример 1.** Создание файла "vbbook.txt":



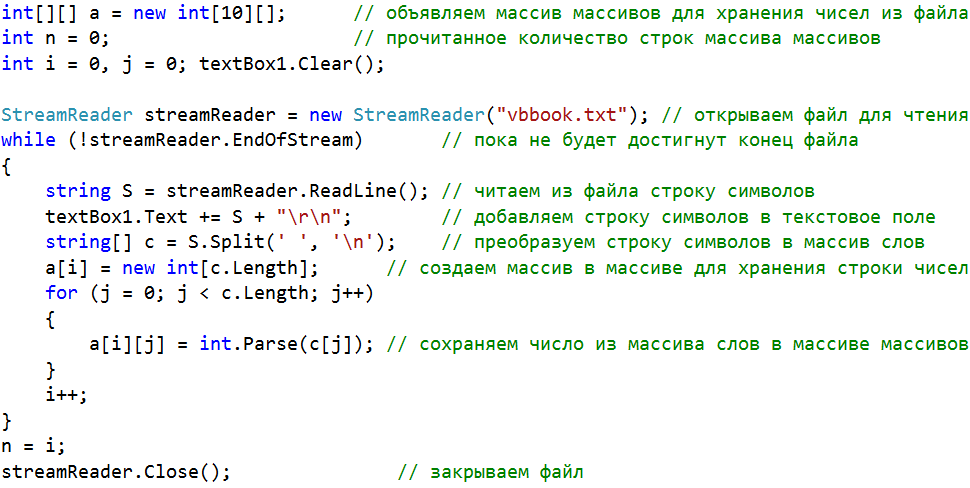
**Пример 2.** Удаление файла "vbbook.txt":



**Пример 3.** Запись текста в файл "vbbook.txt":



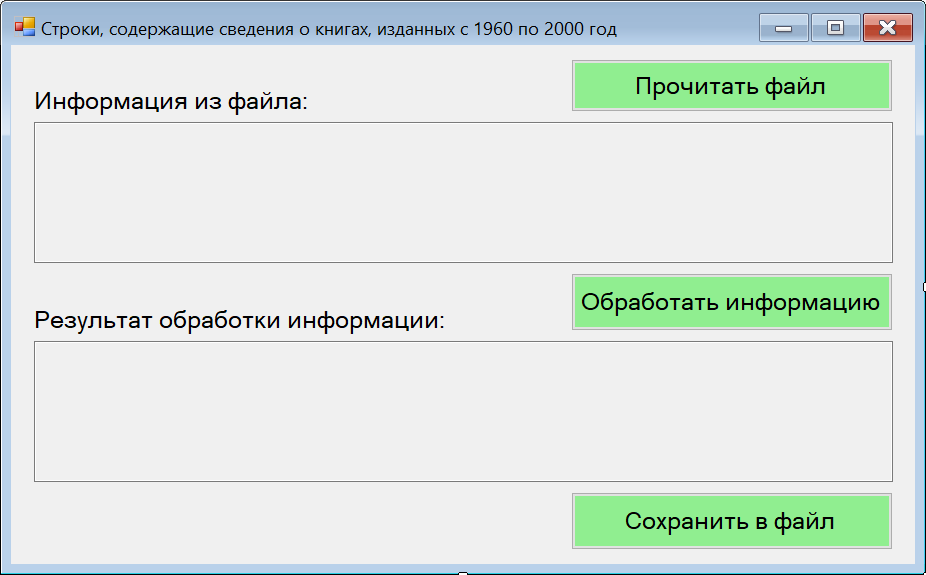
**Пример 4.** Чтение чисел из файла "vbbook.txt" и сохранение их в массиве массивов:



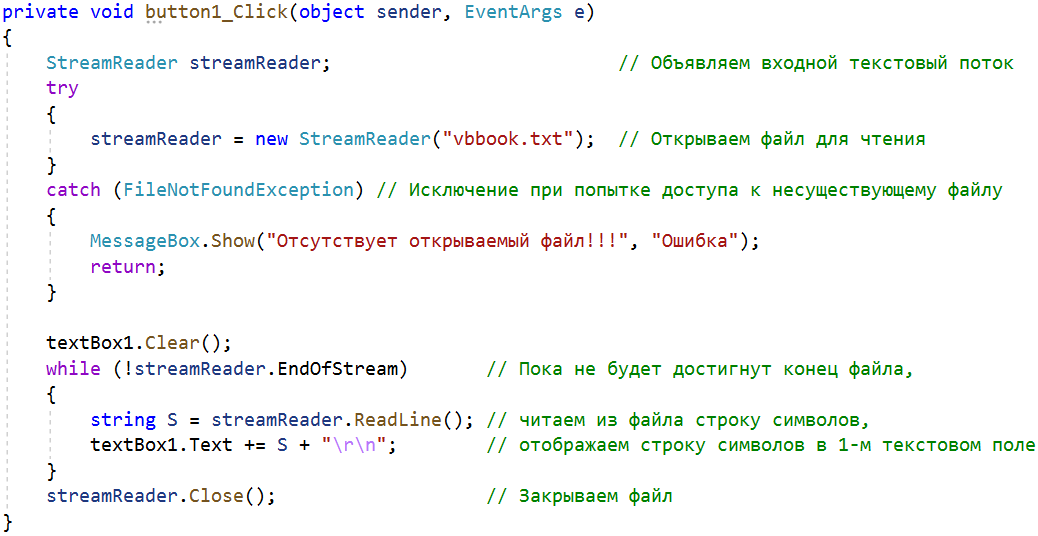
**Задание 1.** Дан текстовый файл. Структура строки текстового файла: фамилия и инициалы автора (20 позиций), название книги (30 позиций), издательство (10 позиций), год издания (4 позиции). Создать новый текстовый файл из строк, содержащих сведения о книгах, изданных за период с 1960 – 2000г.

**Решение.**

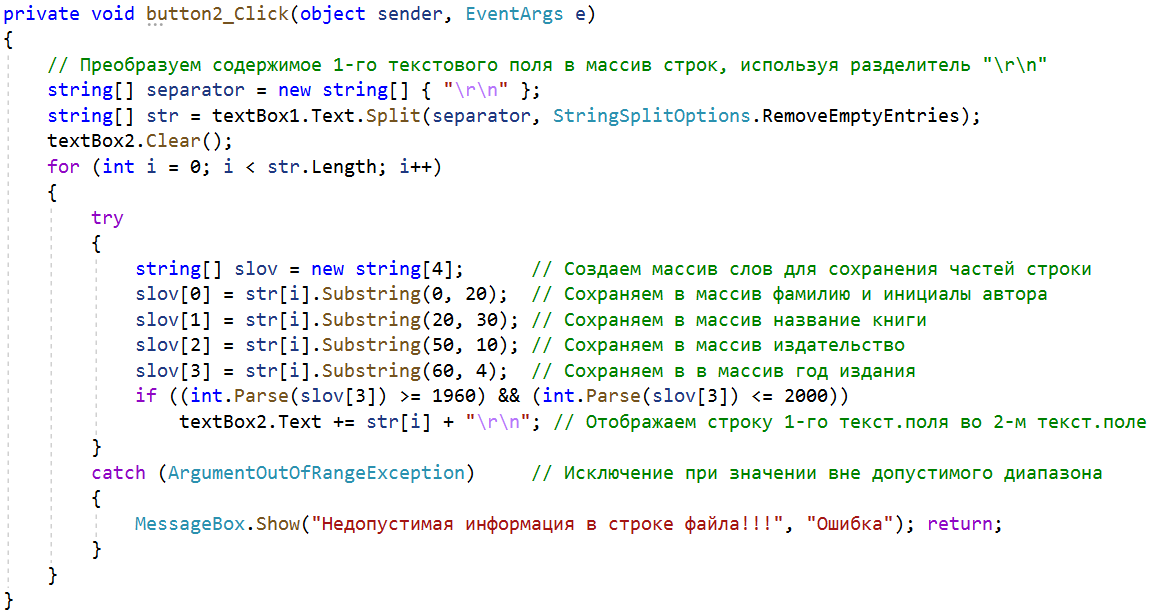
а) Интерфейс приложения Windows Forms:



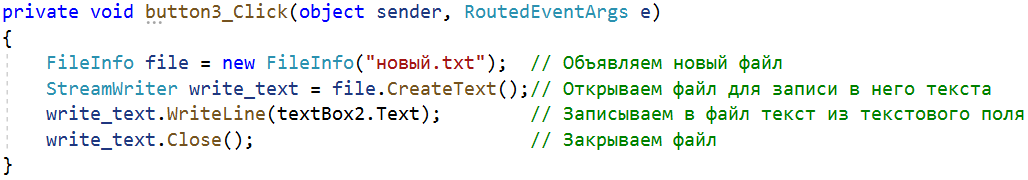
Код, выполняемый при щелчке на кнопку «Прочитать файл»:



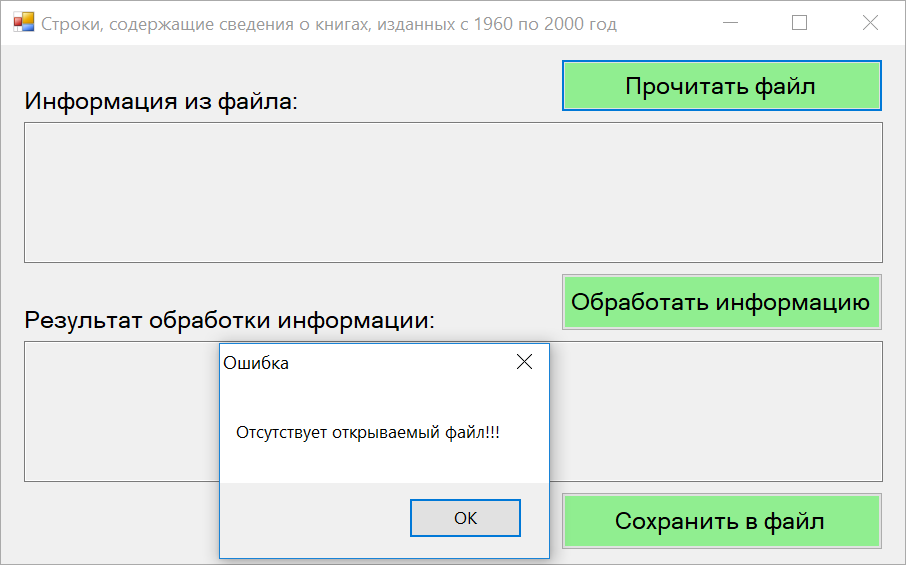
Код, выполняемый при щелчке на кнопку «Обработать информацию»:

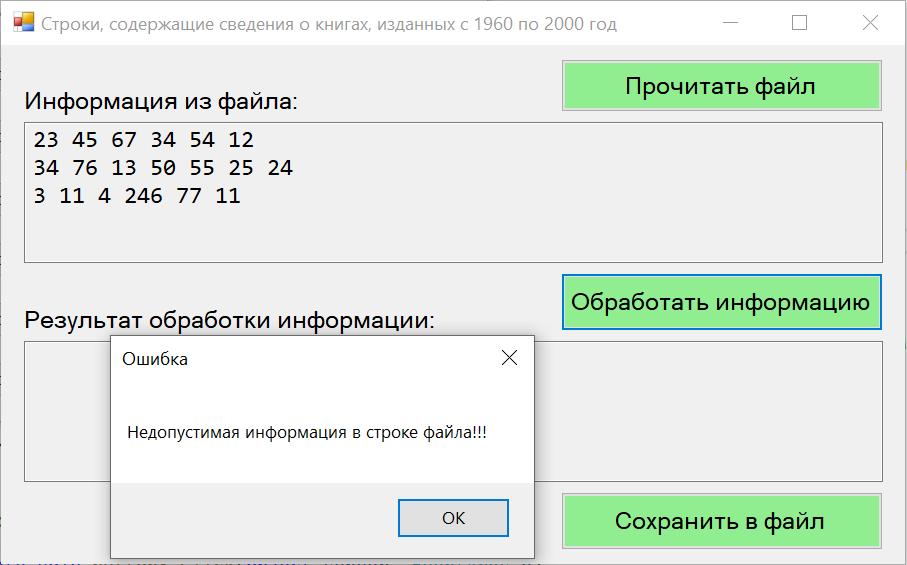


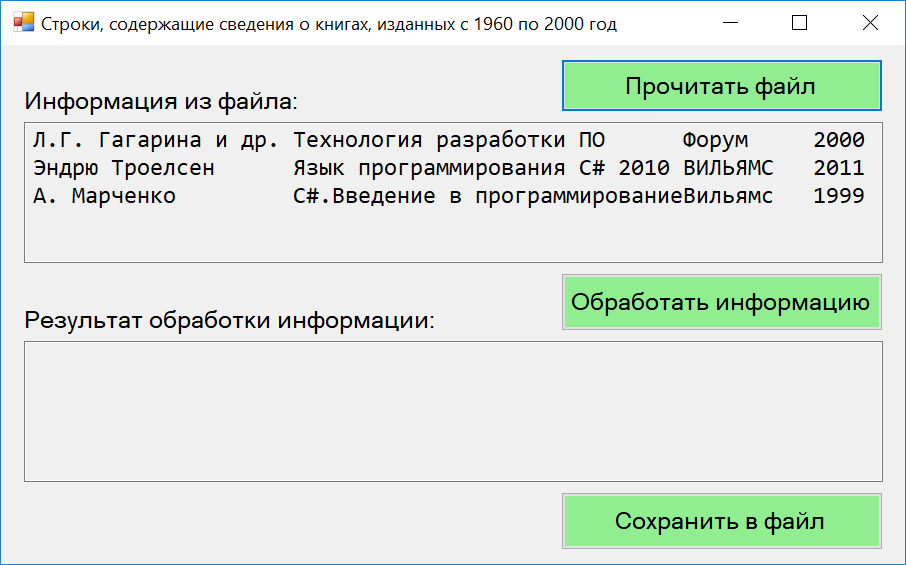
Код, выполняемый при щелчке на кнопку «Сохранить в файл»:

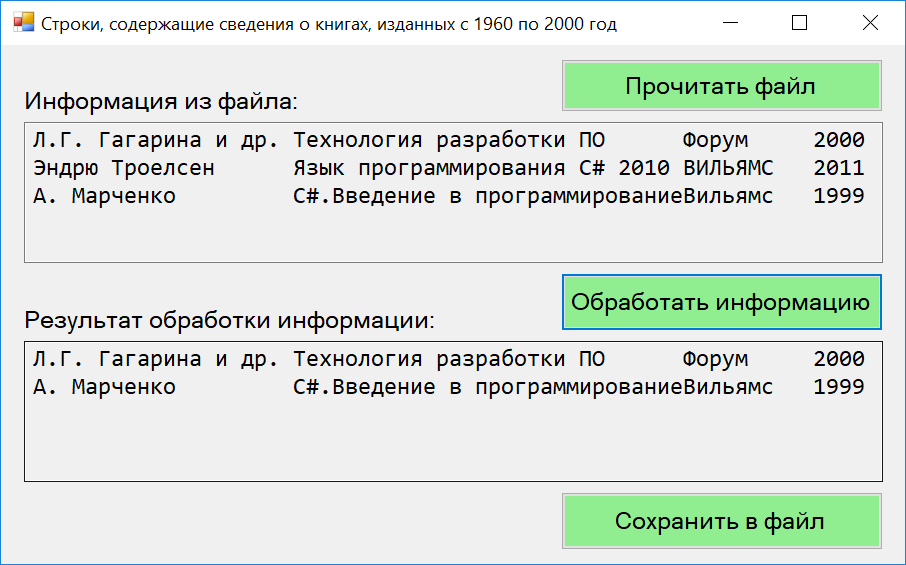


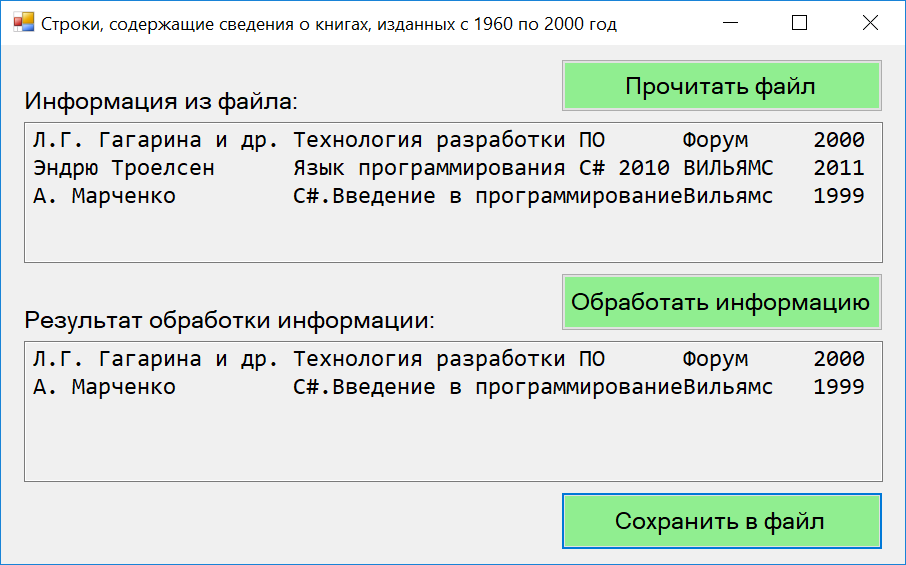
Скрины отлаженной программы:

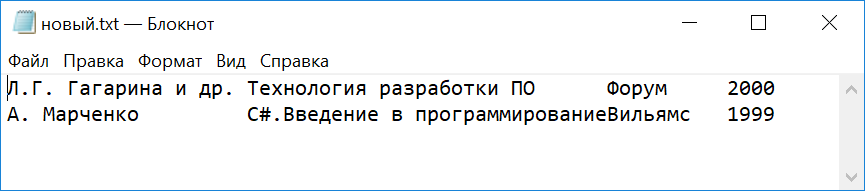




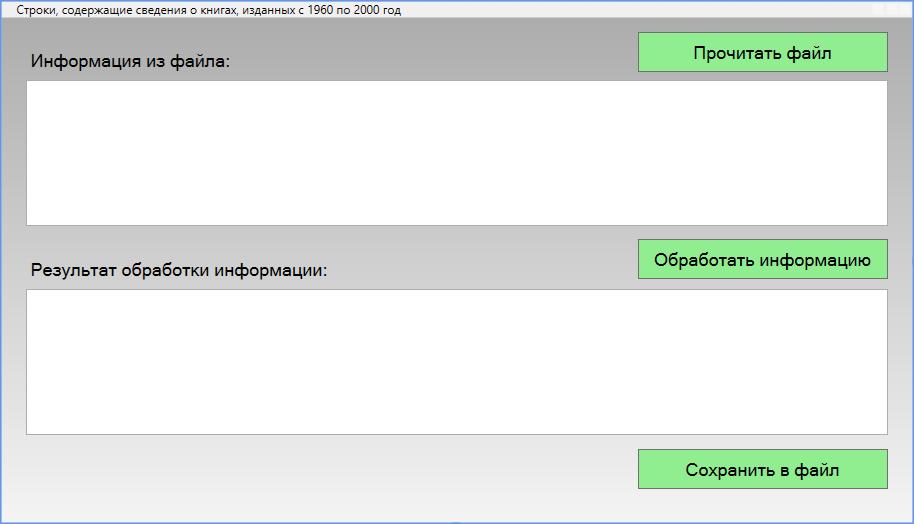




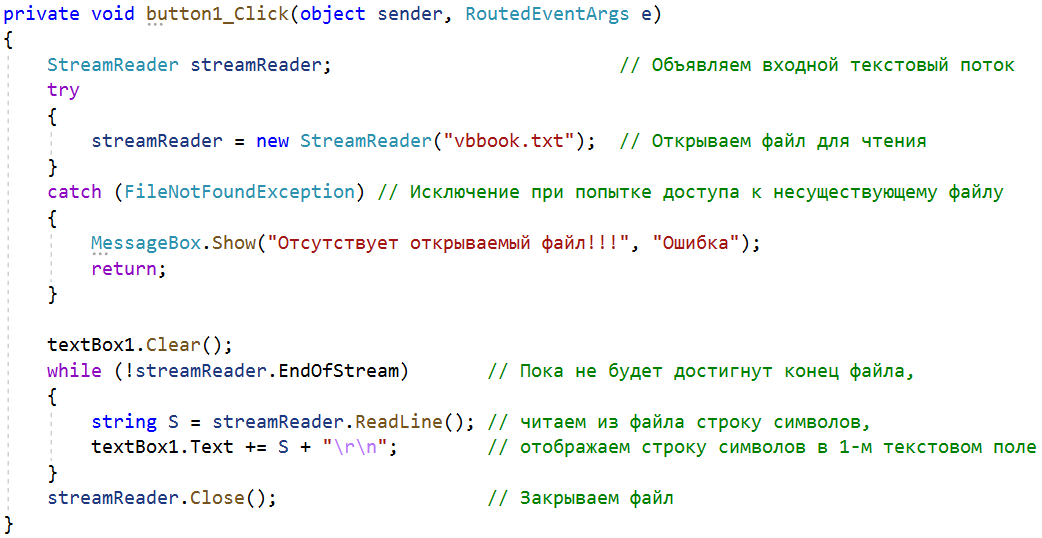




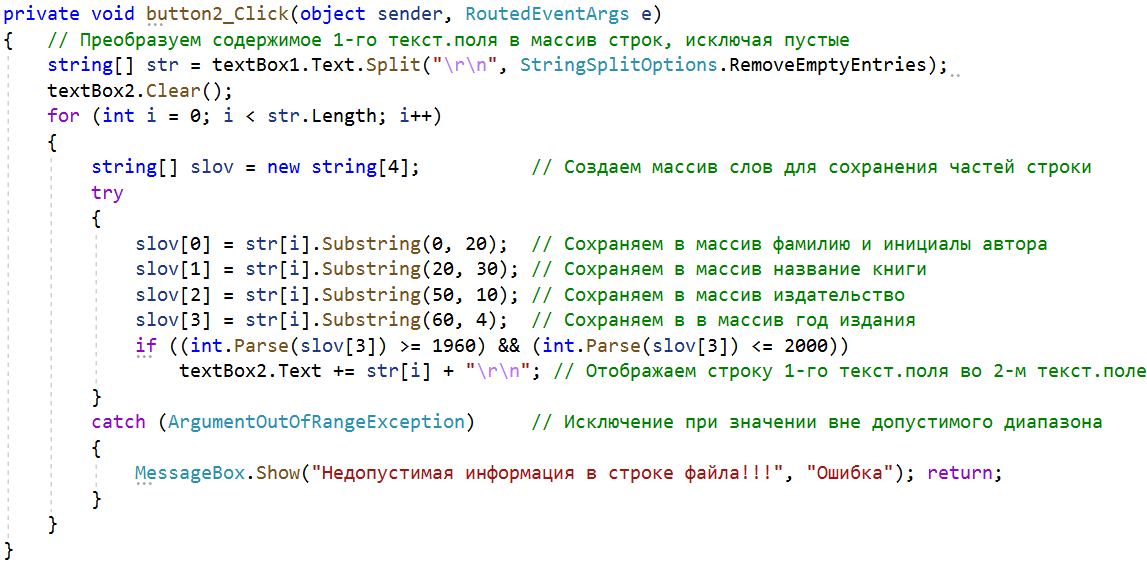
б) Интерфейс приложения WPF:



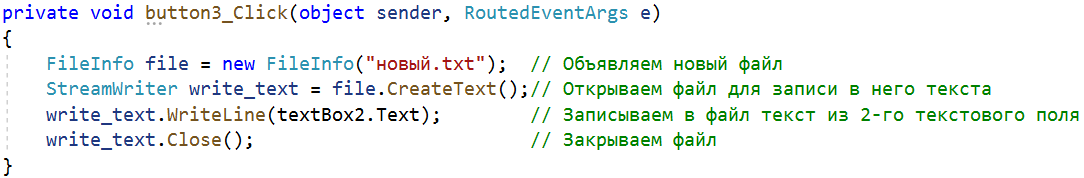
Код, выполняемый при щелчке на кнопку «Прочитать файл»:



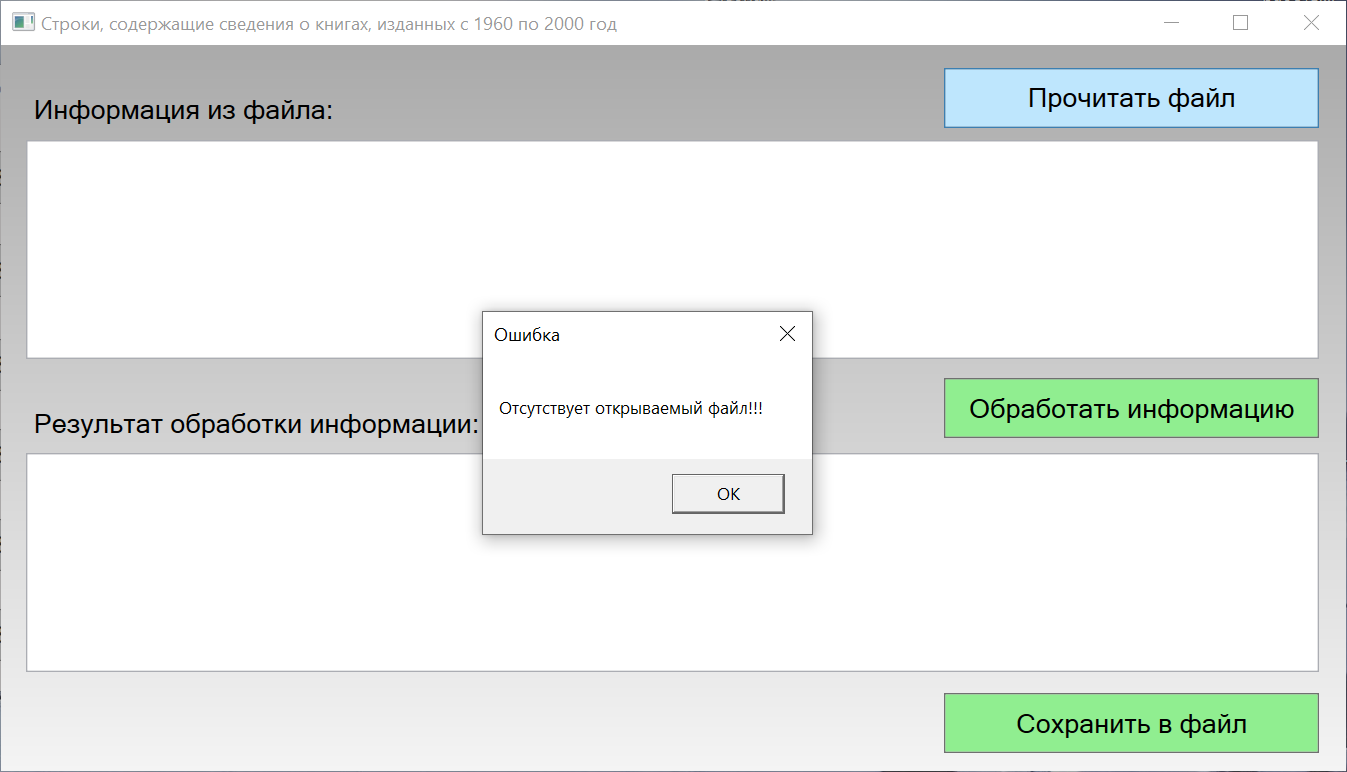
Код, выполняемый при щелчке на кнопку «Обработать информацию»:

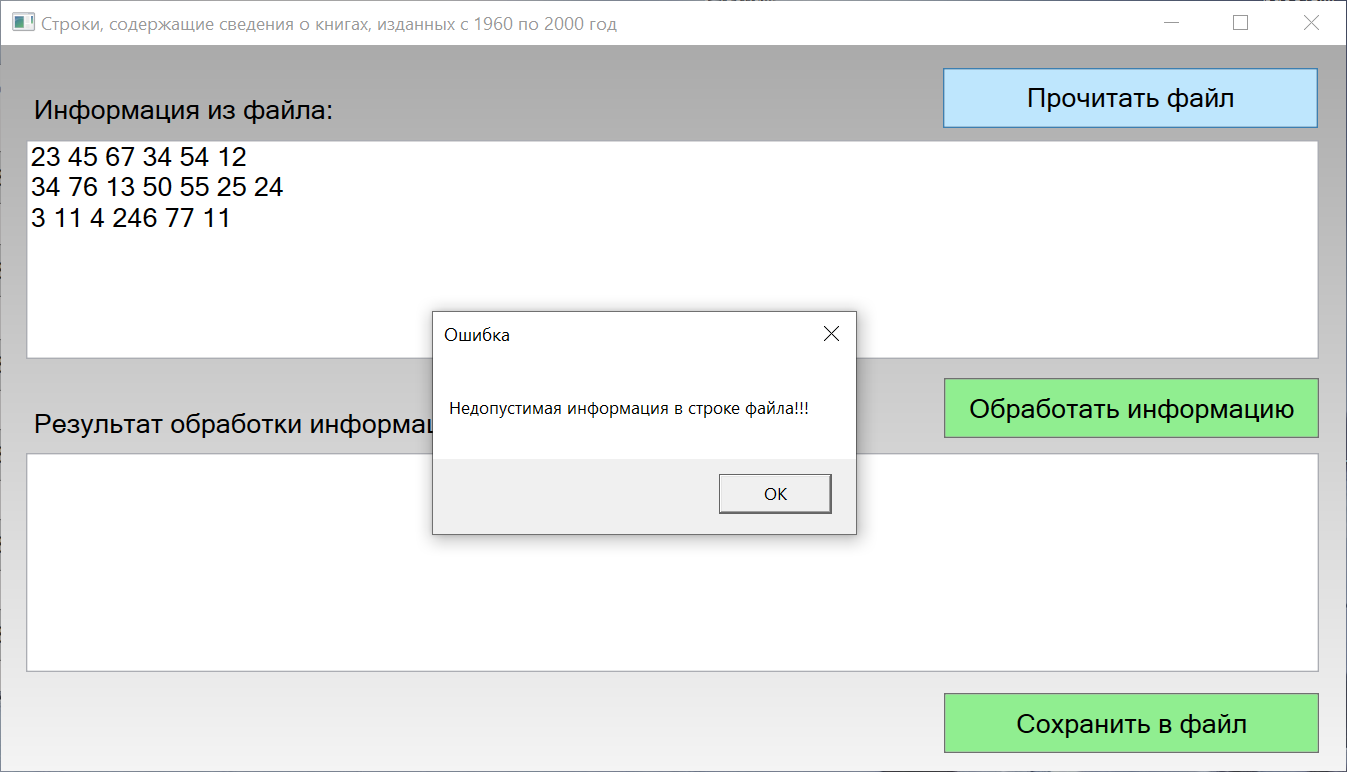


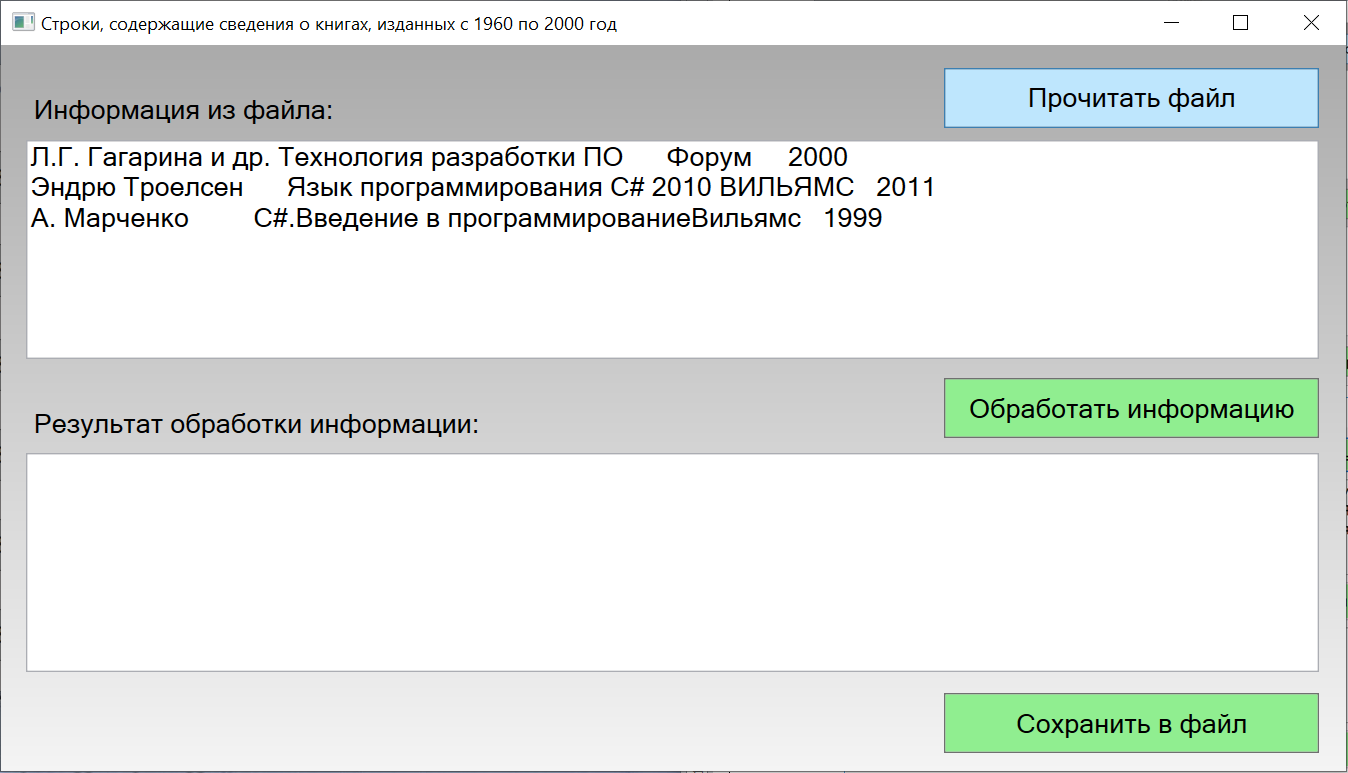
Код, выполняемый при щелчке на кнопку «Сохранить в файл»:

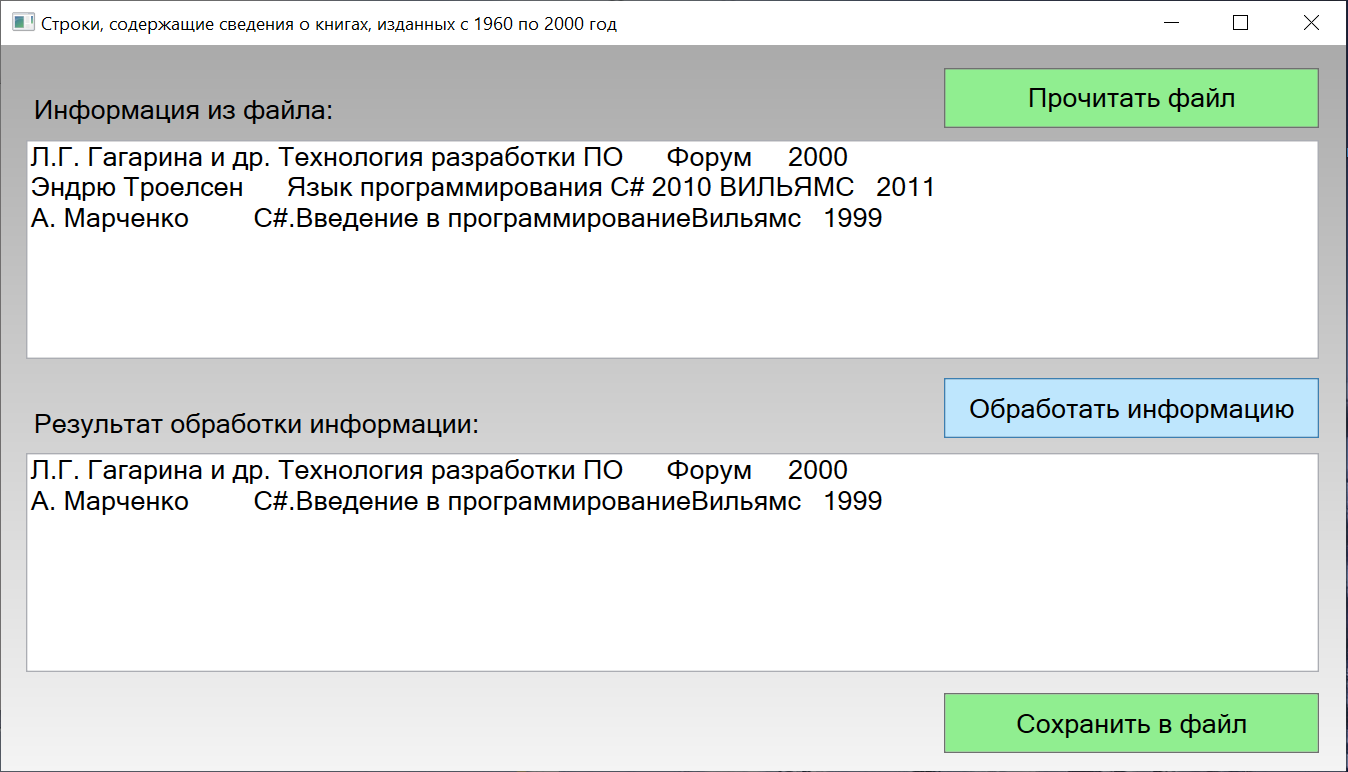


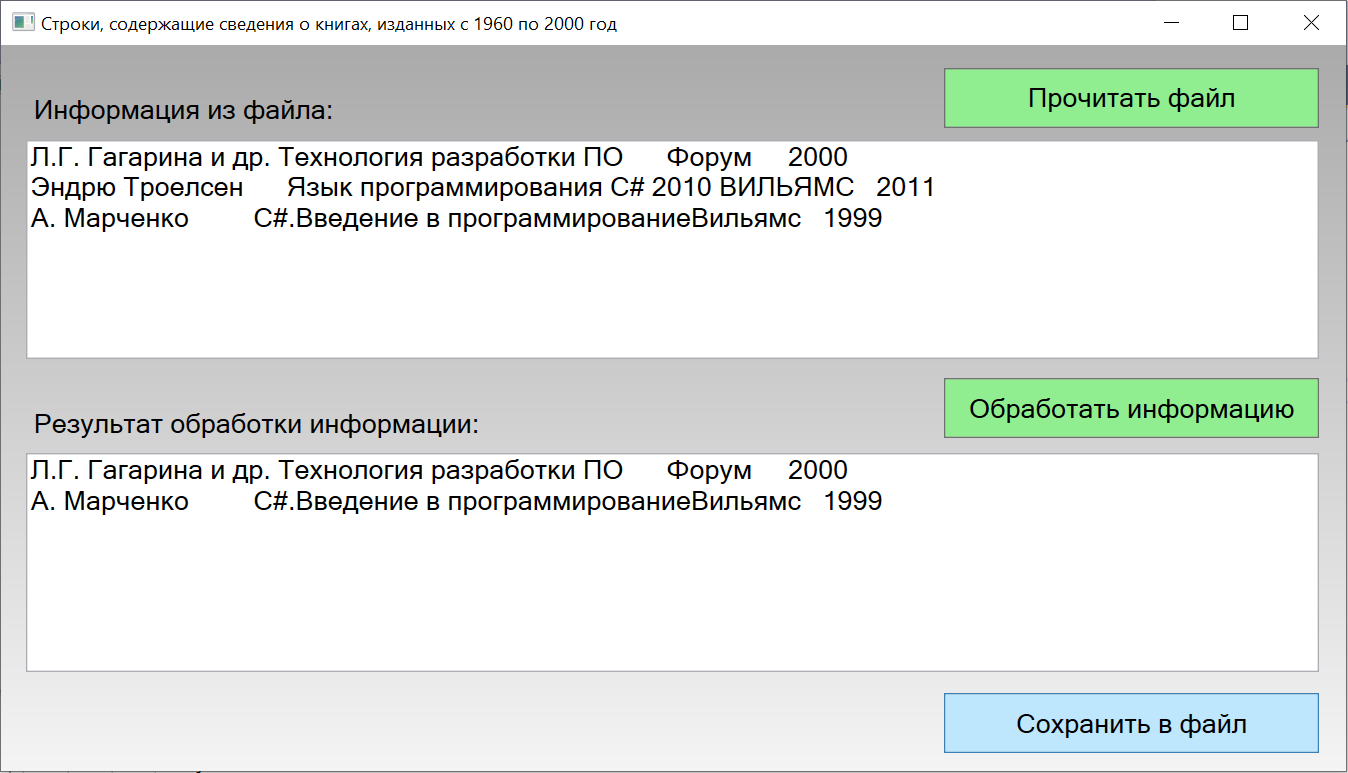
Скрины отлаженной программы:

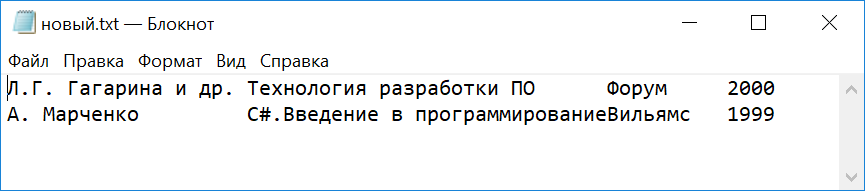




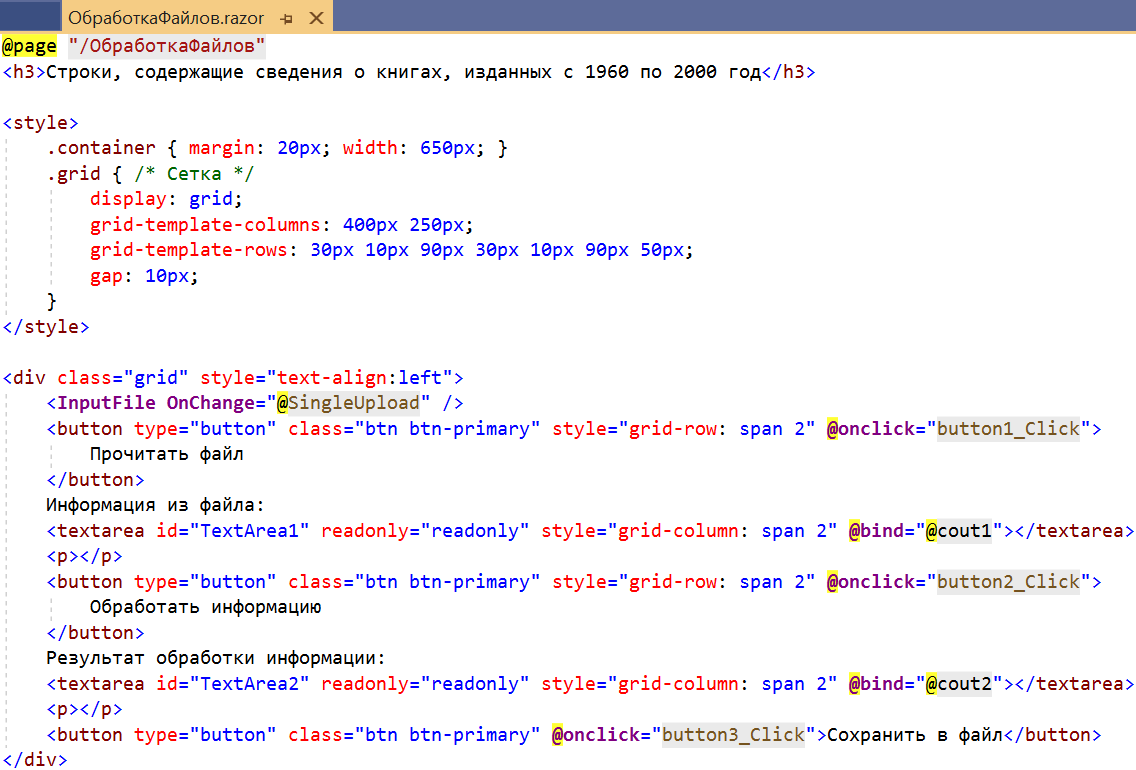


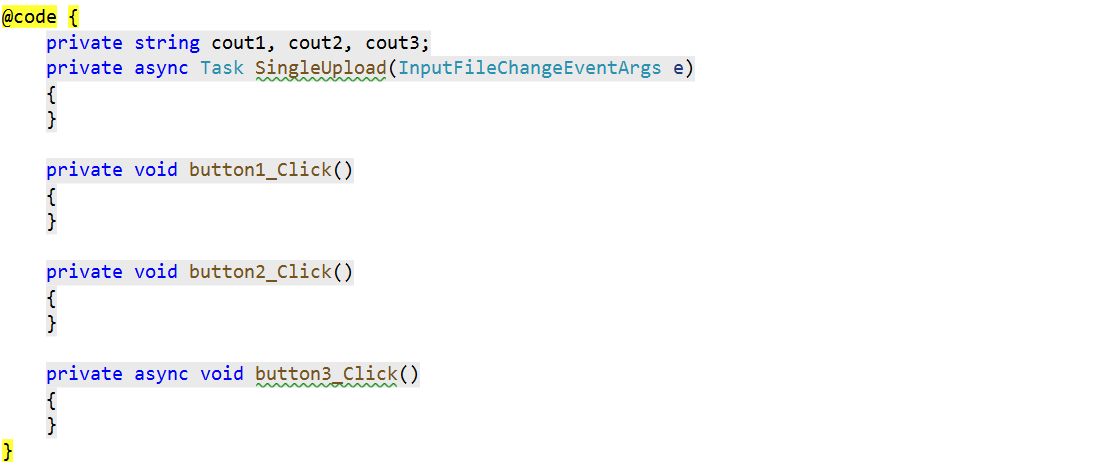


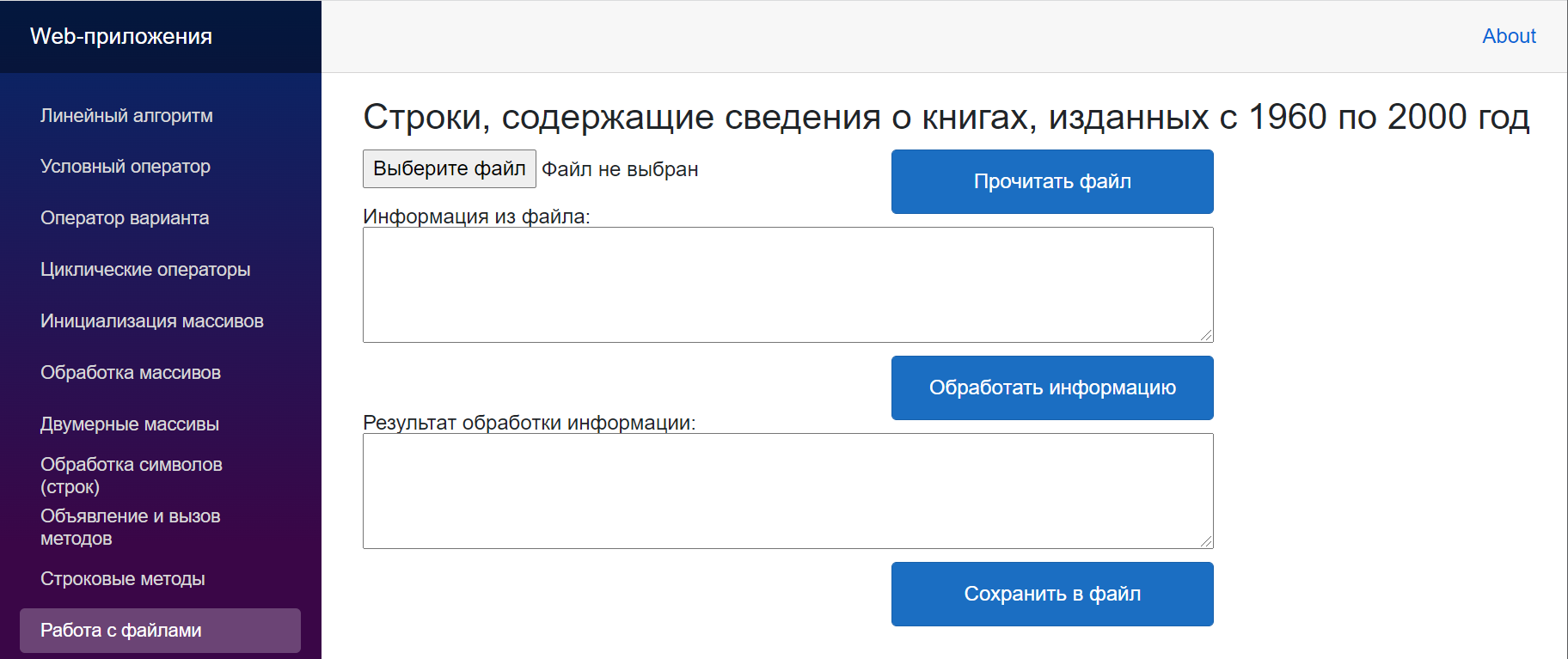




в) Интерфейс приложения WebAssembly Blazor (.NET 5.0):







Коды приложения:

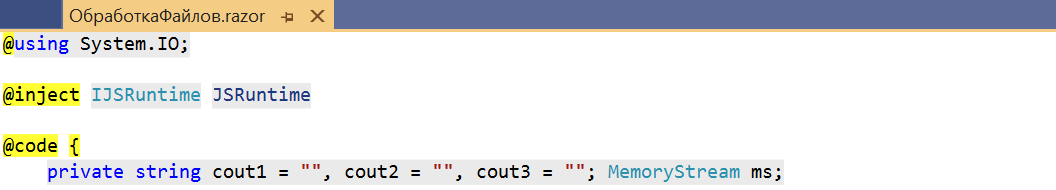
На данный момент в Blazor нет встроенных функций для сохранения файла. Поэтому создадим файл с именем SaveFile.js в папкеwwwroot вашего проекта. Этот файл будет содержать функцию JavaScript, которую будем вызывать из кода C#, и которая принимает два аргумента: имя файла и содержимое, формирующее тело текстового документа.

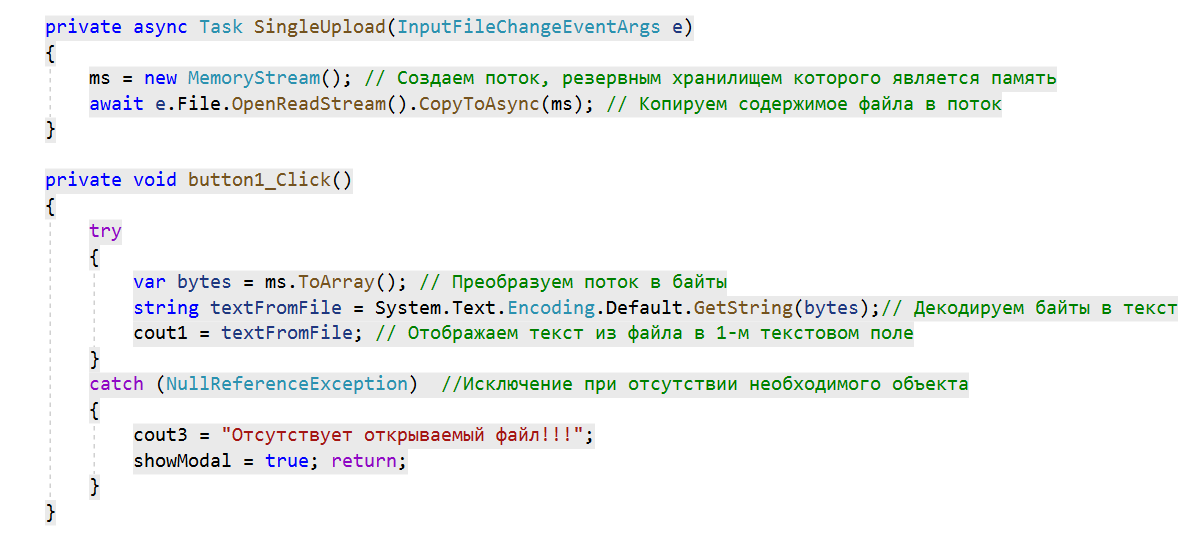


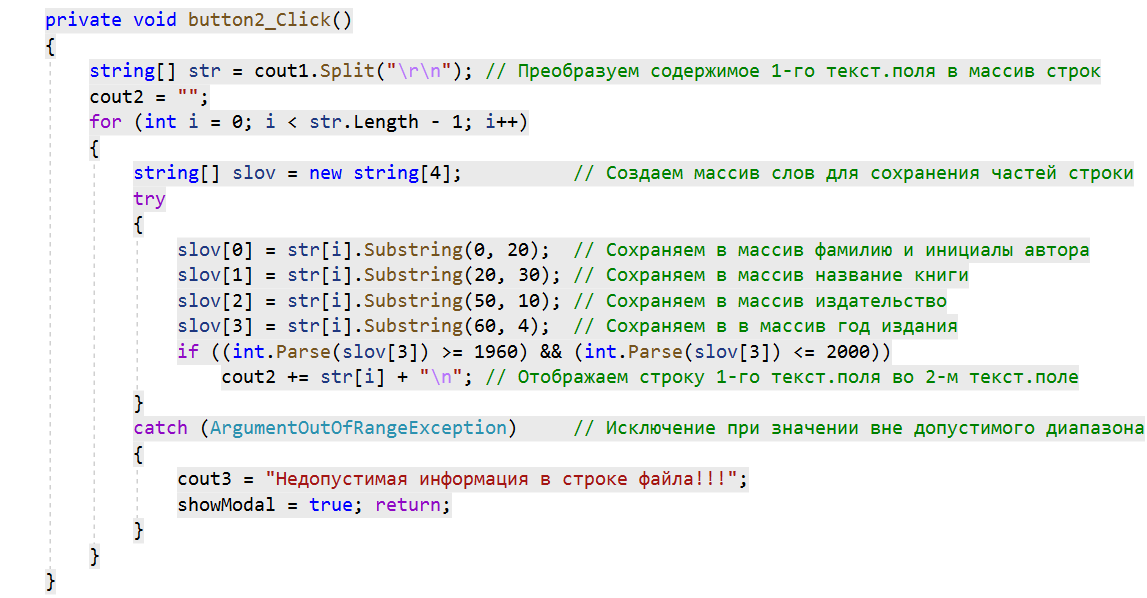
Загрузим скрипт в wwwroot/index.html непосредственно перед закрывающим тегом body </body>.

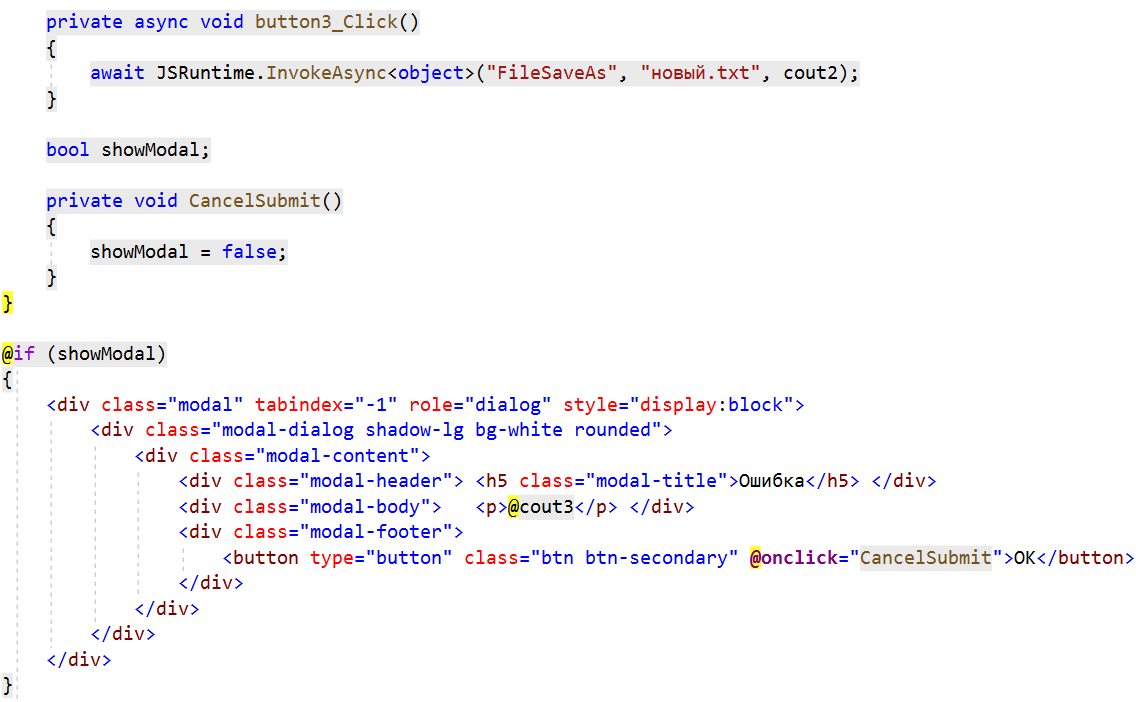


Доступ к файлам в Blazor осуществляется асинхронно, поэтому методы работы с файлами создаются как async-методы. Чтобы использовать код JavaScript в Blazor, необходимо в компонент Razor, из которого выполняется код JavaScript, внедрить экземпляр объекта IJSRuntime.

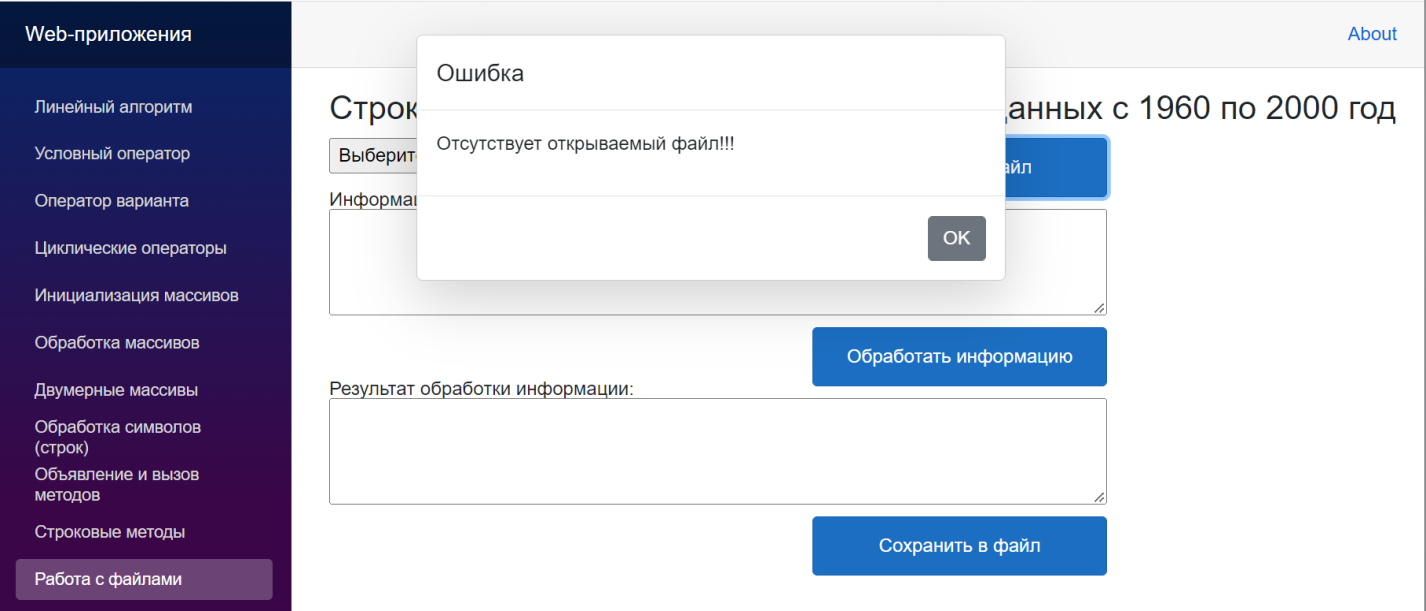


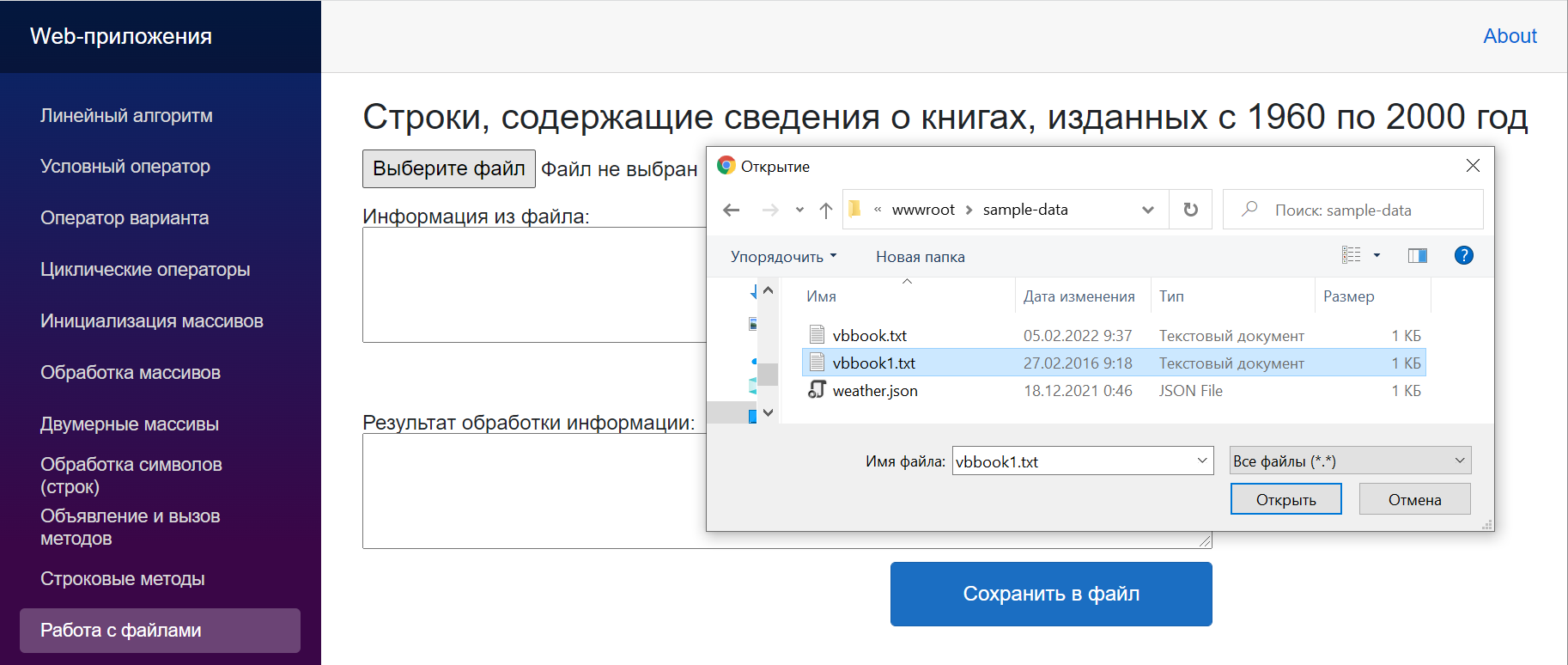


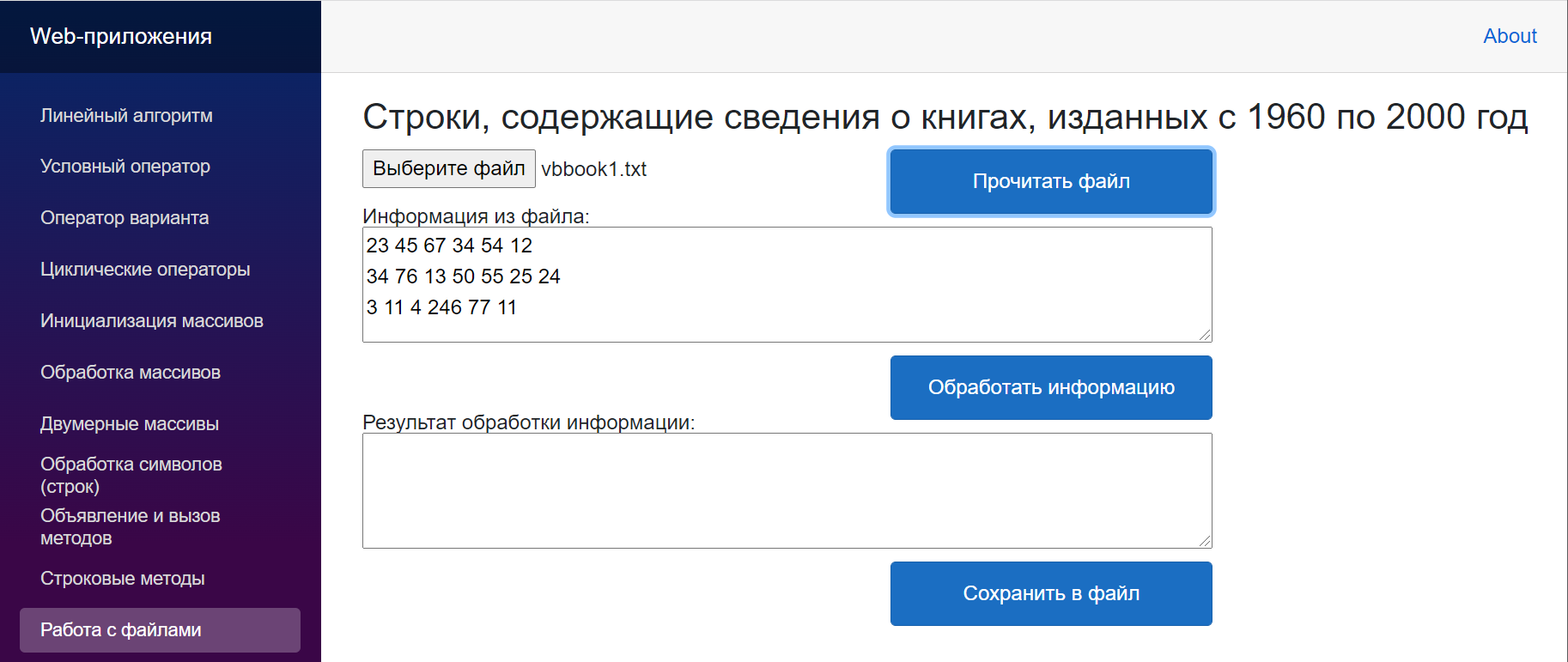


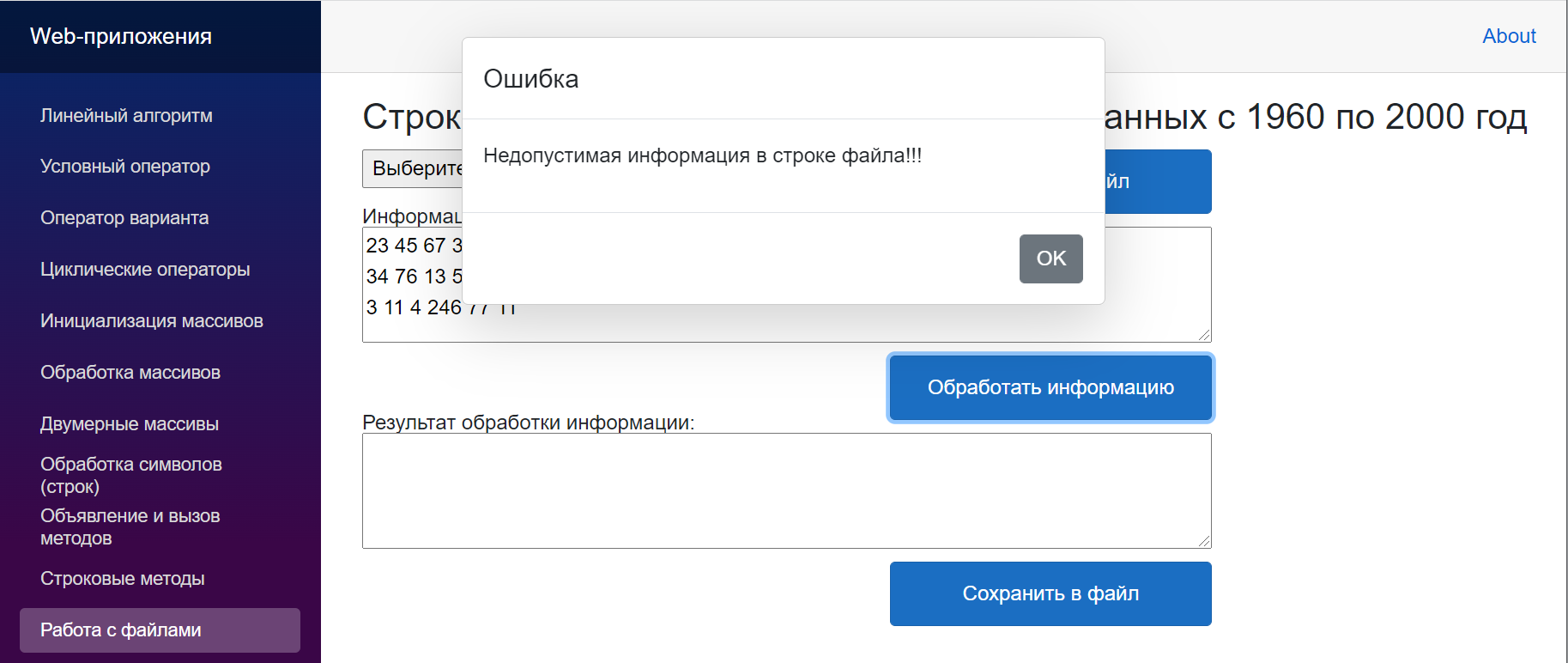


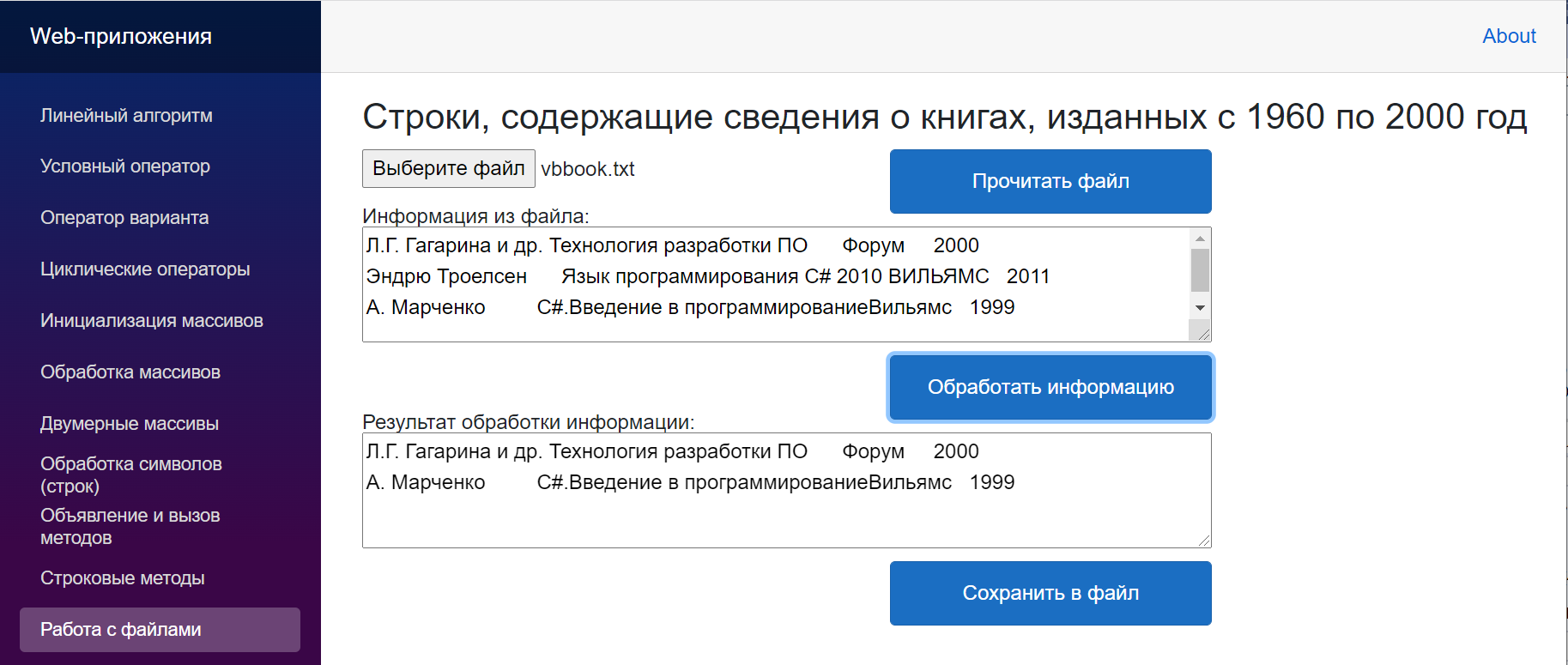
Скрины отлаженной программы:

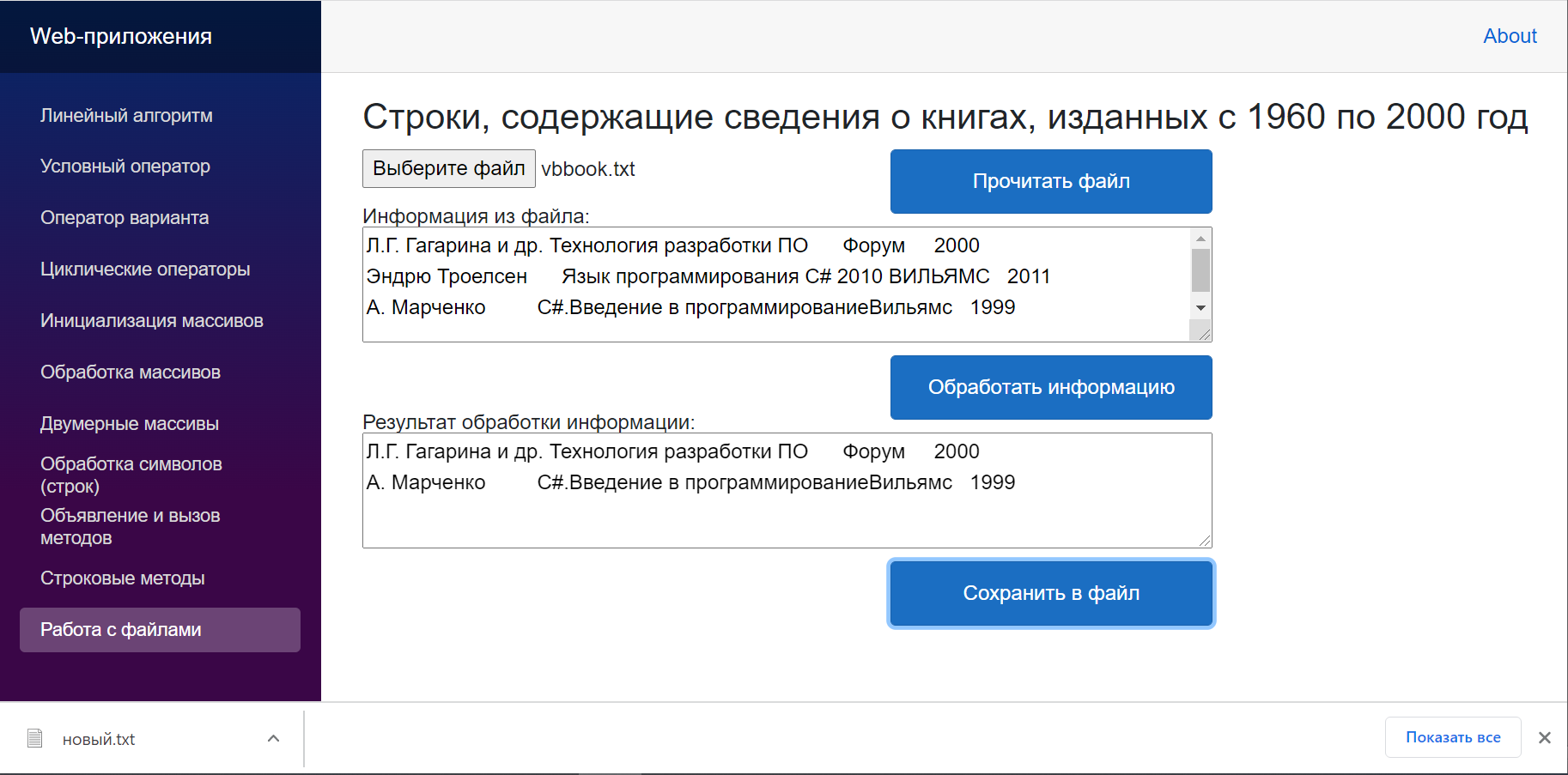


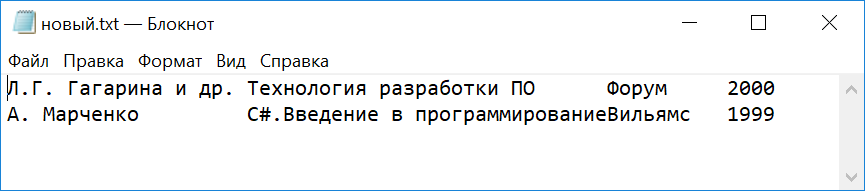












**4 Порядок выполнения работы**

4.1 Изучить теоретические сведения и задание к работе.

4.2 В соответствии с вариантом задания создать форму приложения Windows Forms (WPF, WebAssembly Blazor).

4.3 В соответствии с вариантом задания разработать и отладить коды приложения Windows Forms (WPF, WebAssembly Blazor), применяя методы работы с файлами и используя исключения, выбрасываемые, если форматы или значения входных данных не соответствуют спецификациям параметров обрабатывающих их методов.

**5 Содержание отчета**

5.1 Название работы.

5.2 Цель работы.

5.3 Программные коды и скрины работы разработанного приложения Windows Forms в соответствии с вариантом задания.

**6 Варианты заданий**

6.1 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Определить количество и номера строк, содержащих заданное число. Результаты с соответствующими комментариями дописать в конец исходного файла.

6.2 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти сумму и количество чётных чисел файла. Результаты с соответствующими комментариями дописать в конец исходного файла.

6.3 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Подсчитать количество строк, все элементы которых имеют положительные значения, и результат с соответствующим комментарием поместить в конец исходного файла.

6.4 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Извлечь квадратный корень из каждого числа в тех строках, все элементы которых имеют положительные значения. Создать файл, который будет содержать данные исходного файла, а затем преобразованные данные.

6.5 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Заменить в каждой строке элементы с наибольшим значением нулём. Создать файл, который будет содержать данные исходного файла, а затем преобразованные данные.

6.6 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти максимальные элементы в каждой строке исходного файла и поместить их в новый текстовый файл с соответствующими комментариями.

6.7 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти сумму чисел в каждой строке. Результаты с соответствующими комментариями поместить в новый текстовый файл.

6.8 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Найти среднее арифметическое чисел файла. Результат с соответствующим комментарием поместить в конец исходного файла.

6.9 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Заменить строку с максимальным количеством чисел в строке пустой строкой и удалить строку с минимальным количеством чисел в строке.

6.10 Даны два текстовых файла, состоящих из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Проверить поэлементно файлы на равенство. Если строки файлов не содержат поэлементно одинаковые числа, выдать номер строки и номер числа в строке, указывающих на первое несовпадение. Результат с соответствующим комментарием поместить в новый текстовый файл.

6.11 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Удалить из файла все числа, меньшие заданного числа. Если в текстовом файле таких чисел нет, оставить его без изменений.

6.12 Дан текстовый файл, состоящий из одной или нескольких строк. Каждая строка файла содержит числа, разделённые пробелами. Из каждой строки файла удалить числа, предшествующие первому положительному числу. Если в строке нет положительных чисел, оставить её без изменений.

6.13 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В каждой строке файла найти максимальную длину слова и подсчитать количество слов максимальной длины. После каждой строки файла вставить соответствующую строку-результат.

6.14 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Подсчитать количество слов в каждой строке. Результаты с соответствующими комментариями поместить в виде строк в конце исходного файла.

6.15 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. В начало каждой строки, содержащей заданное слово, добавить слово «ДОБАВЛЕНО».

6.16 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Удалить первое слово из каждой строки файла.

6.17 Дан текстовый файл, состоящий из нескольких строк. Строки содержат слова, разделённые пробелами. Перед каждым словом поместить номер слова в строке.

**7 Используемая литература**

7.1 Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул Технология разработки программного обеспечения. Форум, Инфра-М, 2009

7.2 Эндрю Троелсен Язык программирования С# 2010 и платформа .NET 4.0, М., ВИЛЬЯМС, 2011

7.3 А. Марченко C#. Введение в программирование, М, Вильямс, 2009

7.4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.