Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 12

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Файлы»

Выполнил: ст. гр. ТУУ-111

Богомолов В.Н.

Вариант №1

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_.12.2024

(дата приёмки)

Москва – 2025 г.

1. **Цель**

Освоить методы работы с текстовыми файлами в среде Microsoft Visual Studio на языке C#, реализовать алгоритмы анализа данных из файлов, научиться применять структуры управления (меню) и обеспечивать контроль корректности ввода/вывода.

1. **Формулировка задачи**

Разработать консольное приложение на языке C# в среде Microsoft Visual Studio для выполнения типовых задач по работе с файлами. Программа должна обеспечивать чтение предложений из заранее подготовленного текстового файла с содержимым, достаточным для организации постраничного вывода в консоль, где количество строк заведомо превышает доступное окно консоли. После заполнения доступных строк текстом вывод должен быть приостановлен до нажатия любой клавиши с отображением побуждающего сообщения «Вывести текст далее…». При подтверждении перехода к следующей странице содержимое консоли должно быть очищено. Реализовать интерфейс с текстовым меню, позволяющим пользователю выбрать действия. Обеспечить корректную обработку ошибок ввода-вывода, проверку существования файла перед его чтением (File.Exists), использование кодировки ANSI (Windows-1251). Соблюдать элементы этики программирования, включая осмысленные имена переменных, комментирование кода и организацию пользовательского интерфейса в соответствии с принципами удобства и интуитивности.

1. **Блок-схема алгоритма**

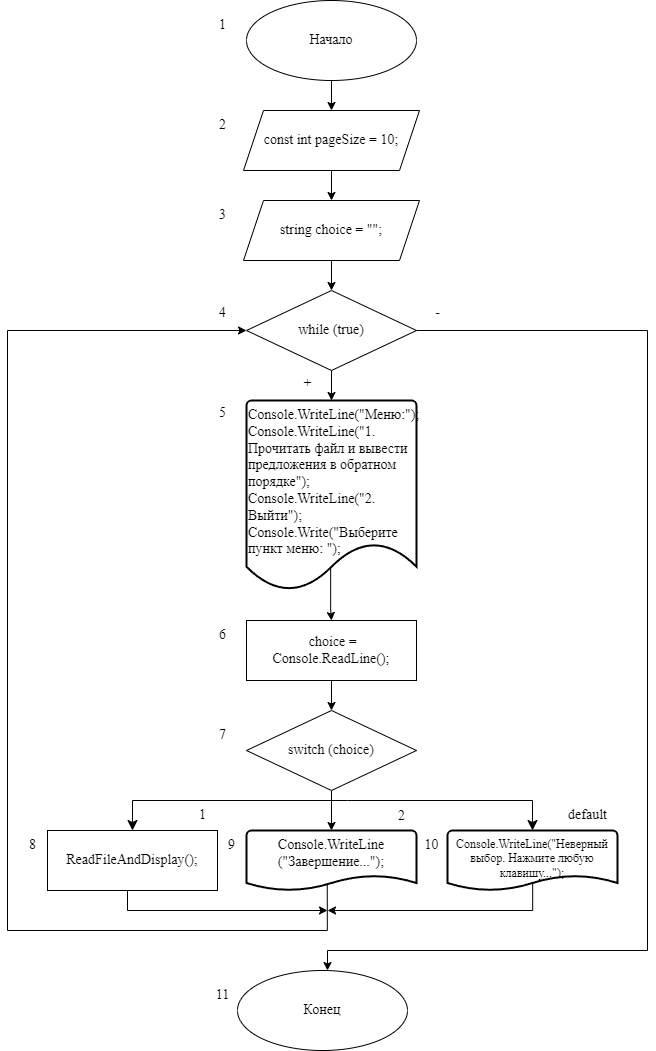
****

Рисунок 3.1.1 – Блок-схема алгоритма

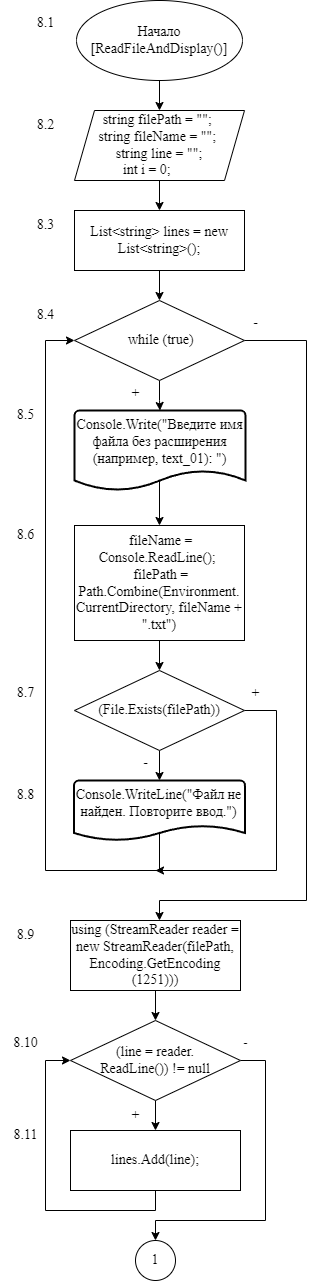


Рисунок 3.1.2 – Блок-схема алгоритма

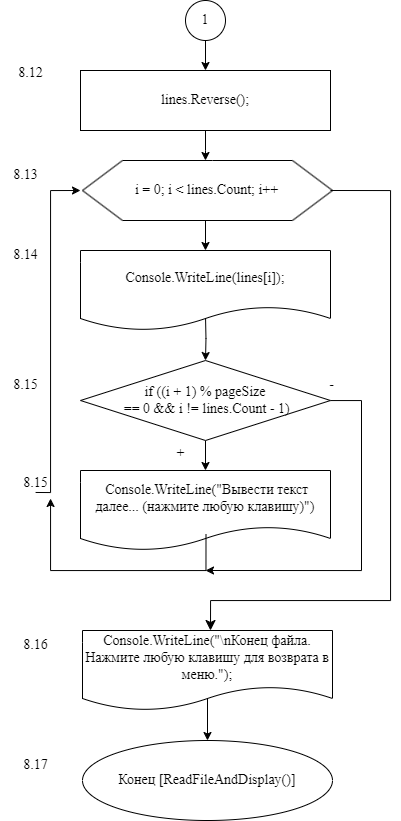


Рисунок 3.1.3 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. Тестовый пример №1

Ввод: прочитать файл и вывести предложения в обратном порядке => имя файла: text\_01

Ожидаемый результат:

Двенадцатое предложение.

Одиннадцатое предложение.

Десятое предложение.

Девятое предложение.

Восьмое предложение.

Седьмое предложение.

Шестое предложение.

Пятое предложение.

Четвёртое предложение.

Третье предложение.

Второе предложение.

Первое предложение.

* 1. Тестовый пример №2

Ввод: прочитать файл и вывести предложения в обратном порядке => имя файла: abc

Ожидаемый результат:

Файл не найден. Повторите ввод.

* 1. Тестовый пример №3

Ввод: abc

Ожидаемый результат:

Неверный выбор. Повторите попытку.

* 1. Тестовый пример №4

Ввод: выход

Ожидаемый результат: завершение.

1. **Листинг (код) программы**

using System;

using System.IO;

using System.Text;

using System.Collections.Generic;

namespace Lab12\_Variant1

{

class Program

{

// Константа — количество строк, выводимых на одну "страницу" консоли

const int pageSize = 10;

static void Main()

{

string choice = ""; // Переменная для хранения выбора пользователя

// Бесконечный цикл для отображения текстового меню

while (true)

{

Console.Clear(); // Очистка экрана перед выводом меню

// Вывод пунктов меню

Console.WriteLine("Меню:");

Console.WriteLine("1. Прочитать файл и вывести предложения в обратном порядке");

Console.WriteLine("2. Выйти");

Console.Write("Выберите пункт меню: ");

choice = Console.ReadLine(); // Считывание выбора пользователя

// Обработка выбора пользователя

switch (choice)

{

case "1":

ReadFileAndDisplay(); // Вызов метода для чтения и отображения файла

break;

case "2":

Console.WriteLine("Завершение...");

return; // Завершение работы программы

default:

// Обработка неправильного ввода

Console.WriteLine("Неверный выбор. Нажмите любую клавишу...");

Console.ReadKey(); // Ожидание ввода перед повтором

break;

}

}

}

// Метод для чтения строк из файла и вывода их в консоль в обратном порядке

static void ReadFileAndDisplay()

{

Console.Clear();

string filePath = ""; // Полный путь к файлу

string fileName = ""; // Имя файла, введённое пользователем

string line = ""; // Переменная для чтения строк из файла

int i = 0; // Счётчик для вывода постранично

// Список строк, считанных из файла

List<string> lines = new List<string>();

// Цикл запроса имени файла до тех пор, пока файл действительно не будет найден

while (true)

{

Console.Write("Введите имя файла без расширения (например, text\_01): ");

fileName = Console.ReadLine(); // Ввод имени файла без расширения

filePath = Path.Combine(Environment.CurrentDirectory, fileName + ".txt"); // Полный путь к файлу

if (File.Exists(filePath)) // Проверка существования файла

break;

else

Console.WriteLine("Файл не найден. Повторите ввод."); // Предупреждение при ошибке

}

// Открываем файл на чтение с кодировкой Windows-1251 (ANSI)

using (StreamReader reader = new StreamReader(filePath, Encoding.GetEncoding(1251)))

{

// Читаем файл построчно и добавляем строки в список

while ((line = reader.ReadLine()) != null)

{

lines.Add(line);

}

}

// Меняем порядок строк на обратный

lines.Reverse();

// Цикл постраничного вывода строк

for (i = 0; i < lines.Count; i++)

{

Console.WriteLine(lines[i]); // Вывод строки

// После вывода pageSize строк — пауза и очистка экрана

if ((i + 1) % pageSize == 0 && i != lines.Count - 1)

{

Console.WriteLine("Вывести текст далее... (нажмите любую клавишу)");

Console.ReadKey(); // Ожидание нажатия клавиши

Console.Clear(); // Очистка консоли перед следующей "страницей"

}

}

// Завершающее сообщение

Console.WriteLine("\nКонец файла. Нажмите любую клавишу для возврата в меню.");

Console.ReadKey();

}

}

}

1. **Расчет тестовых примеров на ПК**

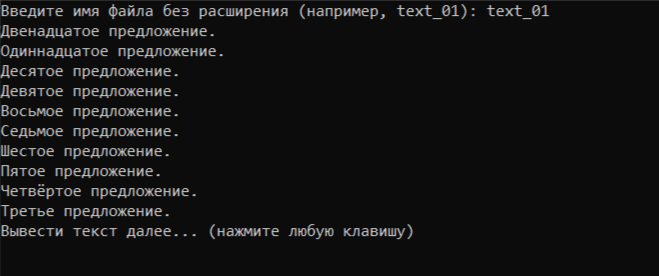
****

Рисунок 6.1.1 – Тестовый пример №1

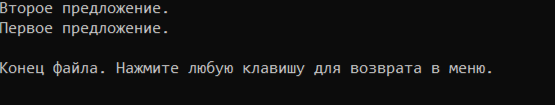
****

Рисунок 6.1.2 – Тестовый пример №1

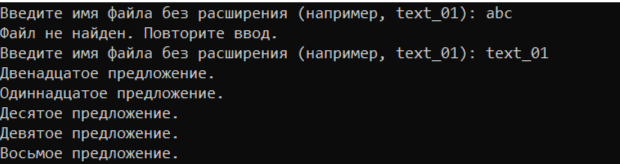


Рисунок 6.1.3 – Тестовый пример №2

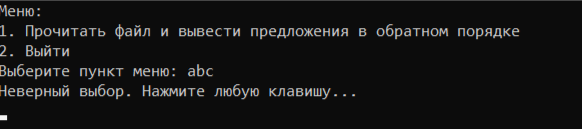


Рисунок 6.1.4 – Тестовый пример №3

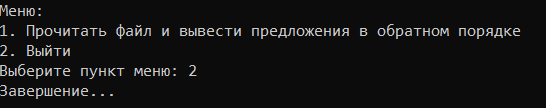


Рисунок 6.1.5 – Тестовый пример №4

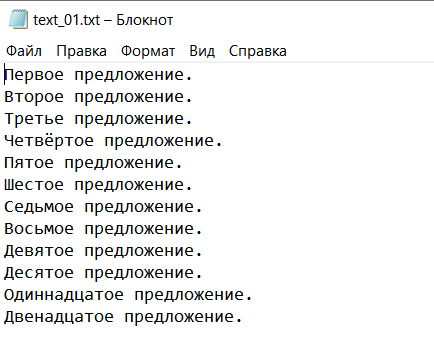


Рисунок 6.1.6 – Файл «text\_01»

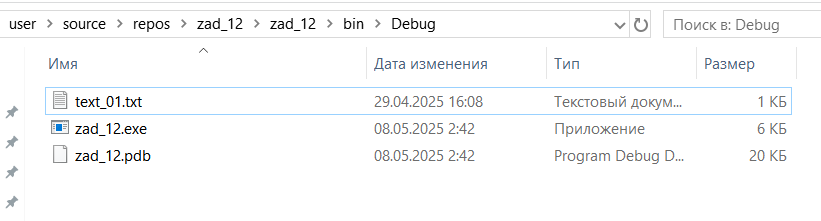


Рисунок 6.1.7 – Расположение файла

1. **Вывод**

В данной лабораторной работе на языке C# в среде Microsoft Visual Studio было разработано консольное приложение, читающее предложения из файла, выводящее их в обратном порядке с постраничным отображением. Были реализованы: проверка существования файла, чтение с кодировкой ANSI, управление выводом на экран с очисткой консоли, пользовательское меню. Программа корректно прошла тестирование и соответствует требованиям.