инистерство 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

Михалин. А. В. Вариант №2

30.05.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

(дата приёмки)

Москва – 2025 г.

**1. Цель**

В интегрированной среде разработки *Microsoft Visual Studio* на языке *Visual C#* в консольном режиме составить программное обеспечение для решения типовых задач программирования по тематике «Файлы» («*Files*»), заданных по варианту. Продумать организацию программного обеспечения, в котором чтение файла и его запись логически разделены пользовательским выбором пункта текстового меню. Предусмотреть выход из приложения.

**2. Формулировка задачи**

В интегрированной среде разработки (*IDE*) *Microsoft Visual Studio* создать консольную программу для учебных целей. Убедиться, что *IDE* создала решение (*solution*) с классической структурой каталогов и файлов программного продукта, разрабатываемого на языке *Visual C#* (*csproj*). Создать в каталоге решения (*solution*) «*Debug*», вложенном в каталог «*bin*», текстовый файл с именем «*nmbrs.txt*» и записать в него вручную (не программно) простые числа из диапазона от 1 до 500 через запятую «,» (таблица простых чисел размещена в Приложении). Суть ручного заполнения заключается в том, чтобы не составлять дополнительный алгоритм расчёта простых чисел при выполнении данного задания. Составление программного обеспечения для расчёта простых чисел допускается, но оно не должно становиться частью подготавливаемого отчёта по данной работе. Считать содержимое файла «*nmbrs.txt*» и вывести в консоль каждое третье значение, записанное через точку с запятой «;». По завершении чтения файла через пустую строку после выведенной конструкции вывести сумму значений, которые не приняли участия в выводе в консоль.

**3. Блок-схема алгоритма**



Рисунок 3.1 – Блок-схема алгоритма



Рисунок 3.2 – Блок-схема алгоритма

**4. Подбор тестовых примеров**

В качестве тестового примера для данной задачи является текстовый файл chisla в котором записаны числа: 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97,101,103,107,109,113,127,131,137,139,149,151,157,163,167,173,179,181,191,193,197,199,211,223,227,229,233,239,241,251,257,263,269,271,277,281,283,293,307,311,313,317,331,337,347,349,353,359,367,373,379,383,389,397,401,409,419,421,431,433,439,443,449,457,461,463,467,479,487,491,499

В результате программа должна вывести числа: 5;13;23;37;47;61;73;89;103;113;137;151;167;181;197;223;233;251;269;281;307;317;347;359;379;397;419;433;449;463;487

И число: 14525

**5. Листинг**

using System;

using System.Collections.Generic;

using System.IO;

namespace zadacha12

{

internal class Program

{

static void Main(string[] args)

{

ConsoleKeyInfo keyInfo;

do

{

Console.Clear();

// Выводит главное меню с инструкцией для пользователя

Console.WriteLine("Нажмите 'S' для запуска программы или 'Q' для выхода.");

// Считывает нажатую клавишу без отображения в консоли

keyInfo = Console.ReadKey(true);

// Если нажата клавиша 'S', запускается основная логика программы

if (keyInfo.Key == ConsoleKey.S)

{

RunProgram();

Console.WriteLine();

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

// Ожидает нажатия клавиши для возврата в меню

Console.ReadKey(true);

}

} while (keyInfo.Key != ConsoleKey.Q);

// Повторяет цикл до тех пор, пока пользователь не нажмёт 'Q' для выхода

}

static void RunProgram()

{

// Формирует путь к файлу chisla.txt в директории запуска приложения

string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "chisla.txt");

// Проверяет существование файла, если файла нет — сообщает и завершает метод

if (!File.Exists(filePath))

{

Console.WriteLine("Файл chisla.txt не найден.");

else

{

// Считывает весь текст из файла

string content = File.ReadAllText(filePath);

// Разделяет содержимое файла по запятым, получая массив строк

string[] parts = content.Split(',');

// Создаёт список для хранения чисел из файла

List<int> numbers = new List<int>();

// Переменная для суммы чисел, которые не будут выведены (не каждое третье)

int sumSkipped = 0;

// Обрабатывает каждую полученную строку

foreach (string part in parts)

{

string trimmed = part.Trim(); // Удаляет пробелы по краям

if (trimmed != "")

{

// Попытка преобразовать строку в число, если успешно — добавляет в список

if (int.TryParse(trimmed, out int num))

{

numbers.Add(num);

}

}

}

Console.WriteLine("Каждое третье число:");

// Перебирает все числа

for (int i = 0; i < numbers.Count; i++)

{

// Проверяет, является ли число "каждым третьим" (считая с 1)

if ((i + 1) % 3 == 0)

{

Console.Write(numbers[i]); // Выводит это число

// Добавляет точку с запятой после числа, если не пос

if (i + 3 < numbers.Count)

{

Console.Write(";");

}

}

else

{

// Суммирует числа, которые не выводятся

sumSkipped += numbers[i];

}

}

Console.WriteLine();

Console.WriteLine();

// Выводит сумму пропущенных чисел

Console.WriteLine("Сумма пропущенных чисел: " + sumSkipped);

}

}

}

}

**6. Расчет тестовых примеров на ПК**

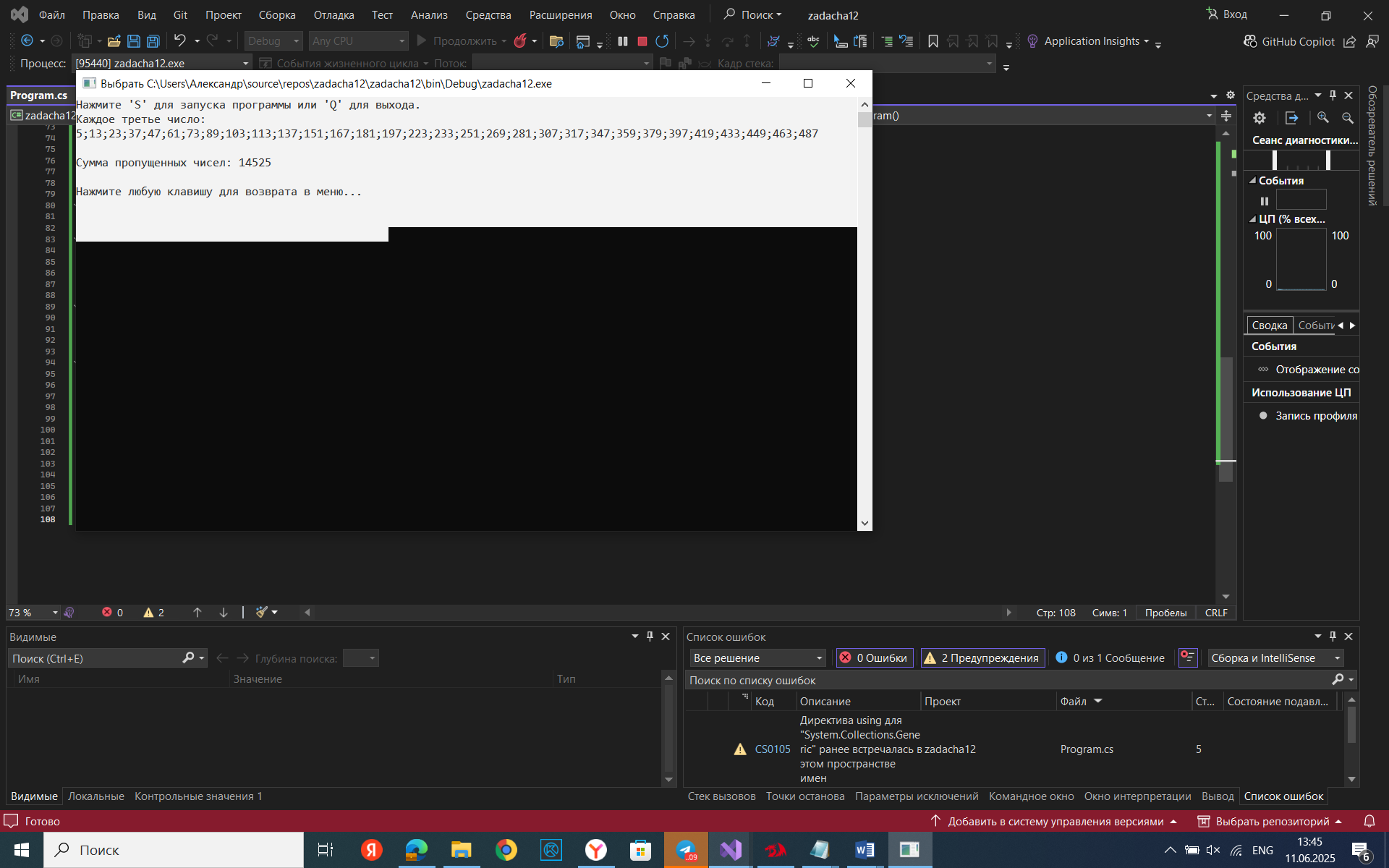
****

Рисунок 6.1 – Тестовый пример с выводом результата работы программы

**7.Вывод**

В ходе выполнения данной работы была составлена программа для на языке *Visual C#*, были получены навыки решения типовых задач по тематике «Файлы».