**Государственное образовательное учреждение**

**среднего профессионально образования «Тираспольский техникум информатики и права»**

**ОТЧЁТ**

**Практическое задание №4**

по дисциплине «Разработка программных модулей»

**тема: «Текстовые файлы»**

Работу выполнил

студент группы 215

Иванченко Даниил Богданович

Руководитель,

преподаватель

Подсекина Т.С.

Тирасполь, 2025

**Вариант 11**

**Задание 1**. создать текстовый файл с произвольной информацией. Организовать просмотр содержимого файла. Организовать чтение и обработку данных из файла в соответствии с индивидуальным заданием. Сохранить полученные результаты в новый текстовый файл. «Государство»: название страны; столица; государственный язык; население; площадь территории; денежная единица; государственный строй; глава государства. Вывести данные про государства, население которых больше 20 млн жителей.

В листинге 1 приведен код программы.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.IO;  class Program  {  class Country  {  public string Name;  public string Capital;  public string Language;  public int Population; // в миллионах  public double Area; // площадь территории  public string Currency;  public string GovernmentType;  public string HeadOfState;  public override string ToString()  {  return string.Format("{0}; {1}; {2}; {3}; {4}; {5}; {6}; {7}",  Name, Capital, Language, Population, Area, Currency, GovernmentType, HeadOfState);  }  }  static void Main()  {  string inputFile = @"D:\countries.txt";  string outputFile = @"D:\countries\_filtered.txt";  CreateSampleFile(inputFile);  Console.WriteLine("Содержимое файла \"{0}\":", inputFile);  string[] lines = File.ReadAllLines(inputFile);  foreach (string line in lines)  {  Console.WriteLine(line);  }  List<Country> countries = ReadCountries(inputFile);  List<Country> filtered = new List<Country>();  foreach (Country c in countries)  {  if (c.Population > 20)  filtered.Add(c);  }  WriteCountries(filtered, outputFile);  Console.WriteLine("\nГосударства с населением более 20 млн:");  foreach (Country c in filtered)  {  Console.WriteLine(c.ToString());  }  }  static void CreateSampleFile(string filePath)  {  string[] sampleData = {  "Россия; Москва; Русский; 146; 17098242; Рубль; Федерация; Президент",  "Канада; Оттава; Английский, французский; 38; 9984670; Канадский доллар; Конституционная монархия; Генерал-губернатор",  "Исландия; Рейкьявик; Исландский; 0; 103000; Крона; Республика; Президент",  "Германия; Берлин; Немецкий; 83; 357386; Евро; Федеративная республика; Президент",  "Швейцария; Берн; Немецкий, французский, итальянский; 8; 41285; Швейцарский франк; Федерация; Федеральный совет",  "США; Вашингтон; Английский; 331; 9833517; Доллар США; Федеральная республика; Президент"  };  File.WriteAllLines(filePath, sampleData);  }  static List<Country> ReadCountries(string filePath)  {  List<Country> countries = new List<Country>();  string[] lines = File.ReadAllLines(filePath);  foreach (string line in lines)  {  string[] parts = line.Split(';');  if (parts.Length == 8)  {  for (int i = 0; i < parts.Length; i++)  {  parts[i] = parts[i].Trim();  }  int population;  double area;  bool popOk = int.TryParse(parts[3], out population);  bool areaOk = double.TryParse(parts[4], out area);  if (popOk && areaOk)  {  Country country = new Country  {  Name = parts[0],  Capital = parts[1],  Language = parts[2],  Population = population,  Area = area,  Currency = parts[5],  GovernmentType = parts[6],  HeadOfState = parts[7]  };  countries.Add(country);  }  }  }  return countries;  }  static void WriteCountries(List<Country> countries, string filePath)  {  List<string> lines = new List<string>();  foreach (Country c in countries)  {  lines.Add(c.ToString());  }  File.WriteAllLines(filePath, lines);  }  } |

Листинг 1 – программа определения алгоритмов

Ниже на рисунках 1 представлены тесты программы при разных входных параметрах.

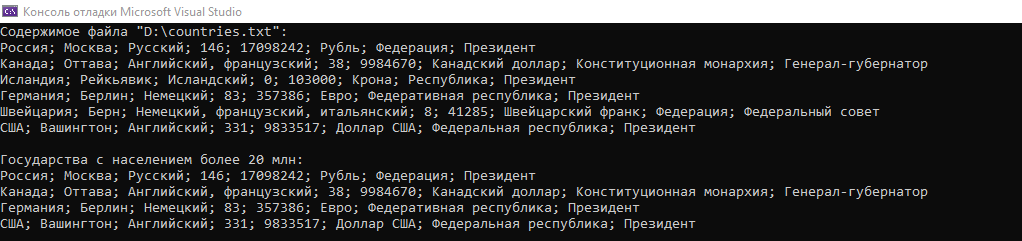


Рисунок 1 – тест 1