Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**«ОБРАБОТКА ДАННЫХ В ФАЙЛАХ»**

**ОТЧЕТ**

**по лабораторной работе №11**

**дисциплины**

**«Технологии программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Мизин Глеб Егорович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

**Индивидуальное задание**: в файле содержатся сведения о производстве товаров работниками (ID, сотрудник, категория товара, з/п рабочего, количество произведенных товаров, цена за единицу товара). 1. Определите сколько рабочих получают больше, чем вырабатывают продукции. 2. Суммарный объем произведенной продукции (в валюте) без категории. 3. Суммарный объем в валюте произведенной продукции по каждой из 4-х категорий. 4. Самый эффективный сотрудник: с максимальной разницей между произведенным продуктом и зарплатой.

namespace Helloworld

{

    class Program

    {

        static void Main(string[] *args*)

        {

            Console.BackgroundColor = ConsoleColor.White;

            Console.ForegroundColor = ConsoleColor.Black;

            Console.Clear();

            Manufacture m = new Manufacture();

            m.Load("lr11\_12.csv");

            Console.ReadKey();

        }

    }

    public class Manufacture

        {

            public Manufacture()

            {

            }

            String ID {get; set;}

            String Name {get; set;}

            String Category {get; set;}

            public float Salary {get; set;}

            public float Count {get; set;}

            public float Price {get; set;}

            public float Substract;

            public Boolean Load(string *pFileName*)

            {

                if (File.Exists(*pFileName*))

                {

                        StreamReader f\_in = new StreamReader(*pFileName*);

                        List<Manufacture> all = new List<Manufacture>();

                        try

                        {

                            String line = f\_in.ReadLine();

                            while((line = f\_in.ReadLine()) != null)

                                {

                                    all.Add(Manufacture.Create(line));

                                }

                        }

                        catch (Exception ex)

                        {

                            Console.WriteLine(ex.Message);

                        }

#if DEBUG

                        TextWriter save\_out = Console.Out;

                        var new\_out = new StreamWriter(@"LR\_11\_OUT.txt");

                        Console.SetOut(new\_out);

                        foreach (var m in all)

                            Console.WriteLine(m);

                        Console.SetOut(save\_out);

                        new\_out.Close();

#endif

                        Console.WriteLine("\*\*\*\*\*\*\*Задача 1\*\*\*\*\*\*\*\*");

                        int Counter = all.FindAll(*m* => *m*.Count \* *m*.Price < *m*.Salary).ToList().Count;

                        Console.WriteLine("Количество сотрудников у которых з/п больше произведённой продукции: {0} \n", Counter);

                        Console.WriteLine("\*\*\*\*\*\*\*Задача 2\*\*\*\*\*\*\*\*");

                        float Sum = 0;

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

                            Sum += all[i].Count \* all[i].Price;

                        Console.WriteLine("Общий объём произведённой продукции (в валюте): {0} $\n", Sum);

                        Console.WriteLine("\*\*\*\*\*\*\*Задача 3\*\*\*\*\*\*\*\*");

                        float SumA = 0;

                        float SumB = 0;

                        float SumC = 0;

                        float SumD = 0;

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

                            {

                            if(all[i].Category == "A")

                                SumA += all[i].Count \* all[i].Price;

                            else if(all[i].Category == "B")

                                SumB += all[i].Count \* all[i].Price;

                            else if(all[i].Category == "C")

                                SumC += all[i].Count \* all[i].Price;

                            else

                                SumD += all[i].Count \* all[i].Price;

                            }

                        Console.WriteLine("Общий объём произведённой продукции типа А (в валюте): {0} $\n"+

                                        "Общий объём произведённой продукции типа B (в валюте): {1} $\n"+

                                        "Общий объём произведённой продукции типа C (в валюте): {2} $\n"+

                                        "Общий объём произведённой продукции типа D (в валюте): {3} $\n",

                                        SumA, SumB, SumC, SumD);

                        Console.WriteLine("\*\*\*\*\*\*\*Задача 4\*\*\*\*\*\*\*\*");

                            float maxE = 0;

                            int ind = 0;

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

                                    if (maxE < all[i].Substract)

                                    {

                                        maxE = all[i].Substract;

                                        ind = i;

                                    }

                                    else maxE = maxE;

                            Console.WriteLine("Данные сотрудника с максимальной эффективностью: {0}\n\n"+

                                            "Эффективность сотрудника: {1}\n", all[ind], maxE);

                }

                return false;

            }

            public static Manufacture Create(string *str*)

            {

                Manufacture m = new Manufacture();

                string[] e = *str*.Split(',');

                m.ID = e[0].Trim();

                m.Name = e[1].Trim();

                m.Category = e[2].Trim();

                m.Salary = Convert.ToSingle(e[3].TrimStart('$').Replace('.',','));

                m.Count = Convert.ToSingle(e[4].Trim());

                m.Price = Convert.ToSingle(e[5].TrimStart('$').Replace('.',','));

                m.Substract = (m.Count \* m.Price - m.Salary);

                return m;

            }

            public override string ToString()

            {

                String s = string.Format(

                    "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"+

                    "ID: {0}, Name: {1}, Category: {2}\n"+

                    "Salary: {3}, Count: {4}, Price: {5}",

                    ID,Name,Category,Salary,Count,Price);

                    return s;

            }

        }

}

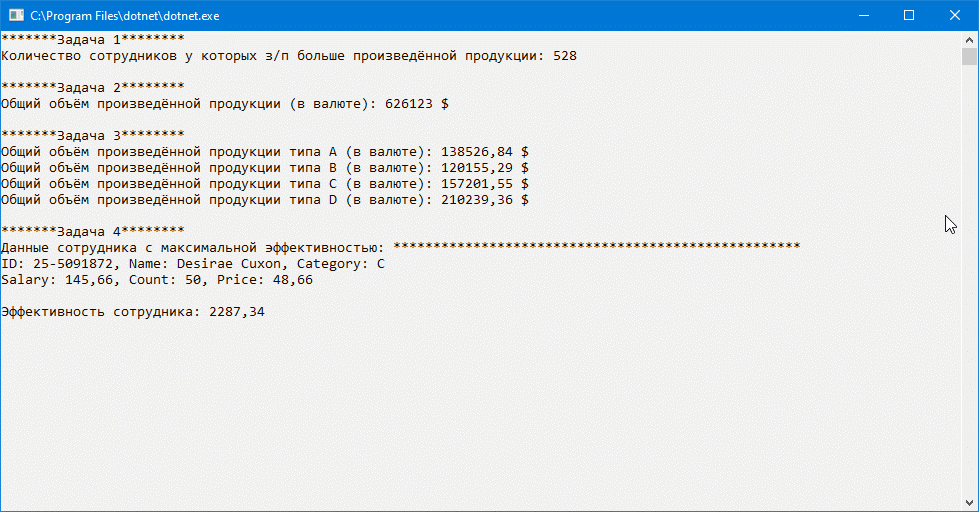


Рисунок 1 – Результат работы программы в режиме Debug

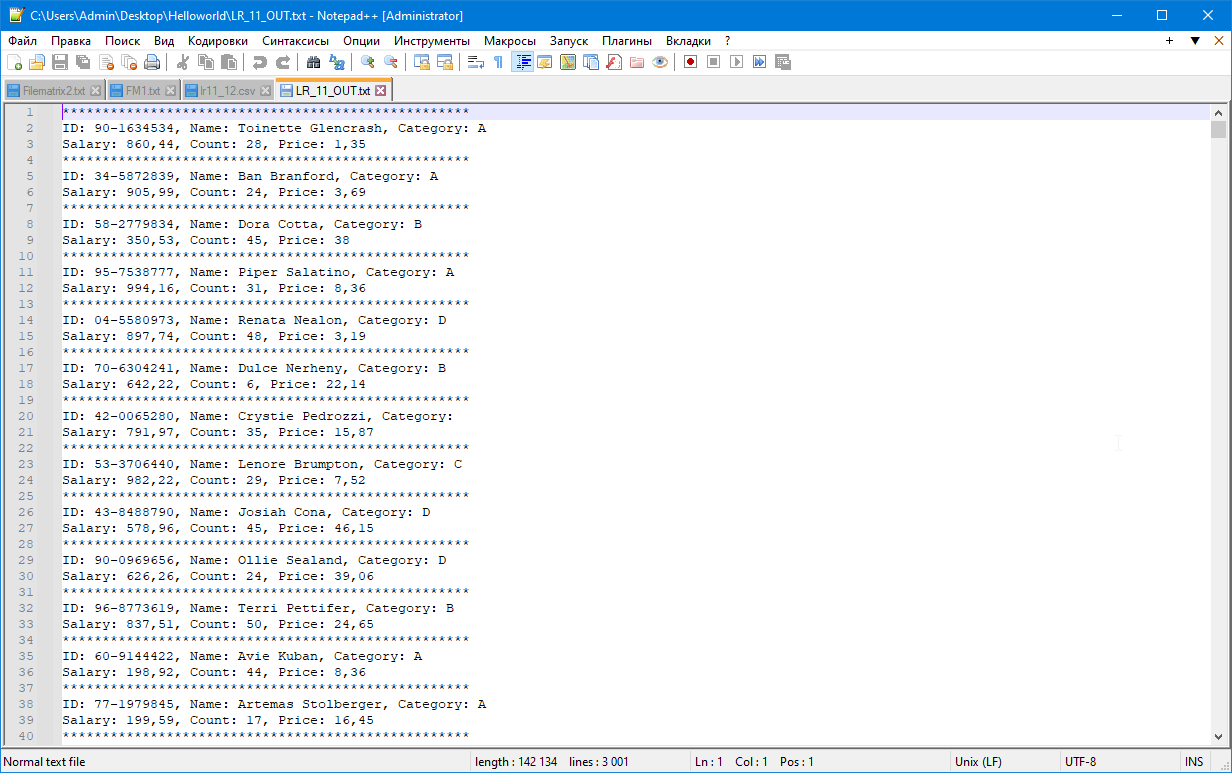


Рисунок 2 – Результат работы программы в режиме Release

**Контрольные вопросы**:

1. Дайте определение понятия «файл».

Файл – именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными в операционных системах

2. Опишите формат файла \*.csv.

CSV-файлы (файлы данных с разделителями-запятыми) — это файлы особого типа, которые можно создавать и редактировать в Excel. В CSV-файлах данные хранятся не в столбцах, а разделенные запятыми. Текст и числа, сохраненные в CSV-файле, можно легко переносить из одной программы в другую.

3. Поясните назначение различных конфигураций (debug или release) при сборке приложения в среде Visual Studio.

Debug и Release нужны для отображения различных выходных данных

4. Приведите пример кода, реализующего различное поведение программы при использовании различных режимов сборки (debug или release).

с.м индивидуальное задание ЛР\_11

5. Что такое поток? Какие типы классов потоков используются при работе с файлами?

Поток – это абстрактное представление последовательного устройства. Последовательное устройство – это нечто такое, что хранит данные в линейной структуре и точно таким же образом обеспечивает доступ к ним: считывает или записывает по одному байту за одну единицу времени.

FileStream

6. Опишите последовательность действий при необходимости записать одну строку в файл. Приведите примеры использования различных классов.

