**8.Перечесление.Структуры.**

Задание 1. Описать структуру с именем TRAIN, содержащую следующие поля:

название пункта назначения;

номер поезда;

время отправления.

Написать программу, выполняющую следующие действия:

ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN (записи должны быть упорядочены по номерам поездов);

вывод на экран информации о поезде, номер которого введен с клавиатуры (если таких поездов нет, вывести соответствующее сообщение).

namespace task1

{

internal class Program

{

static void Main(string[] args)

{

Train[] train = new Train[8];

for(int i = 0; i < train.Length ; i++)

{

Console.WriteLine("Введите пунтк назначения: ");

train[i].Destination = Console.ReadLine();

Console.WriteLine("Введите номер поезда");

train[i].Number = int.Parse(Console.ReadLine());

Console.WriteLine("Введите время отправления");

train[i].Time = DateTime.Parse( Console.ReadLine());

}

Array.Sort(train, (x, y) => x.Number.CompareTo(y.Number));

for (int i = 0; i < train.Length; i++)

{

Console.WriteLine($"Пунтк назначения: {train[i].Destination}");

Console.WriteLine($"Номер поезда: {train[i].Number}");

Console.WriteLine($"Время отправления: {train[i].Time:g}");

}

Console.WriteLine("Введите номер поезда для поиска: ");

int trainNumber = int.Parse(Console.ReadLine());

for (int i = 0; i < train.Length; i++)

{

if (train[i].Number == trainNumber)

{

Console.WriteLine($"Пунтк назначения: {train[i].Destination}");

Console.WriteLine($"Номер поезда: {train[i].Number}");

Console.WriteLine($"Время отправления: {train[i].Time:g}");

}

}

}

}

}

struct Train

{

private string \_destination;

private int \_number;

private DateTime \_time;

public string Destination

{

get { return \_destination; }

set { \_destination = value; }

}

public int Number

{

get { return \_number; }

set { \_number = value; }

}

public DateTime Time

{

get { return \_time; }

set { \_time = value; }

}

public Train(string destination, int number, DateTime time)

{

\_destination = destination;

\_number = number;

\_time = time;

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

Рисунок 1.1 – Результат работы программы

Задание 2. Создайте перечисление, в котором будут содержаться должности сотрудников как имена констант. Присвойте каждой константе значение, задающее количество часов, которые должен отработать сотрудник за месяц.

namespace task2

{

enum EmployeePosition

{

Manager = 160,

Developer = 176,

Designer = 144

}

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Должности и количество отработанных часов в месяц:");

Console.WriteLine("Менеджер: " + (int)EmployeePosition.Manager);

Console.WriteLine("Разработчик: " + (int)EmployeePosition.Developer);

Console.WriteLine("Дизайнер: " + (int)EmployeePosition.Designer);

Console.ReadLine();

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Должности и количество отработанных часов в месяц:  Менеджер: 160  Разработчик: 176  Дизайнер: 144 |

Анализ результатов:

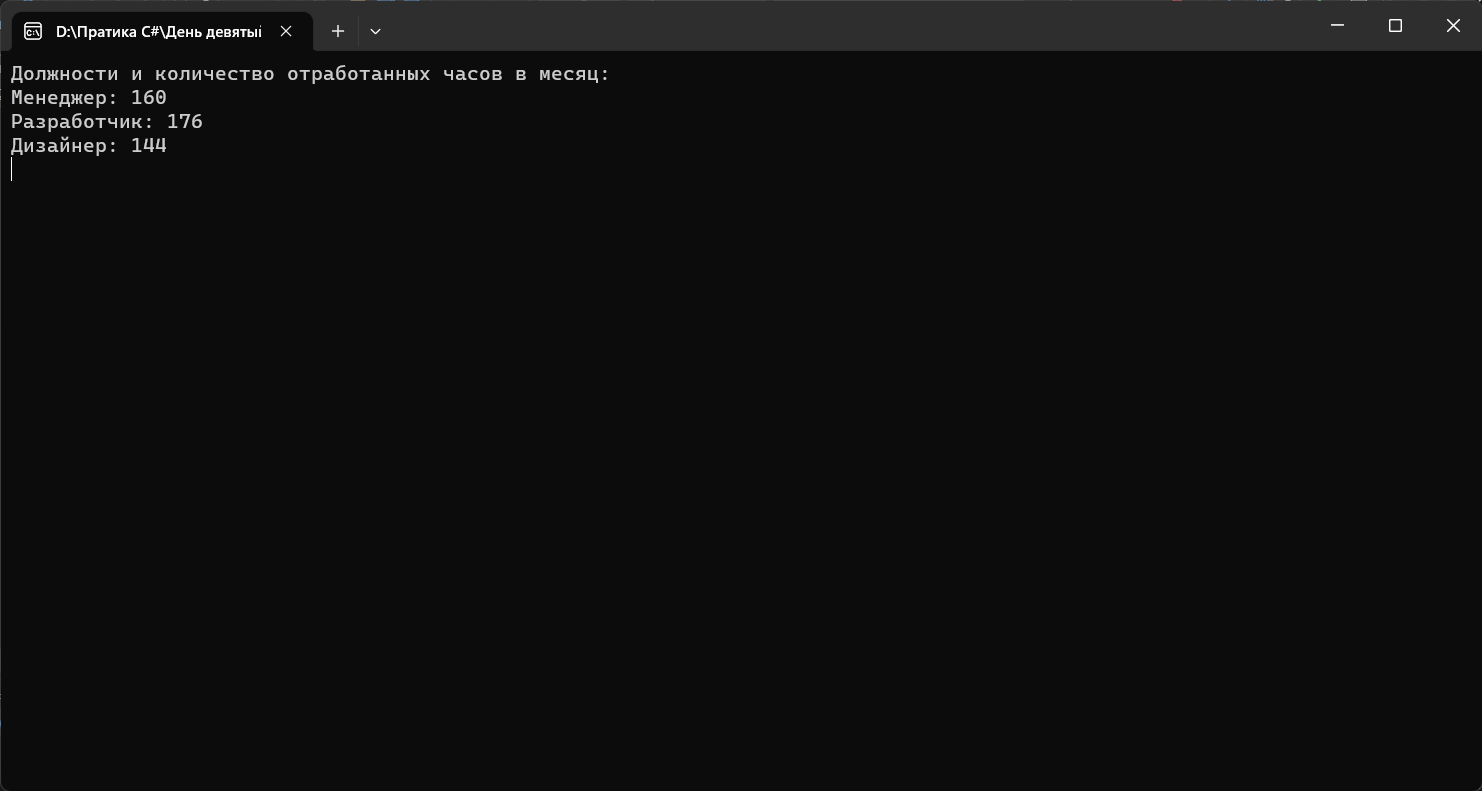
**

Рисунок 1.2 – Результат работы программы

работы программы