**8.** **Перечисления. Структуры**

Задание №1

Описать структуру с именем AEROFLOT, содержащую следующие поля:

-название пункта назначения рейса;

-номер рейса;

-тип самолета.

Написать программу, выполняющую следующие действия:

-ввод с клавиатуры данных в массив, состоящий из семи элементов

типа AEROFLOT (записи должны быть размещены в алфавитном порядке по названиям пунктов назначения);

-вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры (если таких рейсов нет, вывести соответствующее сообщение).

Листинг программы:

Program:

using Z1;

using MyLibrary;

Aeroflot[] aeroflots = new Aeroflot[7];

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

{

string destination = MyTool.Input("Введите пункт назначения: ");

int flightNumber = Convert.ToInt32(MyTool.Input("Введите номер рейса: "));

string typeAircraft = MyTool.Input("Введите тип самолета: "); ;

aeroflots[i] = new Aeroflot(destination, flightNumber, typeAircraft);

}

var aeroflotsSort = aeroflots.OrderBy(a => a.Destination);

foreach (var a in aeroflotsSort)

{

Console.WriteLine($"{a.Destination}");

}

string \_typeAircraft = MyTool.Input("Введите тип самолета: ");

if(aeroflotsSort.Any(s => s.TypeAircraft == \_typeAircraft))

{

foreach(var a in aeroflotsSort)

{

if (a.TypeAircraft == \_typeAircraft)

{

Console.WriteLine($"Пункт назначения: {a.Destination}");

Console.WriteLine($"Номер рейса: {a.FlightNumber}");

Console.WriteLine($"Тип самолета: {a.TypeAircraft}");

Console.WriteLine();

}

}

}

else

{

Console.WriteLine("Нет такого типа");

}

Aeroflot:

namespace Z1

{

record struct Aeroflot

{

private string destination;

private int flightNumber;

private string typeAircraft;

public string Destination

{

get { return destination; }

}

public int FlightNumber

{

get { return flightNumber; }

}

public string TypeAircraft

{

get { return typeAircraft; }

}

public Aeroflot(string destination, int flightNumber, string typeAircraft)

{

this.destination = destination;

this.flightNumber = flightNumber;

this.typeAircraft = typeAircraft;

}

}

}

Таблица 8.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Брест, 123, боинг;  Брест, 124, грузовой;  Минск, 111, боинг;  Минск, 112, грузовой;  Москва, 100, боинг;  Москва, 101, грузовой;  Париж, 108, боинг; | Введите тип самолета: боинг  Пункт назначения: Брест  Номер рейса: 123  Тип самолета: боинг  Пункт назначения: Минск  Номер рейса: 111  Тип самолета: боинг  Пункт назначения: Москва  Номер рейса: 100  Тип самолета: боинг  Пункт назначения: Париж  Номер рейса: 108  Тип самолета: боинг |

Анализ результатов:

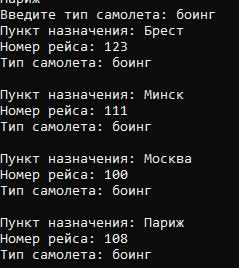


Рисунок 8.1 – Результат выполнения программы задание №1

Задание 2

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

2. Создайте класс Accauntant с методом bool AskForBonus (Post worker, int hours), отражающее давать или нет сотруднику премию. Если сотрудник отработал больше положеных часов в месяц, то ему положена премия.

Листинг программы:

Post:

enum Post

{

director = 30,

work = 40

}

Accauntant:

internal class Accauntant

{

public bool AskForBonus(Post post, int hours) => (int)post < hours;

}

Program:

using Z2;

using MyLibrary;

Accauntant accauntant = new();

int hours = Convert.ToInt32(MyTool.Input("Введите кол-во чисов: "));

if (accauntant.AskForBonus(Post.director, hours))

{

Console.WriteLine("Премия положена");

}

else

{

Console.WriteLine("Премия не положена");

}

Таблица 8.2 – Входные и выходные данные программы задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 12 | Премия не положена |

Анализ результатов:



Рисунок 8.2 – Результат выполнения программы задание №2