**8.Перечисления. Структуры**

Задание №1Описать структуру с именем MARSH, содержащую следующие поля:

название начального пункта маршрута;

название конечного пункта маршрута;

номер маршрута.

Написать программу, выполняющую следующие действия:

ввод с клавиатуры данных в массив, состоящий из восьми

элементов типа MARSH (записи должны быть упорядочены по номерам

маршрутов);

вывод на экран информации о маршруте, номер которого введен с

клавиатуры (если таких маршрутов нет, вывести соответствующее сообщение).

Листинг программы:

internal class MemoryMarshProvider

{

public static List<Marsh> GetMarshes()

{

return new List<Marsh>()

{ new Marsh("Гродно","Минск", 1 ),

new Marsh("Гродно", "Мосты", 2),

new Marsh("Гродно", "Питер", 3),

new Marsh("Гродно", "Гомель", 4),

new Marsh("Гродно", "Волковыск", 5)

};

}

}

struct Marsh

{

public string StartingPoint;

public string Destination;

public int RouteNumber;

public Marsh(String StartingPoint, String Destination, int RouteNumber)

{

this.StartingPoint = StartingPoint;

this.Destination = Destination;

this.RouteNumber = RouteNumber;

}

public string WriteMarsh()

{

return $"Номер маршрута {RouteNumber}\nНачальный пункт маршрута:{StartingPoint}\nКонечный пункт маршрута:{Destination}";

}

}

class Program

{

static void Main()

{

var marsh1 = new Marsh(Faker.Address.City(), Faker.Address.City(), new Random().Next(1, 100));

var marshrutes=MemoryMarshProvider.GetMarshes();

Console.WriteLine("Введите номер маршрута: ");

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

bool hasMarshrutes=false;

foreach (var marsh in marshrutes)

{

if (marsh.RouteNumber==number)

{

Console.WriteLine(marsh.WriteMarsh());

hasMarshrutes=true;

}

}

if (hasMarshrutes==false)

{

Console.WriteLine("Такого маршрута нет");

}

Таблица 8.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Номер маршрута :3 | Начальный пункт маршрута: Гродно  Конечный пункт маршрута: Питер |

Анализ результатов:

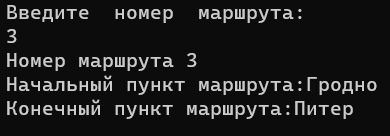


Рисунок 8.1 – Результат работы программы из задания №1

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

Листинг программы:

class program

{

enum JobTitle

{

Director = 140,

Accountant = 120,

Engineer = 135,

Manager = 130

}

static string AskForBonus(JobTitle worker, int hours)

{

if ((int)worker < hours)

{

return "Начислить премию";

}

return "Премию не начислять";

}

static void Main()

{

Console.WriteLine("Выберите вашу должность: \n1. Директор \n2. Бухгалтер \n3. Инженер \n4. Менеджер");

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

var workers=Enum.GetValues(typeof(JobTitle)).Cast<JobTitle>().ToList();

var worker = workers[workNumber - 1];

Console.WriteLine("Введите количество рабочих часов");

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

var message= AskForBonus(worker,hours);

Console.WriteLine(message);

}

Таблица 8.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Должность:1  Количество рабочих часов: 140 | Начислить премию |

Анализ результатов:

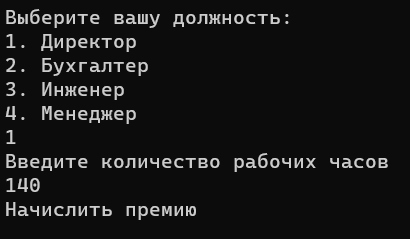


Рисунок 8.2 – Результат работы программы из задания №2