**8. Перечисления. Структуры**

Задание 1. Структуры

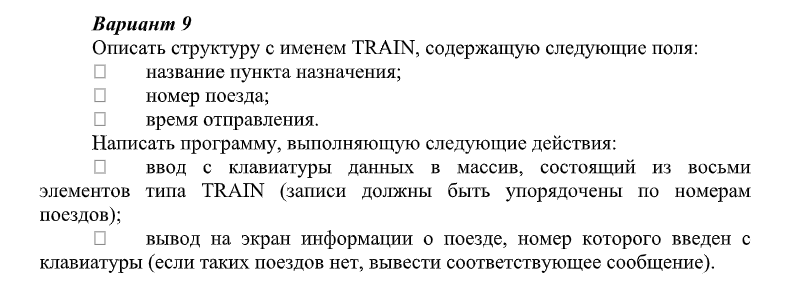


Рисунок 8.1 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

namespace @struct

{

struct TRAIN

{

public string Destination;

public int Number;

public DateTime Departure;

public TRAIN(string destination, int number, DateTime departure)

{

Destination = destination;

Number = number;

Departure = departure;

}

public override string ToString()

{

return string.Format("Пункт назначения: {0}, № поезда: {1}, Отправление: {2}", Destination, Number, Departure);

}

}

class Program

{

static void Main(string[] args)

{

TRAIN[] trains = new TRAIN[6];

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

{

Console.WriteLine("Введите название пункта назначения: ");

string destination = Console.ReadLine();

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

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

Console.WriteLine("Введите дату отправления: ");

DateTime departure = DateTime.Parse(Console.ReadLine());

trains[i] = new TRAIN(destination, number, departure);

}

Console.WriteLine("Введите название пункта назначения для поиска или end для выхода");

string input;

while ((input = Console.ReadLine()) != "end")

{

bool trainFound = false;

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

{

if (trains[i].Destination == input)

{

Console.WriteLine(trains[i]);

trainFound = true;

}

}

if (!trainFound)

Console.WriteLine("Ничего не найдено. Повторите попытку");

}

}

}

}

Таблица 8.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Cидней, 6, 12.12.2023 | Информация о пункте назначения |

Анализ результатов:

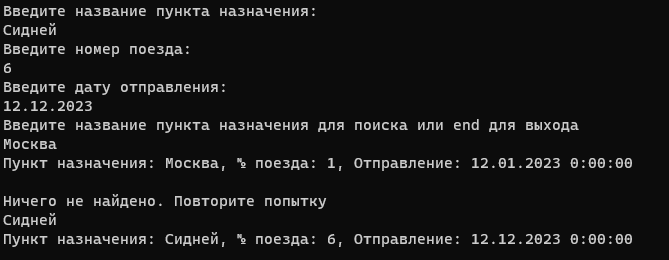


Рисунок 8.2 – Результат работы программы

Источник: собственная разработка

Задание 2. Перечисления

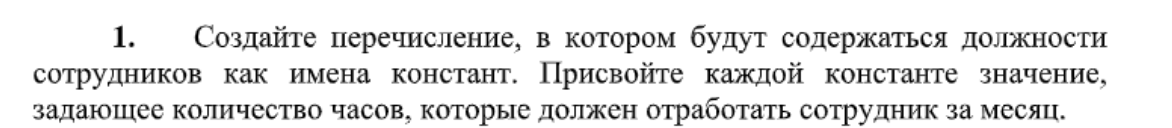


Рисунок 8.3 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace zad2

{

internal class Program

{

public static void Main()

{

Employees andrey = Employees.Andrey;

Employees vadim = Employees.Vadim;

Employees dima = Employees.Dima;

Employees artur = Employees.Artur;

Employees lank = Employees.Lank;

Employees vova = Employees.Vova;

Employees kirill = Employees.Kirill;

string[] employeersNames = { "andrey", "vadim", "dima", "artur", "lank", "vova", "kirill" };

printAllEmployers(employeersNames);

Console.WriteLine("Введите имя сотрудника о котором хотите узнать информацию(для выходы из программы напишите exit): ");

string employeeChecker = Console.ReadLine();

switch (employeeChecker)

{

case "andrey":

EmployeeInformation(andrey);

break;

case "vadim":

EmployeeInformation(vadim);

break;

case "dima":

EmployeeInformation(dima);

break;

case "artur":

EmployeeInformation(artur);

break;

case "lank":

EmployeeInformation(lank);

break;

case "vova":

EmployeeInformation(vova);

break;

case "kirill":

EmployeeInformation(kirill);

break;

case "exit:":

break;

}

}

enum Employees

{

Andrey = 10,

Vadim = 20,

Dima = 5,

Artur = 15,

Lank = 19,

Vova = 30,

Kirill = 44,

}

static void EmployeeInformation(Employees employees)

{

switch (employees)

{

case Employees.Andrey:

Console.WriteLine($"Сотруднику {Employees.Andrey} требуется отработать:{(int)Employees.Andrey} часов за месяц");

break;

case Employees.Vadim:

Console.WriteLine($"Сотруднику {Employees.Vadim} требуется отработать:{(int)Employees.Vadim} часов за месяц");

break;

case Employees.Dima:

Console.WriteLine($"Сотруднику {Employees.Dima} требуется отработать:{(int)Employees.Dima} часов за месяц");

break;

case Employees.Artur:

Console.WriteLine($"Сотруднику {Employees.Artur} требуется отработать:{(int)Employees.Artur} часов за месяц");

break;

case Employees.Lank:

Console.WriteLine($"Сотруднику {Employees.Lank} требуется отработать:{(int)Employees.Lank} часов за месяц");

break;

case Employees.Vova:

Console.WriteLine($"Сотруднику {Employees.Vova} требуется отработать:{(int)Employees.Vova} часов за месяц");

break;

case Employees.Kirill:

Console.WriteLine($"Сотруднику {Employees.Kirill} требуется отработать:{(int)Employees.Kirill} часов за месяц");

break;

}

}

static void printAllEmployers(string[] arr)

{

Console.WriteLine("Все сотрудники:");

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

{

Console.WriteLine(i + "." + arr[i]);

}

}

}

}

Таблица 8.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| artur | Сотруднику Artur требуется: 15 часов за месяц. |

Анализ результатов:

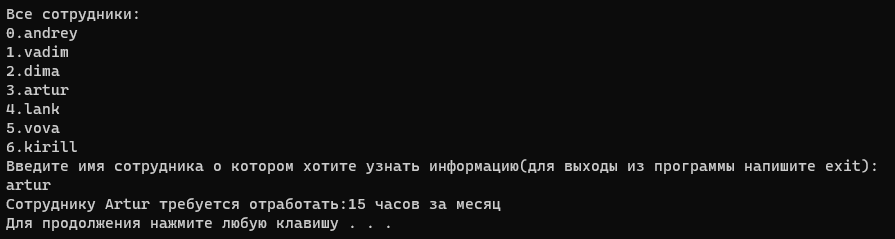


Рисунок 8.4 – Результат работы программы

Источник: собственная разработка