**9 Интерфейсы.**

Задание 1. Интерфейсы Ix, Iy, Iz, содержат объявления методов с одной и той же сигнатурой следующим образом. Реализовать задания представленные на рисунке 9.1.

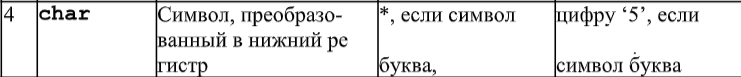


Рисунок 9.1 – Индивидуальные задания

Источник: лабораторная работа

Листинг программы:

try

{

TestClass task1 = new TestClass('F');

Ix taskX = task1;

Iy taskY = task1;

Iz taskZ = task1;

Console.WriteLine("----------------------------------------------");

var x = (Ix)task1;

x.IxF0('F');

Console.WriteLine("----------------------------------------------");

taskX.IxF0('A');

taskX.IxF1();

Console.WriteLine("----------------------------------------------");

taskY.F0('A');

taskY.F1();

Console.WriteLine("----------------------------------------------");

taskZ.F0('A');

taskZ.F1();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class TestClass : Ix, Iy, Iz

{

public char w;

public TestClass(char w)

{

this.w = w;

}

public void IxF0(char w)

{

var second = w.ToString();

var third = second.ToLower();

Console.WriteLine(third);

}

public void IxF1()

{

var second = w.ToString();

var third = second.ToLower();

Console.WriteLine(third);

}

public void F0(char w)

{

if (char.IsLetter(w))

{

Console.WriteLine("\*");

}

else

{

Console.WriteLine("не буква");

}

}

public void F1()

{

if (char.IsLetter(w))

{

Console.WriteLine("\*");

}

else

{

Console.WriteLine("не буква");

}

}

void Iz.F0(char w)

{

if (char.IsLetter(w))

{

Console.WriteLine("5");

}

else

{

Console.WriteLine("не буква");

}

}

void Iz.F1()

{

if (char.IsLetter(w))

{

Console.WriteLine("5");

}

else

{

Console.WriteLine("не буква");

}

}

}

interface Ix

{

void IxF0(char w);

void IxF1();

}

interface Iy

{

void F0(char w);

void F1();

}

interface Iz

{

void F0(char w);

void F1();

}

Таблица 9.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | f  a  f  \*  \*  5  5 |

Анализ результатов:

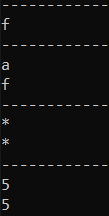


Рисунок 9.2 – Результаты работы программы

Источник: собственная разработка

Задание 2 Описать класс «поезд», содержащий следующие поля: название пункта назначения, номер поезда (может содержать буквы и цифры), время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности: вывод информации о поезде по номеру с помощью индекса, вывод информации о поездах, отправляющихся после введенного с клавиатуры времени, перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов, вывод информации о поездах, отправляющихся в заданный пункт назначения.

Листинг программы:

try

{

Station station = new Station();

Train train = new Train("Москва", "a1", new DateTime(2011, 5, 26, 20, 0, 0, 0));

Train train1 = new Train("Киев", "a2", new DateTime(2011, 5, 26, 20, 10, 0, 0));

Train train2 = new Train("Минск", "a3", new DateTime(2011, 5, 26, 20, 20, 0, 0));

Train train3 = new Train("Париж", "a4", new DateTime(2011, 5, 26, 22, 0, 0, 0));

station.AddRangeTrains(train, train1, train2, train3);

station.ThisDate(new DateTime(2011, 5, 26, 20, 0, 0, 0));

Console.WriteLine("-----------------------------");

station.Print();

station.TripFromPointName("Москва");

station.GetInformation(0);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Train : IComparable<Train>

{

private string \_pointName;

private string \_trainNumber;

private DateTime \_trainDate;

public Train(string PointName, string TrainNumber, DateTime TrainDate)

{

\_pointName = PointName;

\_trainNumber = TrainNumber;

\_trainDate = TrainDate;

}

public string PointName

{

get { return \_pointName; }

set { \_pointName = value; }

}

public string TrainNumber

{

get { return \_trainNumber; }

set { \_trainNumber = value; }

}

public DateTime TrainDate

{

get { return \_trainDate; }

set { \_trainDate = value; }

}

public static bool operator >(Train train1, Train train2)

{

return train1.TrainDate > train2.TrainDate;

}

public static bool operator <(Train train1, Train train2)

{

return train1.TrainDate < train2.TrainDate;

}

public override string ToString()

{

return $"{\_pointName},{\_trainNumber},{\_trainDate}";

}

public int CompareTo(Train? other)

{

return \_trainDate.CompareTo(other.TrainDate);

}

}

class Station

{

List<Train> trains;

public Station()

{

trains = new List<Train>();

}

public void AddTrain(Train train)

{

trains.Add(train);

}

public void AddRangeTrains(params Train[] Trains)

{

trains.AddRange(Trains);

}

public void GetInformation(int index) // получение информации по индексу

{

//var isMore = trains[0] > trains[1]; // проверка перегрузки

//Console.WriteLine(isMore);

if (index > trains.Count - 1)

{

throw new Exception("Рейса не существует");

}

Console.WriteLine("Номер номер поезда: " + trains[index].TrainNumber);

Console.WriteLine("Пункт назначения: " + trains[index].PointName);

Console.WriteLine("Время выезда: " + trains[index].TrainDate);

}

public void ThisDate(DateTime time) // получение данных о поездах, которые поедут после заданного времение

{

DateTime time1 = new DateTime(time.Year, time.Month, time.Day, time.Hour + 1,

time.Minute, time.Second, DateTimeKind.Local);

Console.WriteLine("\nИнформация о поездах, отправляющихся после определенного времени:");

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

if (trains[i].TrainDate > time)

GetInformation(i);

Console.WriteLine("Список окончен\n");

}

public void TripFromPointName(string PointName) // получение поездов, которые направляются в определенную точку

{

Console.WriteLine("В " + PointName + " едут следующие поезда: ");

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

if (trains[i].PointName == PointName)

{

GetInformation(i);

}

Console.WriteLine("Список окончен\n");

}

public void Print()

{

foreach (Train t in trains)

{

Console.WriteLine(t.ToString());

}

}

}

Таблица 9.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Информация о поездах, отправляющихся после определенного времени:  Номер номер поезда: a2  Пункт назначения: Киев  Время выезда: 26.05.2011 20:10:00  Номер номер поезда: a3  Пункт назначения: Минск  Время выезда: 26.05.2011 20:20:00  Номер номер поезда: a4  Пункт назначения: Париж  Время выезда: 26.05.2011 22:00:00  Список окончен  Москва,a1,26.05.2011 20:00:00  Киев,a2,26.05.2011 20:10:00  Минск,a3,26.05.2011 20:20:00  Париж,a4,26.05.2011 22:00:00  В Москва едут следующие поезда:  Номер номер поезда: a1  Пункт назначения: Москва  Время выезда: 26.05.2011 20:00:00  Список окончен  Номер номер поезда: a1  Пункт назначения: Москва  Время выезда: 26.05.2011 20:00:00 |

Анализ результатов:

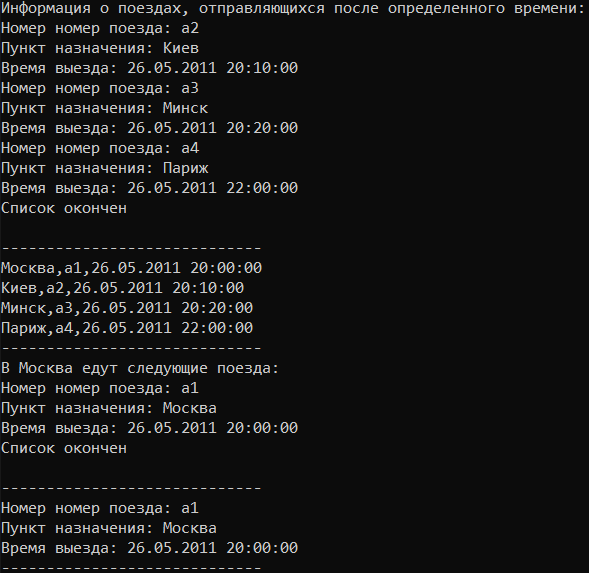


Рисунок 9.3 – Результаты работы программы

Источник: собственная разработка