**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 9**

**ИНТЕРФЕЙСЫ**

Задание 1.



Листинг программы

try

{

Test task1 = new Test(10);

Ix taskX = task1;

Iy taskY = task1;

Iz taskZ = task1;

var x = (Ix)task1;

x.IxF0(10);

taskX.IxF0(10);

taskX.IxF1();

taskY.F0(10);

taskY.F1();

taskZ.F0(10);

taskZ.F1();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Test: Ix, Iy, Iz

{

public double w;

public Test(double w)

{

this.w = w;

}

public void IxF0(double w)

{

Console.WriteLine(Math.Cos(w));

}

public void IxF1()

{

Console.WriteLine(Math.Cos(w));

}

public void F0(double w)

{

Console.WriteLine(Math.Exp(w));

}

public void F1()

{

Console.WriteLine(Math.Exp(w));

}

void Iz.F0(double w)

{

Console.WriteLine(1 / Math.Exp(w));

}

void Iz.F1()

{

Console.WriteLine(1 / Math.Exp(w));

}

}

interface Ix

{

void IxF0(double w);

void IxF1();

}

interface Iy

{

void F0(double w);

void F1();

}

interface Iz

{

void F0(double w);

void F1();

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | -0,8390715290764524  -0,8390715290764524  -0,8390715290764524  22026,465794806718  22026,465794806718  4,539992976248485E-05  4,539992976248485E-05 |

Анализ результатов:

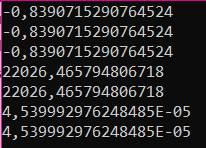


Рисунок 1.1 – Результат работы программы

Задание 2. Выполнить задания, используя для хранения экземпляров разработанных классов массивы. Во всех классах реализовать интерфейс IComparable и перегрузить операции отношения для реализации значимой семантики сравнения объектов по какому-либо полю на усмотрение студента. Описать класс «автостоянка» для хранения сведений об автомобилях. Для каждого автомобиля записываются госномер, цвет, фамилия владельца и признак присутствия на стоянке. Обеспечить возможность поиска автомобиля по разным критериям, вывода списка присутствующих и отсутствующих на стоянке автомобилей, доступа к имеющимся сведениям по номеру места. Написать программу, демонстрирующую все разработанные элементы класса.

Листинг программы:

using System;

using System.Collections.Generic;

class Program

{

static void Main(string[] args)

{

Parking park = new Parking(100);

park.AddCar("8662AB", "red", "Вася");

park.AddCar("4578CK", "green", "Петя");

park.AddCar("8683MX", "blue", "Коля");

park.AddCar("5763AA", "grey", "Дима");

park.PrintAllCar();

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

park.RemCar("8683MX");

park.AddCar("1610BM", "red", "Влад");

park.PrintAllCar();

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

park.Find("red");

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

park.Find("8662AB");

}

}

class Parking

{

private bool[] place; // количество мест на стоянке, фолз - свободно, тру - занято

private List<Car> list = new List<Car>(); // список машин

public Parking(int count)

{

place = new bool[count];

}

public void AddCar(string number, string color, string owner) // добавить машину (номер, цвет, владелец)

{

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

{

if (place[i] == false)

{

place[i] = true;

list.Add(new Car(number, color, owner, i));

return;

}

}

Console.WriteLine("Нет свободных мест");

}

public void RemCar(string number) // удалить машину (номер)

{

foreach (Car car in list)

{

if (car.number == number)

{

list.Remove(car);

place[car.order] = false;

return;

}

}

Console.WriteLine("Нет такой машины: [{0}]", number);

}

public void RemCar(int n) // удалить машину (номер места на стоянке)

{

foreach (Car car in list)

{

if (car.order == n)

{

list.Remove(car);

place[car.order] = false;

return;

}

}

Console.WriteLine("Нет такой машины: [{0}]", n);

}

public void PrintAllCar() // вывести все машины

{

foreach (Car car in list)

{

Console.WriteLine(car);

}

}

public void Find(string param) // поиск (номер/цвет/владелец)

{

Console.WriteLine("Поиск с параметром: [{0}]", param);

foreach (Car car in list)

{

if (car.color == param || car.number == param || car.owner == param)

{

Console.WriteLine(car);

}

}

}

public void Find(int place) // поиск (номер места на стоянке)

{

Console.WriteLine("Поиск с параметром: [{0}]", place);

foreach (Car car in list)

{

if (car.order == place)

{

Console.WriteLine(car);

}

}

}

private struct Car

{

public string number; // номер

public string color; // цвет

public string owner; // владелец

public int order; // номер места на стоянке

public Car(string num, string col, string own, int ord)

{

number = num;

color = col;

owner = own;

order = ord;

}

public override string ToString()

{

return string.Format("Номер:[{0}], Цвет:[{1}], Владелец:[{2}], Место:[{3}]", number, color, owner, order);

}

}

}

Анализ результатов:

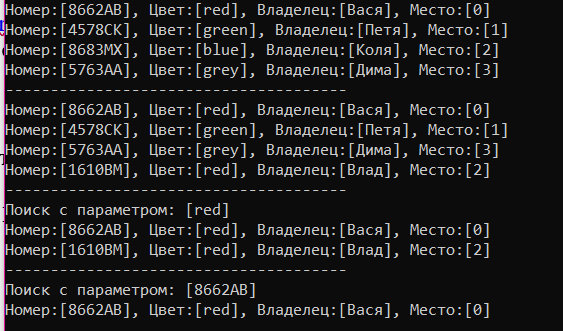


Рисунок 1.2 – Результат работы программы