**Интерфейсы**

**Задание №1.**



Листинг программы:

static void Main(string[] args)

{

TestClass x0 = new TestClass();

TestClass x1 = new TestClass(5);

x0.bF0();

x0.IxF0(10);

x1.IxF1();

x0.F0(5);

x1.F1();

(x0 as Iy).F0(7);

(x1 as Iz).F1();

WriteLine("==========Prism test==========");

WriteLine("==========Ix==========");

Ix ix = x1;

ix.IxF0(5);

ix.IxF1();

WriteLine("==========Iy==========");

Iy iy = x1;

iy.F0(5);

iy.F1();

WriteLine("==========Iz==========");

Iz iz = x1;

iz.F0(5);

iz.F1();

WriteLine("==========Iw==========");

Iw iw = x1;

iw.F0(10);

iw.F1();

ReadKey();

}

Анализ результатов:

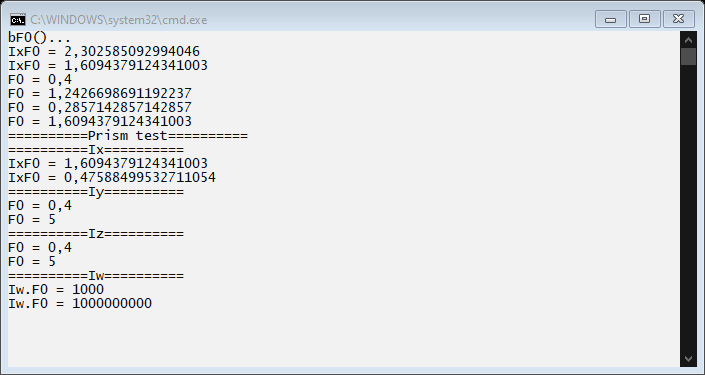


Рисунок 1.1 – Результат работы программы

**Задание №2**. Описать класс «товар», содержащий следующие закрытые поля: название товара; название магазина, в котором продается товар; стоимость товара в рублях. Предусмотреть свойства для получения состояния объекта. Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности: вывод информации о товаре по номеру с помощью индекса; сортировку товаров по названию магазина, по наименованию и по цене; Написать программу, демонстрирующую все разработанные элементы классов.

Листинг программы:

static void Main(string[] args)

{

try

{

const int MINVALUE = 1;

const int MAXVALUE = 100;

Write("Кол-во продуктов: ");

var count = ToInt32(ReadLine());

var wareHouse = new Warehouse();

var myList = new List<Product>();

for (int i = 0; i < count; i++)

{

WriteLine("Название товара:");

string productName = ReadLine();

string shopName = Faker.Company.Name();

double cost = Math.Round(Faker.RandomNumber.Next(MINVALUE, MAXVALUE) \* 0.1, 2);

int id = Faker.RandomNumber.Next(MINVALUE, MAXVALUE);

myList = wareHouse.MakeProductList(new Product(productName, shopName, cost, id));

}

var sorted = wareHouse.SortByProductCost(myList);

wareHouse.WriteThis(sorted);

WriteLine("Введите id продукта: ");

var index = ToInt32(ReadLine());

var findedByIndex = wareHouse.GetProductByIndex(myList, index);

if (findedByIndex != null)

{

wareHouse.WriteThis(findedByIndex);

}

else

{

throw new NullReferenceException("Продукт с введённым ID не существует.");

}

Write("Введите id сравниваемых продуктов:\n" +

"id1 = ");

var firstId = ToInt32(ReadLine());

Write("Введите id сравниваемых продуктов:\n" +

"id2 = ");

var secId = ToInt32(ReadLine());

var firstFinded = wareHouse.GetProductByIndex(myList, firstId);

var secFinded = wareHouse.GetProductByIndex(myList, secId);

if (firstFinded == null || secFinded == null)

{

throw new NullReferenceException("Один или несколько исхомых продуктов не существуют.");

}

else

{

WriteLine($"\n{firstFinded.CompareTo(secFinded)}");

}

}

catch (FormatException)

{

WriteLine("Неверный формат!");

}

catch(NullReferenceException ex)

{

WriteLine(ex.Message);

}

ReadKey();

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Молоко, кефир, молоко, творог | -1 |

Анализ результатов:

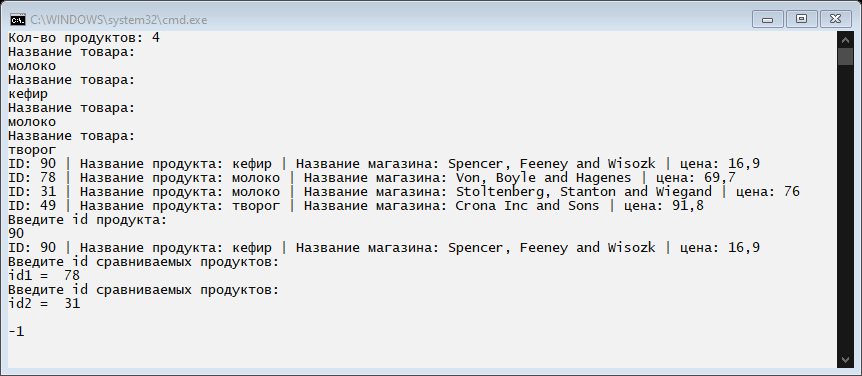


Рисунок 1.2 – Результат работы программы