

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork

{

static class Exension

{

static public void AddElem<T, R>(this SortedList arr, T[] m, R[] s)

{

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

{

arr.Add(m[i],s[i]);

}

}

static public void Print(this SortedList a)

{

foreach (**DictionaryEntry** VARIABLE in a)

{

Console.WriteLine(VARIABLE.Key+" "+VARIABLE.Value);

}

}

}

class Program

{

class MyClass : IComparer

{

public int Compare(object x, object y)

{

return y.GetHashCode()-x.GetHashCode();

}

}

static void Main(string[] args)

{

int[] key = {2,33,44,53,23,6,8};

string[] value = {"vit","bbe","sfd","kol","rew","pot","gtl"};

SortedList sl = new SortedList(new MyClass());

sl.AddElem(key,value);

sl.Print();

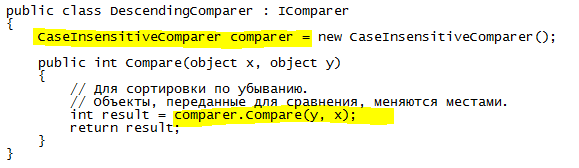
Console.ReadKey();

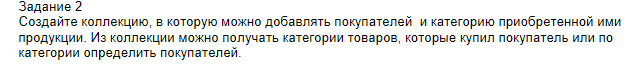
}

}

}

Можно было применить





using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace HomeWork

{

class KeysSensative:IEqualityComparer<string>

{

CaseInsensitiveComparer comparer=new CaseInsensitiveComparer();

public bool Equals(string x, string y)

{

return comparer.Compare(x, y)==0;

}

public int GetHashCode(string obj)

{

return obj.ToLowerInvariant().GetHashCode();

}

}

static class Exension

{

static public void AddElem<T, R>(this IDictionary arr, T[] m, R[] s)

{

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

{

arr.Add(m[i], s[i%2]);

}

}

static public void Print(this IDictionary a)

{

foreach (**DictionaryEntry** VARIABLE in a)

{

Console.WriteLine(VARIABLE.Key+" "+VARIABLE.Value);

}

}

}

enum Products

{

IT,Telephone

}

class Program

{

static void Main(string[] args)

{

Dictionary<string, Products> dic = new Dictionary<string, Products>(new KeysSensative());

string[] name = {"vit","bbe","sfd","kol","rew","pot","gtl","Vit"};

try

{

dic.AddElem(name, new Products[]{Products.IT,Products.Telephone});

}

catch (ArgumentException)

{

Console.WriteLine("Arrgumetn be added");

}

dic.Print();

Console.ReadKey();

}

}

}

