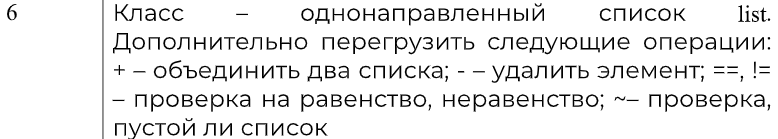
**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**



**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**

using System;

using System.Collections.Generic;

using System.Data.Common;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Laba19\_2;

internal class Spisok<T>

{

List<T> list;

public Spisok() { list = new List<T>(); }

public Spisok(List<T> list)

{

this.list = list;

}

public void Add(T item) => list.Add(item);

public void Clear() { list.Clear(); }

public bool Contains(T item) => list.Contains(item);

public int IndexOf(T item) => list.IndexOf(item);

public void Insert(int index, T item) { list.Insert(index, item); }

public void RemoveAt(int index) { list.RemoveAt(index); }

public bool Remove(T item) => list.Remove(item);

public int Count => list.Count;

public static Spisok<T> operator +(Spisok<T> spisok1, Spisok<T> spisok2)

{

//return (Spisok<T>)(spisok1.list).Concat(spisok2.list);

Spisok<T> spisok3 = new Spisok<T>();

for (int i = 0; i < spisok1.Count; i++)

{

spisok3.Add(spisok1[i]);

}

for (int i = 0; i < spisok2.Count; i++)

{

spisok3.Add(spisok2[i]);

}

return spisok3;

}

public static Spisok<T> operator -(Spisok<T> spisok, T element)

{

foreach (T item in spisok.list)

if (object.Equals(item, element))

{

spisok.list.Remove(item);

return spisok;

}

Console.WriteLine("Элемент не найден! Список не изменён!");

return spisok;

}

public static bool operator ==(Spisok<T> spisok1, Spisok<T> spisok2)

{

return spisok1.list.SequenceEqual(spisok2.list);

}

public static bool operator !=(Spisok<T> spisok1, Spisok<T> spisok2)

{

return !spisok1.list.SequenceEqual(spisok2.list);

}

public static bool operator ~(Spisok<T> spisok)

{

return spisok.Count == 0;

}

public T this[int i]

{

get { return list[i]; }

}

}

**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**

using static System.Math;

using static System.Console;

using static System.Convert;

namespace Laba19\_2;

class Program

{

static void task1()

{

Spisok<int> spisok1 = new Spisok<int>();

Spisok<int> spisok2 = new Spisok<int>();

spisok1.Add(15);

spisok1.Add(30);

spisok2.Add(1);

spisok2.Add(2);

for (int i = 0; i < spisok1.Count; i++)

{

Console.WriteLine($"{spisok1[i]} : {spisok2[i]}");

}

Spisok<int> spisok3 = new Spisok<int>();

spisok3 = spisok1 + spisok2;

Console.WriteLine("//////");

for (int i = 0; i < spisok3.Count; i++)

{

Console.WriteLine($"{spisok3[i]}");

}

WriteLine("Равность: "+(spisok1 == spisok2));

Spisok<int> spisok4 = new Spisok<int>();

spisok4.Add(15);

spisok4.Add(30);

WriteLine("Равность: " + (spisok1 == spisok4));

Spisok<int> spisok5 = new Spisok<int>();

if(~spisok5)

WriteLine("Список пуст!");

}

static void task2()

{

Spisok<string> spisok1 = new Spisok<string>();

Spisok<string> spisok2 = new Spisok<string>();

spisok1.Add("aboba");

spisok1.Add("lola");

spisok2.Add("a");

spisok2.Add("b");

for (int i = 0; i < spisok1.Count; i++)

{

Console.WriteLine($"{spisok1[i]} : {spisok2[i]}");

}

Spisok<string> spisok3 = new Spisok<string>();

spisok3 = spisok1 + spisok2;

Console.WriteLine("//////");

for (int i = 0; i < spisok3.Count; i++)

{

Console.WriteLine($"{spisok3[i]}");

}

WriteLine(spisok1 == spisok2);

Spisok<string> spisok4 = new Spisok<string>();

spisok4.Add("aboba");

spisok4.Add("lola");

WriteLine(spisok1 == spisok4);

Spisok<string> spisok5 = new Spisok<string>();

if(~spisok5)

WriteLine("Список пуст!");

}

static void Main(string[] args)

{

bool menu = true;

while (menu)

{

try

{

Write("1\n 2\nВыберите номер задания (0 для выхода из программы): ");

int sw = ToInt32(ReadLine());

switch (sw)

{

case 0:

{

menu = false;

break;

}

case 1:

{

try

{

task1();

Console.WriteLine("---");

task2();

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

}

WriteLine();

WriteLine();

}

catch (Exception ex)

{

WriteLine(ex.Message);

} } }}