ОБОБЩЕНИЯ

Задание №1. Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте класс MyList&lt;T&gt;. Реализуйте возможность использования его экземпляра аналогично экземпляру класса List&lt;T&gt;. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления элемента, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества элементов.

Листинг программы:

using System;

using System.Collections.Generic;

namespace MyNamespace

{

class Program

{

static void Main(string[] args)

{

MyList<int> myList = new MyList<int>();

myList.Add(1);

myList.Add(2);

myList.Add(3);

int value = myList[1];

Console.WriteLine(value);

int count = myList.Count;

Console.WriteLine(count);

}

}

class MyList<T>

{

private List<T> list;

public MyList()

{

list = new List<T>();

}

public void Add(T item)

{

list.Add(item);

}

public T this[int index]

{

get { return list[index]; }

}

public int Count

{

get { return list.Count; }

}

}

}

Анализ программы:



Рисунок 1.1 - Анализ программы

Источник: собственная разработка

Задание №2. Используя Visual Studio, создайте проект по шаблону Console Application. Создайте класс MyDictionary &lt;TKey, TValue&gt;. Реализуйте возможность использования его экземпляра аналогично экземпляру класса Dictionary. Минимально требуемый интерфейс взаимодействия с экземпляром, должен включать метод добавления пар элементов, индексатор для получения значения элемента по указанному индексу и свойство только для чтения для получения общего количества пар элементов.

Листинг программы:

using System;

using System.Collections.Generic;

namespace MyNamespace

{

class Program

{

static void Main(string[] args)

{

MyDictionary<int, string> myDict = new MyDictionary<int, string>();

myDict.Add(1, "One");

myDict.Add(2, "Two");

myDict.Add(3, "Three");

Console.WriteLine(myDict[1]);

Console.WriteLine(myDict[2]);

Console.WriteLine(myDict[3]);

Console.WriteLine(myDict.Count);

}

class MyDictionary<TKey, TValue>

{

private Dictionary<TKey, TValue> dict = new Dictionary<TKey, TValue>();

public void Add(TKey key, TValue value)

{

dict.Add(key, value);

}

public TValue this[TKey key]

{

get

{

return dict[key];

}

set

{

dict[key] = value;

}

}

public int Count

{

get

{

return dict.Count;

}

}

}

}

}

Анализ программы:



Рисунок 1.2 - Анализ программы

Источник: собственная разработка