Урок номер 1.

**Классная работа.**

**1.**

Создайте класс **MyClass**, в котором создайте автосвойство типа **int**. Создайте класс **MyCollection<T>,** реализующий основной функционал коллекции элементов. При создании **MyCollection<T>** используйте наследование и реализацию **IEnumerable<T>, IEnumerator<T>.** В методе Main наполните экземпляр коллекции элементами **MyClass**, выполните дважды обход этой коллекции с выводом на экран значений автосвойства элементов коллекции.

**2.**

Выполните предыдущую задачу, используя оператор **yield**

**3.**

Выполните задачу 1, реализуя классом **MyCollection** интерфейс **ICollection<T>.**

**4.**

**Домашняя работа.**

**1.**

Выполните классную задачу 1 реализуя интерфейс IList<T>.

**2.**

а. Создайте класс **MyCollection,** реализуяэтим классом интерфейсы IList<T> и IList. Реализуемый интерфейс IList<T> должен быть закрыт типом string в месте объявления класса. В данной коллекции создайте массив элементов. Коллекцию реализуйте так, чтобы она работала потокобезопасно.

б. Создайте класс **Handler** в котором создайте не статический метод **Count**. Задача метода **Count** состоит в следующем:

* Метод должен принять экземпляр класса **MyCollection** и в цикле foreach вывести на экран все элементы коллекции **MyCollection** + номер потока,в котором метод выполняется. Вывод каждого из этих значений должен происходить с интервалом в 100 миллисекунд. Через каждые 10 выводимых значений, метод должен вывести на экран консоли фразу «Номер текущего потока» и сам номер. После вывода этой фразы, перед переходом на следующую итерацию, должна быть задержка в 2 секунды.
* После завершения работы цикла foreach в методе **Count** должно бросаться исключение **Exception** с сообщением внутри о номере потока, в котором оно возникло.
* В методе **Count** область с циклом foreach и вызовом исключения должна быть заключена в критическую секцию, обеспечивая доступ к ней только одному потоку.

в. В классе **Handler** создайте событие ExceptionEvent, задача которого – среагировать на возникшее исключение в методе **Count.** Создайте механизмы вызова этого события.

Создайте метод-обработчик события, который будет принимать экземпляр полученного **Exception** и выводить на экран его сообщение.

г. В методе Main создайте экземпляр **MyCollection** и заполните его в цикле на 100 итераций строковыми значениями (можно просто строковыми значениями переменной итерации).

Создайте два потока, сообщите с ними метод **Count,** и запустите потоки в работу, отправив в них экземпляр **MyCollection**, с задержкой в полсекунды.

Внешний вид консоли после завершения работы программы:

