1. На какие основные виды/типы делятся все коллекции .NET? Охарактеризуйте каждый из них.

* **необобщенные**

- хранение данных любого типа, причем в одной коллекции допускается наличие разнотипных данных.

-динам. массив, стек, очередь, словари, кот. хранят пары «ключ-значение»

- такие коллекции не типизированы, тк в них хранятся ссылки на данные типа object

(не обеспечивают типовую безопасность)

- System.Collections – там классы и интерфейсы

* **обобщенные**

- типизированные, обеспечивают типовую безопасность;

-связные списки, стеки, очереди и словари.

- System.Collections.Generic

* **специальные**

- System.Collections.Specialized

- оперируют данными конкретного типа или же делают это каким-то особым образом. Например, имеются специальные коллекции для символьных строк, а также специальные коллекции, в которых используется однонаправленный список.

* **поразрядные**

- Коллекция типа BitArray в System.Collections

- операции над двоичными разрядами (И,ИЛИ, исключающее или)

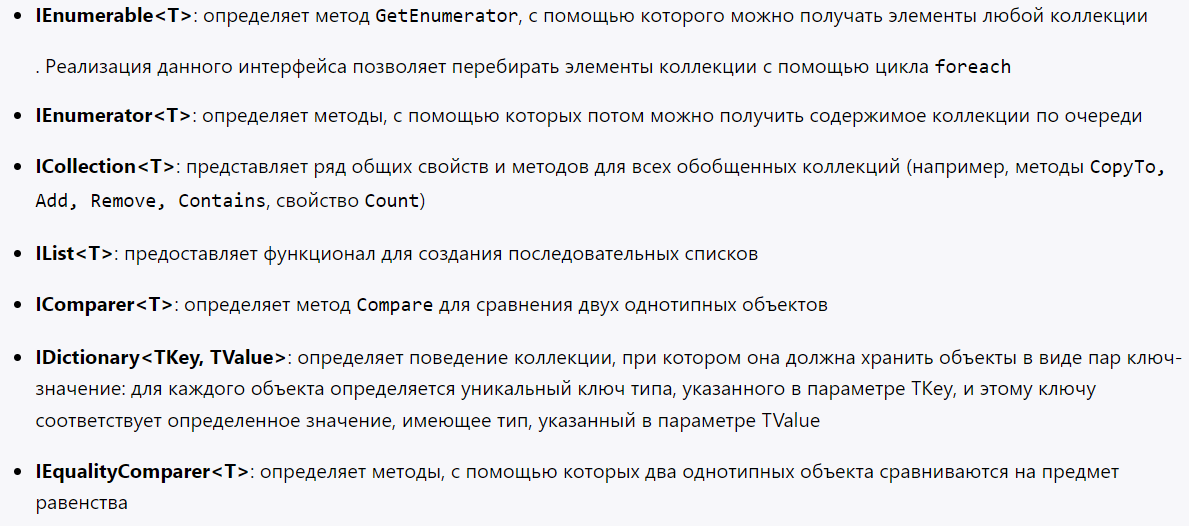
* **параллельные**

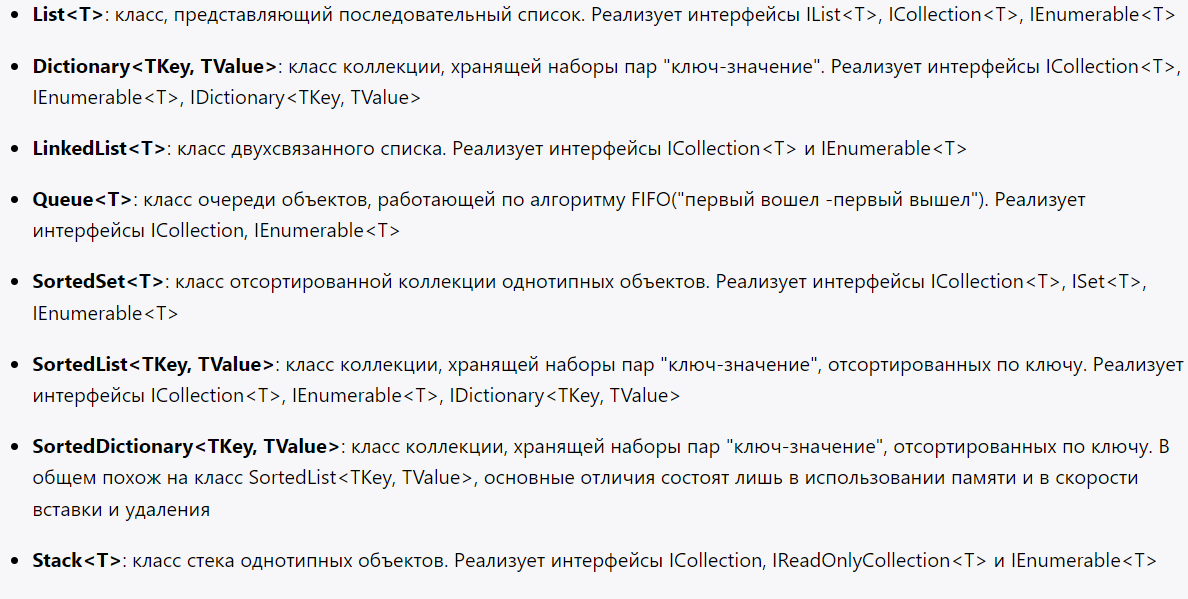
- поддерживает многопоточный доступ к коллекции

- Это обобщенные коллекции, определенные в System.Collections.Concurrent

2. Что такое generic-коллекции? Назовите примеры известных вам generic-коллекций.

Generic-коллекции – это те же самые обобщенные классы. Их использование перед необобщёнными коллекциями имеет те же преимущества: повышение производительности (не надо тратить время на упаковку и распаковку объекта) и повышенная типобезопасность.





3. В чем разница между ArrayList и Array?

ArrayList - определяется массив переменной длины, который состоит из ссылок на объекты и может динамически увеличивать и уменьшать свой размер. Это список, каждый элемент связан друг с другом через указатели.

Array – массив, имеющий фиксированный размер.

4. Охарактеризуйте коллекции, которые вы использовали в своем варианте.

List<T> - создает динамический массив, строго типизированный список объектов, доступных по индексу. Эквивалент ArrayList

SortedList<TKey, TValue> - создает отсортированный список по ключу из пар ключ-значение. Ключи реализовывают IComparable<T>, IComparer<T> не дублир. Массив.

Dictionary <Tkey, TValue> - сохраняет пары ключ-значение, как хэш-таблица.

5. Чем отличаются коллекции, расположенные в пространстве имен System.Collections.Concurrent?

- коллекции классов, предназначенные для безопасной работы в многопоточной среде, которыми можно воспользоваться при создании многопоточных приложений

6. Какое пространство имен необходимо подключить в проект, чтобы иметь возможность использовать generic-коллекции?

System.Collections.Generic

7. Что такое наблюдаемая коллекция? Как ее можно использовать?

**ObservableCollection<T>**

- пользовательский интерфейс получает информацию об изменениях коллекции

- унаследован от Collection<T>, использует внутри себя List<T>, INotifyCollectionChanged

8. Охарактеризуйте интерфейсы IEnumerator, IEnumerable. В чем отличие назначений интерфейсов IEnumerator и IEnumerable.

IEnumerable – интерфейс, в нем один метод GetEnumerator() интерфейса IEnumerator. Перечислитель, с помощью которого становится возможен последовательный перебор коллекции. Для foreach.

IEnumerator – простой перебор: имеет свойство Current и методы MoveNext и Reset.

Позволяет перебирать элементы коллекции.

9. Поясните принцип работы коллекций:

a. LinkedList <T> - сохраняет элементы в двунаправленном списке, каждый элемент хранит ссылку на следующий и на предыдущий элемент. ICollection

b. HashSet <T> - сохраняет ряд уникальных значений, использует хэш-таблицу

c. Dictionary <Tkey, TValue> - сохраняет пары ключ-значение, как хэш-таблица.

d. ConcurrentBag <Tkey, TValue> - реализует концепцию отображения потоков на используемые внутренне массивы. Не определяет порядок добавления или извлечения элементов.

e. Stack<T>, Queue<T> - создаёт стек LIFO, создаёт очередь FIFO

f. SortedList – хранит наборы пар ключ-значение, отсортированные по ключу

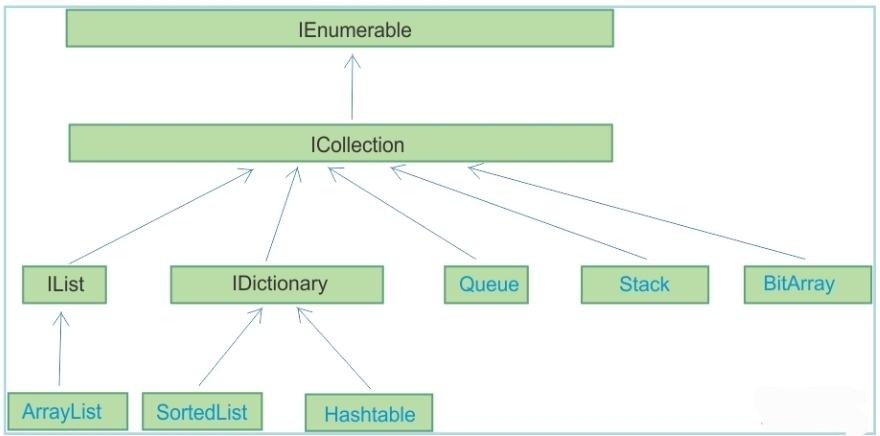
List<T> - создает динамический массив, строго типизированный список объектов, доступных по индексу. Эквивалент ArrayList

SortedDictionary<TKey, TValue> - создает отсортированный список из пар ключ-значение, упорядоченных по ключу. Вставка медленнее, извлечение быстрее, использует больше памяти чем ниже челик. Это двоичное дерево поиска.

SortedList<TKey, TValue> - создает отсортированный список из пар ключ-значение. Ключи реализовывают IComparable<T>, IComparer<T> не дублир. Массив.

SortedSet<T> - создает отсортированное множество, не дублир., не влияет на производительность.

Классы необобщенных:



Классы обобщенных:

