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

* **Необобщенные коллекции:** это коллекции, которые хранят элементы типа Object и требуют приведения типов при добавлении или извлечении элементов. Они находятся в пространстве имен System.Collections и включают такие классы, как ArrayList, Hashtable, Queue, Stack и т.д. Они не являются строго типизированными и обычно менее производительны, чем обобщенные коллекции.обобщенные
* **Специальные коллекции:** это коллекции, которые предназначены для определенных целей, таких как хранение строк, битов или пар ключ-значение. Они находятся в пространстве имен System.Collections.Specialized и включают такие классы, как StringCollection, BitArray, NameValueCollection и т.д. Они могут быть обобщенными или необобщенными в зависимости от типа элементов.
* **Коллекции с поразрядной организацией**: это коллекции, которые используют битовые операции для хранения и манипулирования данными. Они находятся в пространстве имен System.Collections.BitArray и включают такие классы, как BitArray, BitVector32 и т.д. Они обеспечивают компактное и эффективное представление булевых значений, целых чисел или битовых полей.
* **Обобщенные коллекции**: это коллекции, которые принимают параметр типа при создании и хранят элементы одного типа. Они находятся в пространстве имен System.Collections.Generic и включают такие классы, как List<T>, Dictionary<TKey,TValue>, Queue<T>, Stack<T> и т.д. Они являются строго типизированными и обычно обеспечивают более высокую производительность, чем необобщенные коллекции.
* **Параллельные коллекции**: это коллекции, которые обеспечивают потокобезопасный доступ к элементам из нескольких потоков. Они находятся в пространстве имен System.Collections.Concurrent и включают такие классы, как ConcurrentDictionary<TKey,TValue>, ConcurrentQueue<T>, ConcurrentStack<T> и т.д. Они используют синхронизацию и блокировку для предотвращения состояния гонки и повреждения данных1.

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

Обобщённые коллекции – это те же самые обобщённые классы(классы, обладающие параметрическим полиморфизмом, т.е. классы, которые изменяют своё поведение в зависимости от приписываемого им типа). Их использование перед необобщёнными коллекциями имеет те же преимущества: повышение производительности 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> - упорядоченная коллекция, список. Заполняется по порядку.

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

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.

Эти два интерфейса позволяют перебирать элементы коллекций.

IEnumerable имеет один метод GetEnumerator и возвращает объект интерфейса IEnumerator.

IEnumerator возвращает текущий элемент коллекции и имеет два метода MoveNext и Reset (для перемещения).

Назначение интерфейса IEnumerable – предоставить перечислитель для коллекции, то есть объект, который умеет перебирать элементы.

Назначение IEnumerator – реализация логики перебора элементов, т.е. определить, как получить текущий элемент и идти дальше.

9. Поясните принцип работы коллекций:

a. LinkedList <T> - сохраняет элементы в двунаправленном списке

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

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

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, создаёт очередь

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