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

► необобщенные: наличие разнотипных данных ,ссылки на данные типа object (не обеспечивают типовую безопасность) ,System.Collections   
► обобщенные: обеспечивают типовую безопасность ,System.Collections.Generic   
► специальные :System.Collections.Specialized   
► с поразрядной организацией :BitArray   
► параллельные: многопоточный доступ к коллекции ,System.Collections.Concurrent

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

Коллекция – совокупность объектов.Стандартизируют обработку групп объектов в проге.  
  
 3. В чем разница между ArrayList и Array?

Основное различие между Array и ArrayList состоит в том, что Array является структурой данных фиксированной длины, а ArrayList является классом Collection переменной длины. Вкратце, Array имеет статическую природу, тогда как ArrayList имеет динамическую природу.

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

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

Даже с использованием примитивов синхронизации, предоставляемых C # и .NET Framework, написание кода синхронизации является достаточно сложным. Оба класса-оболочки которые описаны выше реализуют синхронизацию только для небольшого подмножества операций.

Параллельные коллекции в пространстве имен System.Collections.Concurrent покрывают все кейсы. Они предоставляют поточно-ориентированные реализации интерфейсов коллекции.

Исключениями являются классы ConcurrentQueue <T> и ConcurrentStack <T>, которые предоставляют независимые реализации, аналогичные их непараллельным аналогам Queue <T> и Stack <T>, поскольку в .NET Framework нет подходящих интерфейсов для реализации.

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

System.Collections   
System.Collections.Generic   
System.Collections.Specialized   
BitArray   
System.Collections.Concurrent

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

ObservableCollection. Он по функциональности похож на список List за тем исключением, что позволяет известить внешние объекты о том, что коллекция была изменена.

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

Интерфейс IEnumerable имеет метод, возвращающий ссылку на другой интерфейс - перечислитель: для foreach

public interface IEnumerable

{

    IEnumerator GetEnumerator();

}

А интерфейс IEnumerator определяет функционал для перебора внутренних объектов в контейнере:

public interface IEnumerator

{

    bool MoveNext(); // перемещение на одну позицию вперед в контейнере элементов

    object Current {get;}  // текущий элемент в контейнере

    void Reset(); // перемещение в начало контейнера

}

9. Поясните принцип работы коллекций:   
a. LinkedList сохр. эл-ты в двунаправленном списке  
b. HashSet -- сохр. ряд уникальных значений, исп. хэштаблицу  
c. Dictionary - сохр. пары «ключ-значение»  
d. ConcurrentBag - является потокобезопасной реализацией контейнера, оптимизированной для сценариев, в которых один и тот же поток будет создавать и потреблять данные, хранящиеся в контейнере.  
e. Stack, Queue -созд. Стек/Очередь  
f. SortedList, SortedList.- созд. отсортированное мн-во, хран. наборы пар «ключ-значение», отсорт. по ключу