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.

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.

IEnumerable – это интерфейс который определяет один метод GetEnumerator, который возвращает IEnumerator, что в свою очередь позволяет читать только доступ к коллекции.

Коллекция, которая реализует IEnumerable, может использоваться с оператором foreach.

IEnumerator – вещь, которая может перечислять: имеет свойство Current и методы MoveNext и Reset.

IEnumerable – это вещь, которую можно перечислить. У неё есть метод GetEnumerator, который возвращает IEnumerator.

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