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

**Необобщенные коллекции**

Реализуют ряд основных структур данных, включая динамический массив, стек, очередь, а также словари, в которых можно хранить пары "ключ-значение". В отношении необобщенных коллекций важно иметь в виду следующее: они оперируют данными типа object. Таким образом, необобщенные коллекции могут служить для хранения данных любого типа, причем в одной коллекции допускается наличие разнотипных данных. Очевидно, что такие коллекции не типизированы, поскольку в них хранятся ссылки на данные типа object. Классы и интерфейсы необобщенных коллекций находятся в пространстве имен **System.Collections**.

**Специальные коллекции**

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. Специальные коллекции объявляются в пространстве имен **System.Collections.Specialized**.

**Поразрядная коллекция**

В прикладном интерфейсе Collections API определена одна коллекция с поразрядной организацией — это BitArray. Коллекция типа BitArray поддерживает поразрядные операции, т.е. операции над отдельными двоичными разрядами, например И, ИЛИ, исключающее ИЛИ, а следовательно, она существенно отличается своими возможностями от остальных типов коллекций. Коллекция типа BitArray объявляется в пространстве имен System.Collections.

**Обобщенные коллекции**

Обеспечивают обобщенную реализацию нескольких стандартных структур данных, включая связные списки, стеки, очереди и словари. Такие коллекции являются типизированными в силу их обобщенного характера. Это означает, что в обобщенной коллекции могут храниться только такие элементы данных, которые совместимы по типу с данной коллекцией. Благодаря этому исключается случайное несовпадение типов. Обобщенные коллекции объявляются в пространстве имен **System.Collections.Generic**.

**Параллельные коллекции**

Поддерживают многопоточный доступ к коллекции. Это обобщенные коллекции, определенные в пространстве имен **System.Collections.Concurrent**.

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

Обеспечивают обобщенную реализацию нескольких стандартных структур данных, включая связные списки, стеки, очереди и словари. Такие коллекции являются типизированными в силу их обобщенного характера. Это означает, что в обобщенной коллекции могут храниться только такие элементы данных, которые совместимы по типу с данной коллекцией. Благодаря этому исключается случайное несовпадение типов. Обобщенные коллекции объявляются в пространстве имен **System.Collections.Generic**.

3. В чем разница между ArrayList и Array?

Разница между **массивом** и **ArrayList** заключается в следующем:

1. Реализация массива - это простой массив фиксированного размера, **но** реализация ArrayList-это массив динамического размера.
2. Массив может содержать как примитивы, так и объекты, **но** ArrayList может содержать только элементы объекта
3. Вы не можете использовать дженерики вместе с массивом, **но** ArrayList позволяет нам использовать дженерики для обеспечения безопасности типов.
4. Вы можете использовать \*length \*variable для вычисления длины массива, **но** метод size() для вычисления размера ArrayList.
5. Массив использует оператор присваивания для хранения элементов, **но** ArrayList использует \*add() \*to вставляет элементы.

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

**Параллельные коллекции**

Поддерживают многопоточный доступ к коллекции. Это обобщенные коллекции, определенные в пространстве имен **System.Collections.Concurrent**.

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

Поддерживают многопоточный доступ к коллекции. Это обобщенные коллекции, определенные в пространстве имен **System.Collections.Concurrent**.

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?

Коллекция которая извещает о добавлении и удалении элементов.

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

Интерфейс IEnumerable имеет метод, возвращающий ссылку на другой интерфейс – перечислитель

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

9. Поясните принцип работы коллекций:

a. LinkedList Сохраняет элементы в двунаправленном списке

b. HashSet Сохраняет ряд уникальных значений, используя хештаблицу

c. Dictionary Сохраняет пары "ключ-значение". Обеспечивает такие же функциональные возможности, как и необобщенный класс Hashtabled. ConcurrentBag

Этот класс не определяет никакого порядка для добавления или извлечения элементов. Он реализует концепцию отображения потоков на используемые внутренне массивы, и старается избежать блокировок.

e. Stack, Queue Создает очередь. Обеспечивает такие же функциональные возможности, как и необобщенный класс Queue

Создает стек. Обеспечивает такие же функциональные возможности, как и необобщенный класс Stack

f. SortedList, SortedList.

Создает отсортированный список из пар "ключ-значение". Обеспечивает такие же функциональные возможности, как и необобщенный класс SortedList

Создает отсортированное множество