|  |  |  |
| --- | --- | --- |
|  | Основная функциональность | Примеры типичного использования |
| Set | Коллекция моделирует математическое множество. Т.е в реализации интерфейса Set не допускаются элементы-дубликаты. Множество может содержать по крайней мере одно значение null  Все имплементации основаны на Мар.  Имплементации: HashSet (упорядочивает элементы на основе хеш-функции), LinkedHashSet (подкласс HashSet, гарантирует порядок вставки), TreeSet (отсортированное мнежество, используется Comparator или метод compareTo) | Set часто используются для проверки принадлежности объекта заданному множеству. На практике обычно выбирается реализация HashSet, оптимизированная для быстрого поиска.  Также используется в случаях, когда есть необходимость хранить данные не допускающие повторений. Например, для хранения перечня имен пользователей какой-либо системы |
| List | Эта коллекция хранит элементы в определенном порядке, а также может содержать их дубликаты. Элементы в списке упорядочены, за каждым из них закреплена позиция. К элементу можно обратится по индексу. Позиция может изменяться в результате добавления или удаления элементов  Имплементации: LinkedList (использует двусвязный список), ArrayList и Vector (реализованы на основе массивов) | Списки используются для хранения разного рода данных, которые допускают дублирование. Например, для создания базы данных в магазине |
| Queue | Этот интерфейс описывает коллекции с предопределённым способом вставки и извлечения элементов, а именно — очереди FIFO (first-in-first-out). Помимо методов, определённых в интерфейсе Collection, определяет дополнительные методы для извлечения и добавления элементов в очередь  Имплементации: LinkedList, PriorityQueue (особенностью данной очереди является возможность управления порядком элементов), ArrayDeque (реализация интерфейса Deque, который расширяет интерфейс Queue методами, позволяющими реализовать конструкцию вида LIFO (last-in-first-out). Эта коллекция представляет собой реализацию с использованием массивов, подобно ArrayList, но не позволяет обращаться к элементам по индексу и хранение null) | Используются для реализации набора данных, для которых важна очередность. Например, для учета проведения вакцинации |
| Map | Определяет отображение ключей и значений. Ключи в карте уникальны. Каждый ключ отвечает одному значению, но одному значению могут отвечать несколько ключей. Ключи и значения – объекты  Имплементации: HashMap (упорядочивает ключи на основе хеш-функции), LinkedHashMap (гарантирует порядок вставки), TreeMap (карта, отсортированная по значениям ключей, с использованием Comparator или метода compareTo), Hashtable (структура данных реализуется как хэш-таблица, она не позволяет использовать null в качестве значения или ключа) | Используется для отображения одних объектов на другие. Например, в карте студент-оценка, где оценка(значение) может быть сопоставлена со многими студентами(ключи), но каждый студент имеет только одну оценку. |