|  |  |  |
| --- | --- | --- |
|  | Основная функциональность | Примеры типичного использования |
| Set | Набор уникальных значений (множество). | Множества часто используются для проверки принадлежности, чтобы можно было легко проверить, принадлежит ли объект заданному множеству, поэтому обычно выбирается реализация HashSet, оптимизированная для быстрого поиска. |
| List | Списки позволяют хранить любые наборы значений, в том числе повторяющиеся и нулевые. Хороши тем, что количество элементов не фиксировано. Хотя и менее производительны, чем массив. | Списки индексированы, потому индекс элемента неявно связан с его значением. Удобно использовать для решения различных вычислительных задач или когда просто нужно в любой момент быстро получить необходимый элемент из центра списка. |
| Queue | Предоставление доступа к списку в порядке очереди. | Неблокирующие очереди заточены на скорость и работу без блокирования потоков. Блокирующие очереди используются, когда нужно «притормозить» потоки «Producer» или «Consumer», если не выполнены какие-либо условия, например, очередь пуста или перепонена, или же нет свободного «Consumer»'a. Пример – рассылка электронной почты. |
| Map | Набор уникальных ключей с соответствующими им значениями. | Функциональность, в принципе, отражает и назначение. Когда нужно по ключу быстро найти значение, используют мэп. К примеру, поиск информации о работнике. Или имени пользователя по уникальному коду (обычно это hashcode). |