Сравним коллекции Java по следующим параметрам:

1. **На чем основана** — какая структура данных или коллекция используется под капотом.
2. **Дубли** — разрешены или нет в коллекции повторяющиеся значения.
3. **Null** — позволяет ли коллекция вставлять null.
4. **Синхронизированность** — все ли методы коллекции синхронизированы.
5. **Потокобезопасность** — безопасно ли использовать коллекцию в многопоточной среде.
6. **Тип итератора** — поведение итератора в многопоточной среде.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Базовый интерфейс** | **Тип коллекции** | **На чем основана** | **Дубли** | **Null** | **Синхрони- зированность** | **Потоко- безопасность** | **Тип итератора** |
| List | ArrayList<E> | Массив | Да | Да | Нет | Нет | Быстрое падение |
| LinkedList<E> | Двусвязный список | Нет | Нет | Быстрое падение |
| CopyOnWriteArrayList<E> | Массив | Нет | Да | Моментальный снимок |
| Vector<E> | Массив | Да | Да | Быстрое падение |
| Stack<E> | Массив | Да | Да | Быстрое падение |
| Queue | ConcurrentLinkedQueue<E> | Односвязный список | Да | Нет | Нет | Да | Слабая согласованность |
| PriorityQueue<E> | Сбалансированная двоичная куча, основанная на массиве | Нет | Нет | Быстрое падение |
| LinkedBlockingQueue<E> | Односвязный список | Нет | Да | Слабая согласованность |
| PriorityBlockingQueue<E> | Сбалансированная двоичная куча, основанная на массиве | Нет | Да | Моментальный снимок |
| ArrayBlockingQueue<E> | Массив фиксированного размера | Нет | Да | Слабая согласованность |
| DelayQueue<E> | PriorityQueue<E> | Нет | Нет | Моментальный снимок |
| SynchronousQueue<E> | Ничего не хранит, нулевая емкость | Нет | Да | Collections.emptyIterator |
| LinkedTransferQueue<E> | Односвязный список | Нет | Да | Слабая согласованность |
| ArrayDeque<E> | Массив | Нет | Нет | Быстрое падение |
| ConcurrentLinkedDeque<E> | Двусвязный список | Нет | Да | Слабая согласованность |
| LinkedBlockingDeque<E> | Двусвязный список | Нет | Да | Слабая согласованность |
| Map | HashMap<K,V> | Хеш-таблица | Для значений | Для ключей и значений | Нет | Нет | Быстрое падение |
| LinkedHashMap<K,V> | Хеш-таблица + двусвязный список | Для ключей и значений | Нет | Нет | Быстрое падение |
| TreeMap<K,V> | Красно-чёрное дерево | Нельзя использовать null в ключах, если используется естественная сортировка или компаратор не принимает null | Нет | Нет | Быстрое падение |
| ConcurrentHashMap<K,V> | Хеш-таблица | Нет | Нет | Да | Отражает состояние на какой-то момент времени после создания |
| ConcurrentSkipListMap<K,V> | Список с пропусками | Для значений | Нет | Да | Слабая согласованность |
| WeakHashMap<K,V> | Хеш-таблица | Для ключей и значений | Нет | Нет | Быстрое падение |
| IdentityHashMap<K,V> | Хеш-таблица | Для ключей и значений | Нет | Нет | Быстрое падение |
| EnumMap<K,V> | Массив | Для значений | Нет | Нет | Слабая согласованность |
| Hashtable<K,V> | Хеш-таблица | Нет | Да | Да | Быстрое падение |
| Properties | ConcurrentHashMap | Нет | Да | Да | Не гарантированное быстрое падение |
| Set | HashSet<E> | HashMap | Нет | Да | Нет | Нет | Быстрое падение |
| LinkedHashSet<E> | LinkedHashMap | Да | Нет | Нет | Быстрое падение |
| TreeSet<E> | TreeMap | Нельзя использовать null в ключах, если используется естественная сортировка или компаратор не принимает null | Нет | Нет | Быстрое падение |
| EnumSet<E> | Битовая карта (bit array) | Нет | Нет | Нет | Слабая согласованность |
| CopyOnWriteArraySet<E> | CopyOnWriteArrayList | Да | Нет | Да | Моментальный снимок |
| ConcurrentSkipListSet<E> | ConcurrentSkipListMap | Нет | Нет | Да | Слабая согласованность |

*Общий обзор коллекций Java*

Типы итераторов:

1. **Быстрое падение (fail-fast)** — итератор при создании сохраняет "номер поколения" коллекции, который является просто количеством изменений, сделанных в коллекции. При итерации сравнивается текущее поколение и сохраненное. Если они отличаются, то бросается исключение ConcurrentModificationException.
2. **Моментальный снимок (snapshot-style)** — итератор сохраняет снимок коллекции и использует его. Исключения ConcurrentModificationExceptionникогда не бывает.
3. **Слабая согласованность (weakly consistent)** — при итерации можно выполнять некоторые изменения коллекции (но гарантий нет). Не бросает ConcurrentModificationException.