implements

CORE-2 коллекции

март 2023

интерфейс Interface Iterable<T> – это интерфейс, который реализуют коллекции, у него есть метод iterator<T>, который возвращает объект Итератор по элементам типа Т Розовый – справедливо, оправдано: О(n) или линейное время (n - это количество эл. коллекции) класс Iterator — объект с состоянием итерации (бегунок, курсор, указатель). Жёлтый - хорошо O(log n) - деревья Зелёный - превосходно! О(1) или константное время сруктура данных, особенность что под капотом порядок NULL? обход коллекции, поиск сравнение элем. Iterable Временная сложность BigO Collection Среднее Среднее List упорядоченный список с дубликатами Индекс Поиск Вставка Удалени Индекс Поиск Вставка Удаление O(n) динамический массив объектов ДА, по индексу элем. массива ДА sort() + Comparator 0(1) O(n) O(n) 0(1) O(n) O(n) O(n) ArrayList Iterator массив 0(1) FIFO|LIFO LinkedList дву- одно- связанный список ДА, в порядке добавления (бусы) ДА Iterator (t на поиск позиции) O(n) O(n) 0(1) 0(1) O(n) O(n) 0(1) Vector синхронизированный, устарел использовать Collections.svnchronizedList() 0(1) O(n) O(n) O(n) 0(1) O(n) O(n) O(n) int search(Object o) LIFO Stack НЕ упорядоченное множество уникальных Equals() HashSet HashMap СЛУЧАЙНЫЙ (зависит от хэш и бакета) ДА contains() по бакету, хэшкоду n/a 0(1) 0(1) 0(1) n/a O(n) O(n) O(n) на основе хэш-таблицы LinkedHashSet LinkedHashMap n/a 0(1) 0(1) 0(1) n/a O(n) O(n) O(n) на основе связанного списка ДА, в порядке вставки SortedSet упорядоченный по возрастанию ДА, по возрастанию Compare() NavigableSet добавлена навигация NavigableMap ДА, добавлена навигация Iterator, descendingIterator $O(\log (n)) O(\log (n)) O(\log (n))$ n/a $O(\log(n))$ $O(\log(n))$ $O(\log(n))$ TreeSet дерево TreeMap ДА, крсн-черн дерево FIFO очередь Oueue вход - хвост, выход - голова ДА, в порядке добавления FIFO FIFO PriorityQueue Очередь, выход по приоритету ДА, в порядке сравнения: Comparable, natural order, Comparator FIFOILIFO Deque LIFO/FIFO очередь двусторонняя ArrayDeque имеет реализацию кольцевой буфер ОТДЕЛЬНАЯ ВЕТВЬ, не наследуется от Collection ДА, хэш 0 Мар<К, V> карта пар ключ-значение (еще говорят "словарь") SortedMap NavigableMap Iterator, descendingIterator AbstractMap class contains() по бакету, хэшкоду хэш, ссылка, equals HashMap массив бакетов + LinkedList (1св) массив бакетов СЛУЧАЙНЫЙ (зависит от хэш и бакета) ДА n/a 0(1) 0(1) 0(1) n/a O(n) O(n) O(n) при перестроении КЧ дерево + LinkedList LinkedHashMap массив бакетов + LinkedList (2св) default insertion-order (как вставили) доступ по порядку последнего n/a 0(1) 0(1) 0(1) n/a O(n) O(n) O(n) поддержка двусвязного списка можно accessOrder = true (конструктор) обращения к эл. (access-order) Слабые ключи, элемент удаляется GC WeekHashMap TreeMap дерево с возможностью навигации по возраст. или как задано Comparator не всегда, n/a O(log (n)) O(log (n)) O(log (n) n/a O(log(n)) O(log(n)) O(log(n))Comparator массив 13 14 15 Iterable Map startertutorials com null null null null null Collection SortedMap Set List Queue hash | 99486 NavigableMap hash hash 101603 нода/узел PriorityQueu HashSet null key "idx" next ← key "key" null next null ArrayList Deque value null value "two" value "one" LinkedList По дефолту: Перестроение в дерево Interface 16 корзин, LF = 0,75. произойдёт, когда в hash 51 Class массиве будет 64 бакета и Когда заполняется key хотя бы в одном бакете нода/узел 75%, т.е. 12/16, массив next null Stack TreeSet добавится 8я нода

интерфейсы абстрактные классы обычные класс

→ наследование ---> реализация интерфейса

увеличится в два раза

value "zero"

static final int TREEIFY_THRESHOLD = 8