

Коллекции из стандартной библиотеки Java

Юрий Литвинов
yurii.litvinov@gmail.com

30.01.2019г

Коллекции

- ▶ Интерфейсы коллекций
- ▶ Реализации общего назначения
- ▶ Специализированные реализации
- ▶ Многопоточные реализации
- ▶ Абстрактные реализации
- ▶ Обёртки
- ▶ Алгоритмы
- ▶ Инфраструктура

Основные интерфейсы

- ▶ `Iterable` — всё, по чему можно итерироваться
- ▶ `Collection` — просто группа объектов
 - ▶ `Set`, `SortedSet` — множества
 - ▶ `List` — упорядоченная коллекция, вводит понятие “позиция”
 - ▶ `Queue`, `Deque` — очереди, деки
- ▶ `Map` — отображение
 - ▶ `SortedMap`
 - ▶ `NavigableMap` — позволяет получить элемент, ближайший к искомому

Реализации общего назначения

Интерфейс	Хеш-таблица	Массив	Дерево	Список	Хеш-таблица + Список
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Deque		ArrayDeque		LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

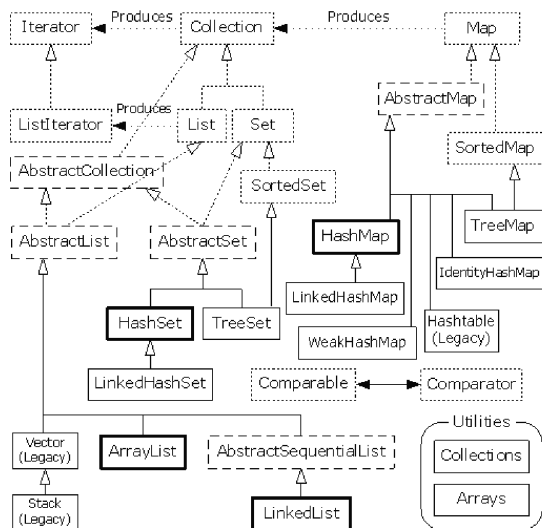
Абстрактные реализации

- ▶ **AbstractCollection**
 - ▶ Требуется `iterator()` с `hasNext()` и `next()` и `size()`
 - ▶ Для возможности модификации — переопределить `add()` и, для итератора, `remove()`
 - ▶ Надо конструктор без аргументов и принимающий `Collection`
 - ▶ Могут быть переопределены остальные методы, для большей эффективности
- ▶ **AbstractSet**
- ▶ **AbstractList**
- ▶ **AbstractSequentialList**
- ▶ **AbstractMap**
 - ▶ Требуется реализовать `entrySet()`
 - ▶ “View Collection” — сам элементы не хранит
 - ▶ Для возможности модификации — переопределить `put()` и, для итератора, `remove()`

Инфраструктура

- ▶ Iterator
 - ▶ hasNext()
 - ▶ next()
 - ▶ remove() — можно не реализовывать
- ▶ ListIterator — двунаправленный, работает с индексами
- ▶ Comparable — задаёт Natural Ordering
- ▶ Comparator — задаёт произвольный порядок

Диаграмма



© B. Eckel, Thinking in Java

Исключения

- ▶ `UnsupportedOperationException` — реализация не поддерживает метод, объявленный в интерфейсе (помечен как `optional`)
- ▶ `ConcurrentModificationException` — попытка продолжить обход изменившейся коллекции

Утилиты

- ▶ `java.util.Collections`
 - ▶ `binarySearch`, `copy`, `sort`, `shuffle` и т.д.
 - ▶ `emptyList`, `emptyMap`, ...
 - ▶ `singleton`, `singletonList`, ...
 - ▶ `unmodifiableList`, `unmodifiableMap`, ...
- ▶ `java.util.Arrays`
 - ▶ `asList`
 - ▶ `sort`, `compare`, `equals`, `fill` и т.д. для разных примитивных типов

Задача

Переделать список из задачи про хеш-таблицы, чтобы он был генериком и реализовывал стандартный интерфейс `List<E>`. Нужны:

- ▶ Мутабельность
- ▶ Более эффективный итератор, чем по умолчанию
 - ▶ С `remove()`
- ▶ Эффективный `ListIterator`
- ▶ Тесты