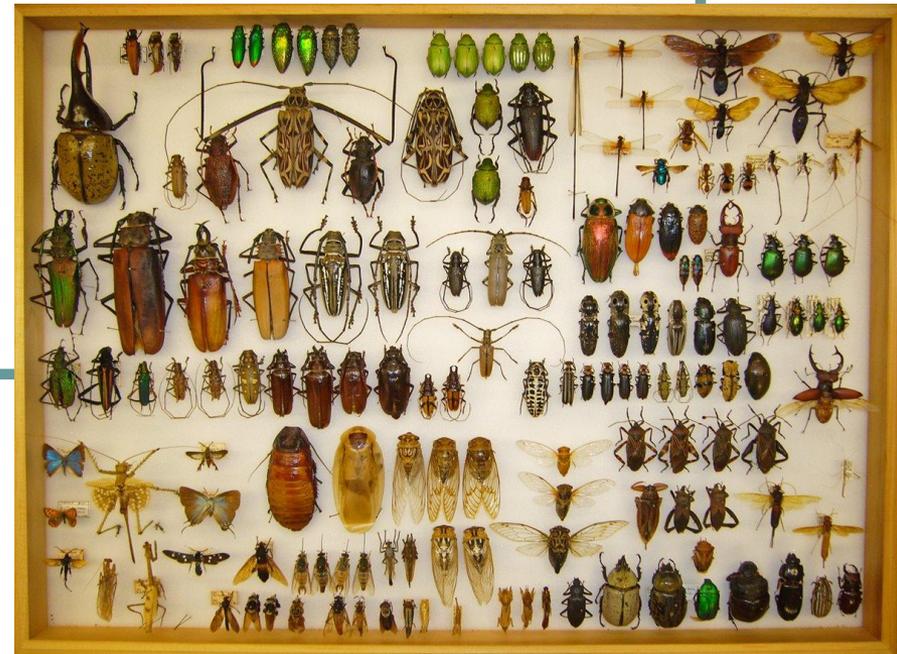


Java Collection Framework



Коллекции – это наше всё

Стандартные коллекции являются фундаментальной основой программирования на языке Java. Они используются почти в любом приложении, поэтому каждый уважающий себя разработчик просто обязан уметь эффективно использовать Java Collections Framework.

Определение

Коллекция — это контейнер для группы элементов, которые представляют собой единое целое. Например: банка конфет, список имён и тд. Коллекции используются почти в каждом языке программирования, и Java не является исключением. Как только коллекции появились в Java, то насчитывали всего несколько классов: **Vector**, **Stack**, **Hashtable**, **Array**. Но уже в Java 1.2 появился полноценный Java Collections Framework.

Интерфейсы

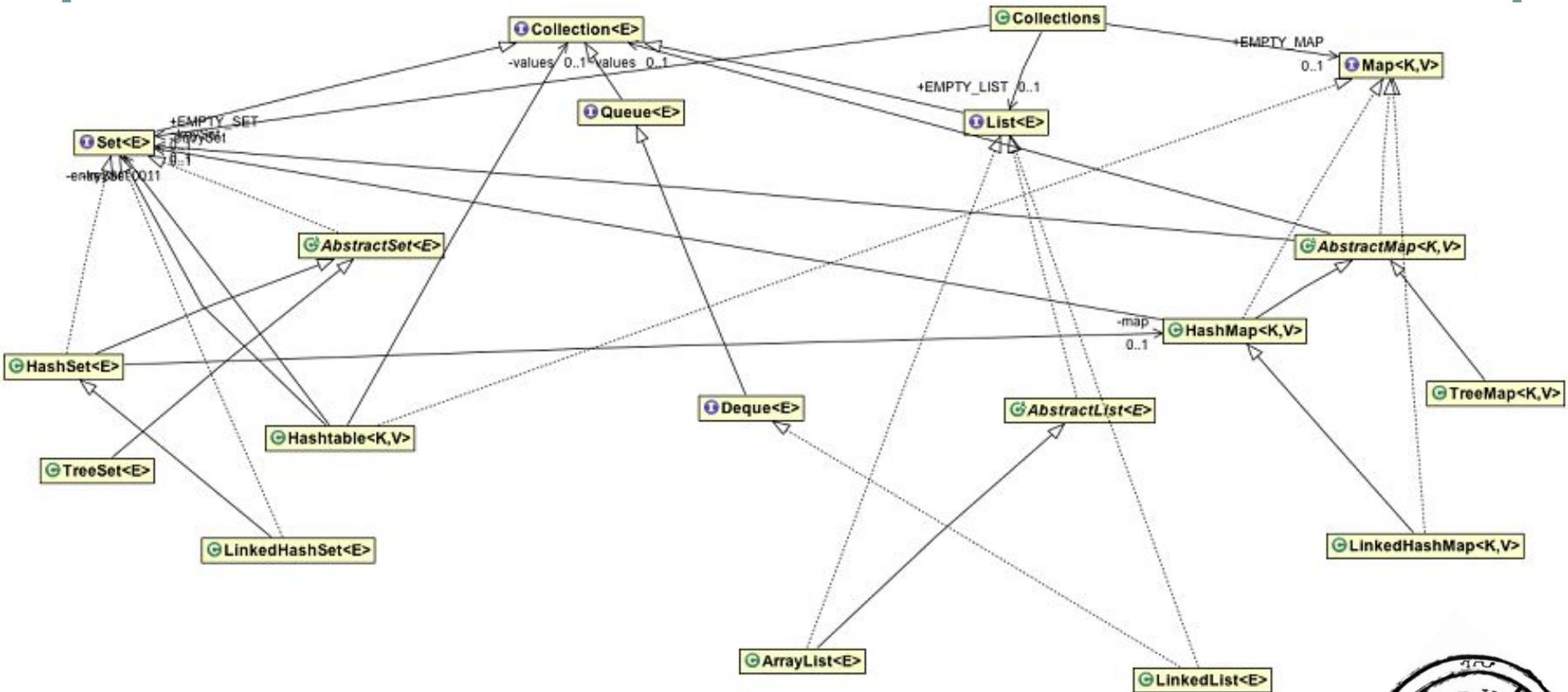
Интерфейсы обеспечивают абстрактные типы данных для представления коллекций.

java.util.Collection — находится на вершине иерархии, и содержит наиболее важные методы: `size()`, `iterator()`, `add()`, `remove()`, `clear()`. Каждая коллекция должна реализовывать эти методы. Также есть другие важные интерфейсы: **java.util.List**, **java.util.Set**, **java.util.Queue** и **java.util.Map**. **Map** является единственным интерфейсом, который не наследуется от интерфейса **Collection**, но является неотъемлемой частью коллекций. Все интерфейсы фреймворка находятся в пакете `java.util`.

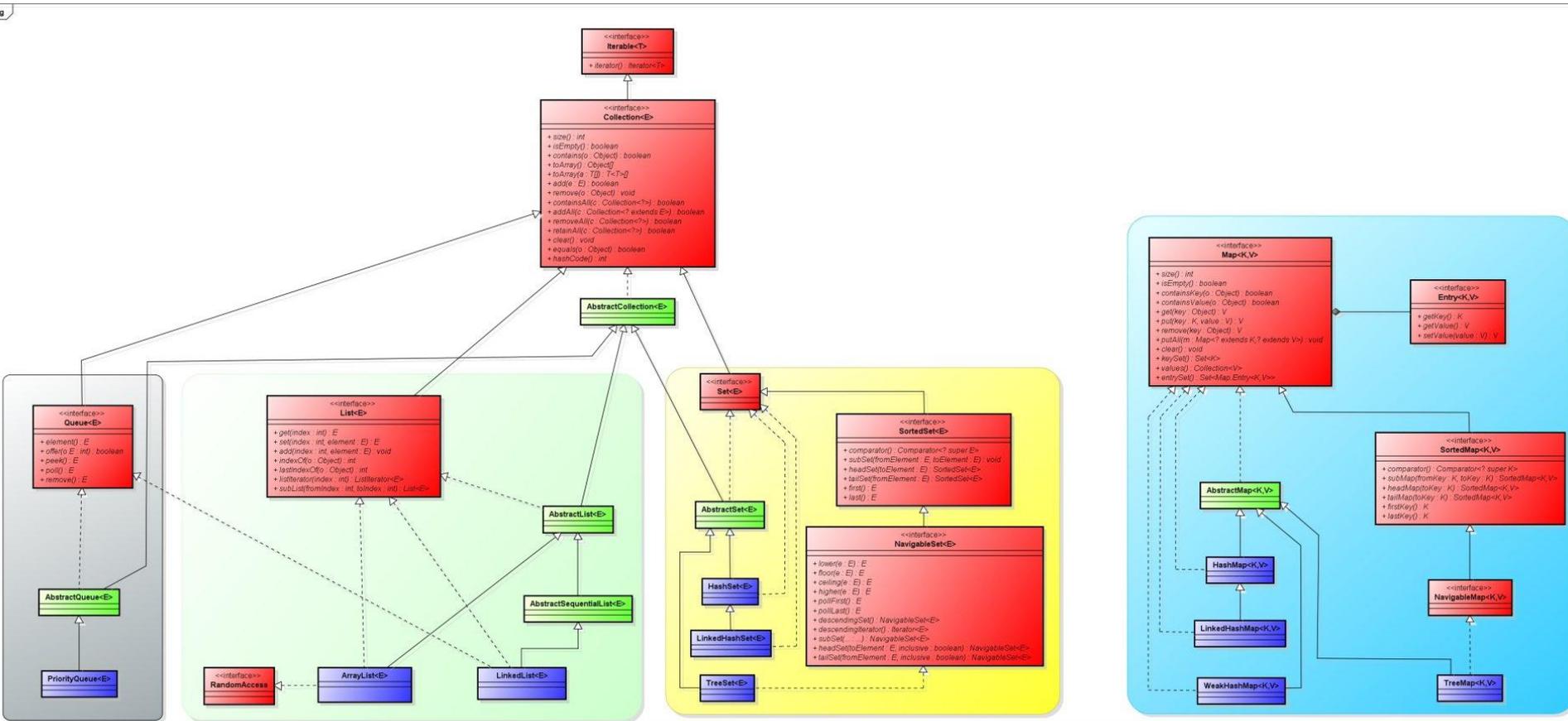
Реализации интерфейсов

JCF предоставляет готовые классы с реализацией вышеупомянутых интерфейсов. С помощью классов `ArrayList`, `LinkedList`, `HashMap`, `TreeMap`, `HashSet`, `TreeSet` можно решить огромное количество задач, но если нужна специальная реализация той или иной коллекции, то можно отнаследоваться и работать со своей реализацией. В Java 1.5 добавили потокобезопасные коллекции, которые позволили изменять содержимое коллекции время итерации по элементам. Эти классы находятся в пакете **`java.util.concurrent`**. Все классы коллекций находятся в пакетах `java.util` и `java.util.concurrent`.

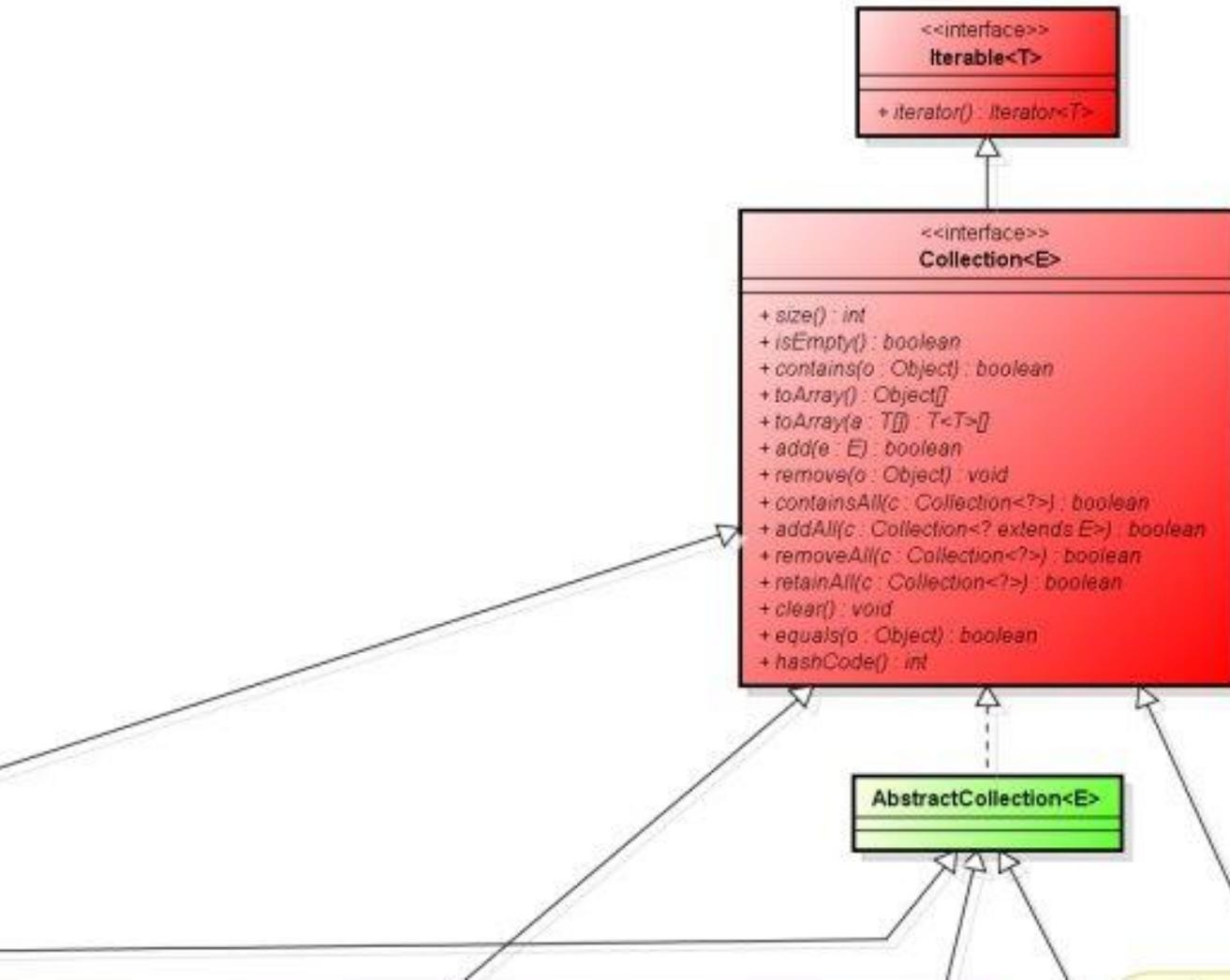
Диаграмма классов JCF

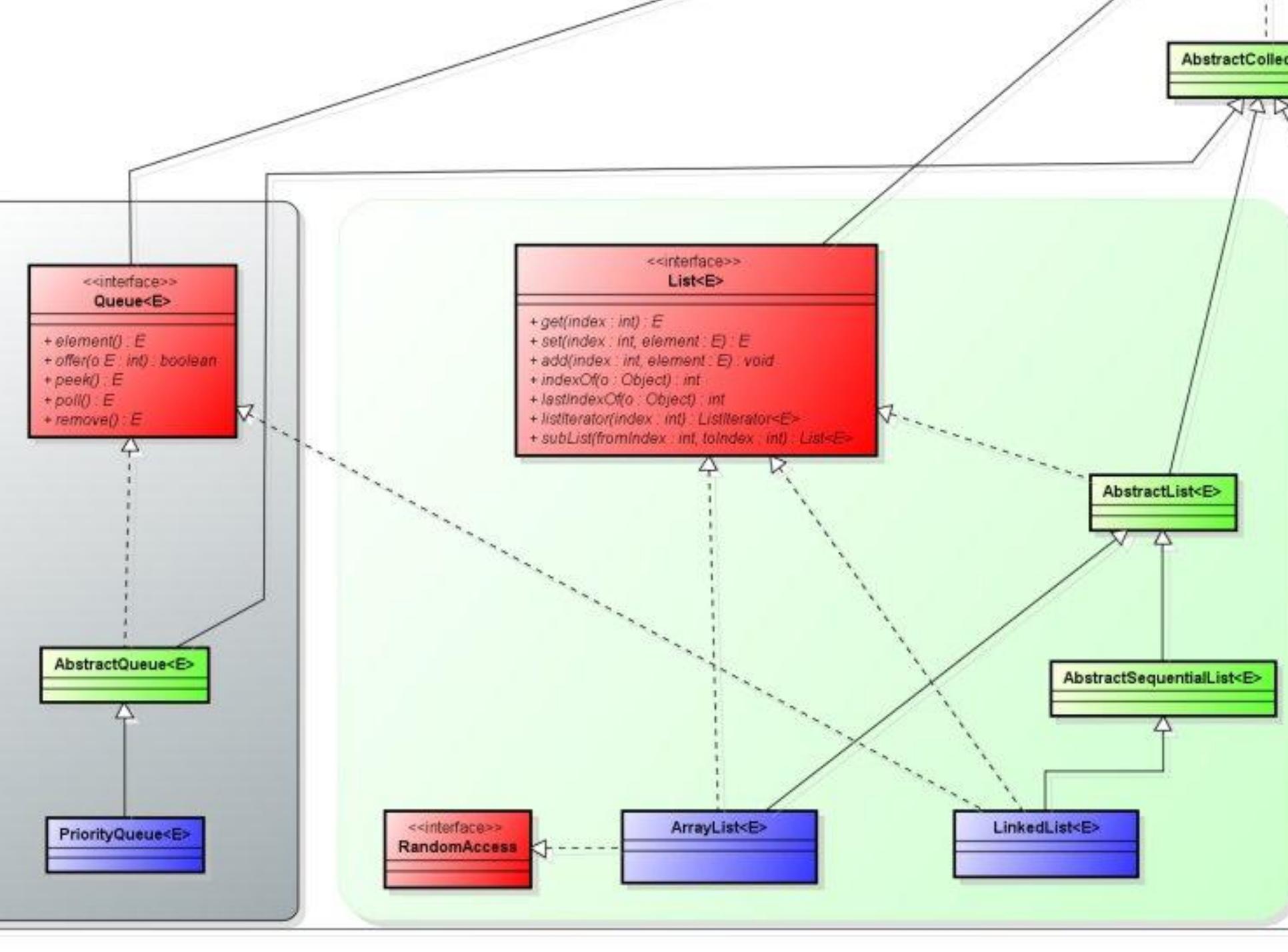
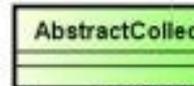
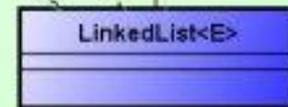
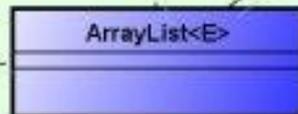
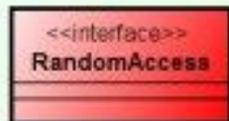
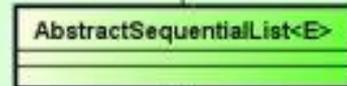
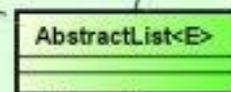
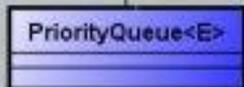
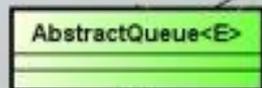


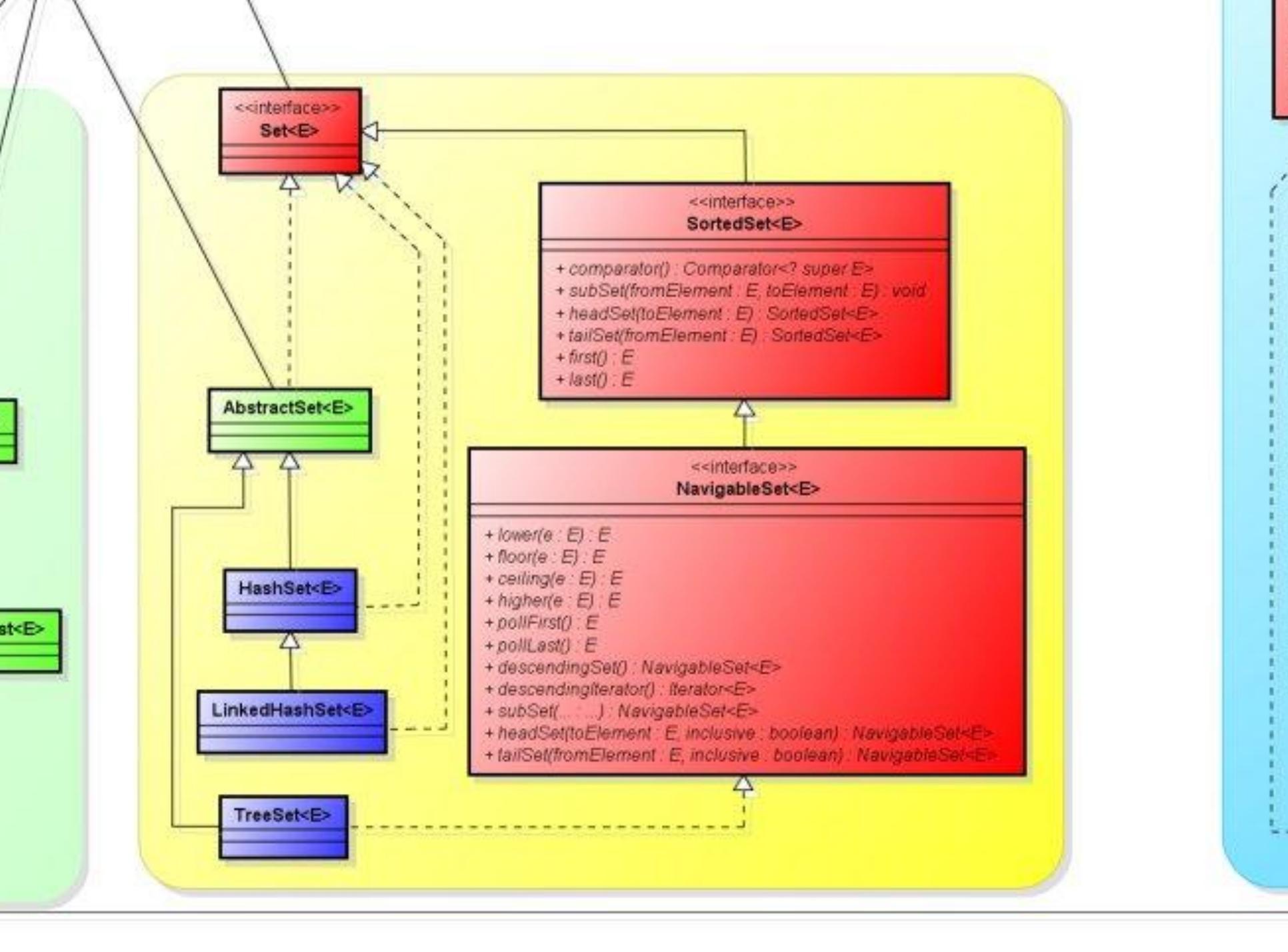
Ну ладно, ладно... теперь нормальная диаграмма 😊

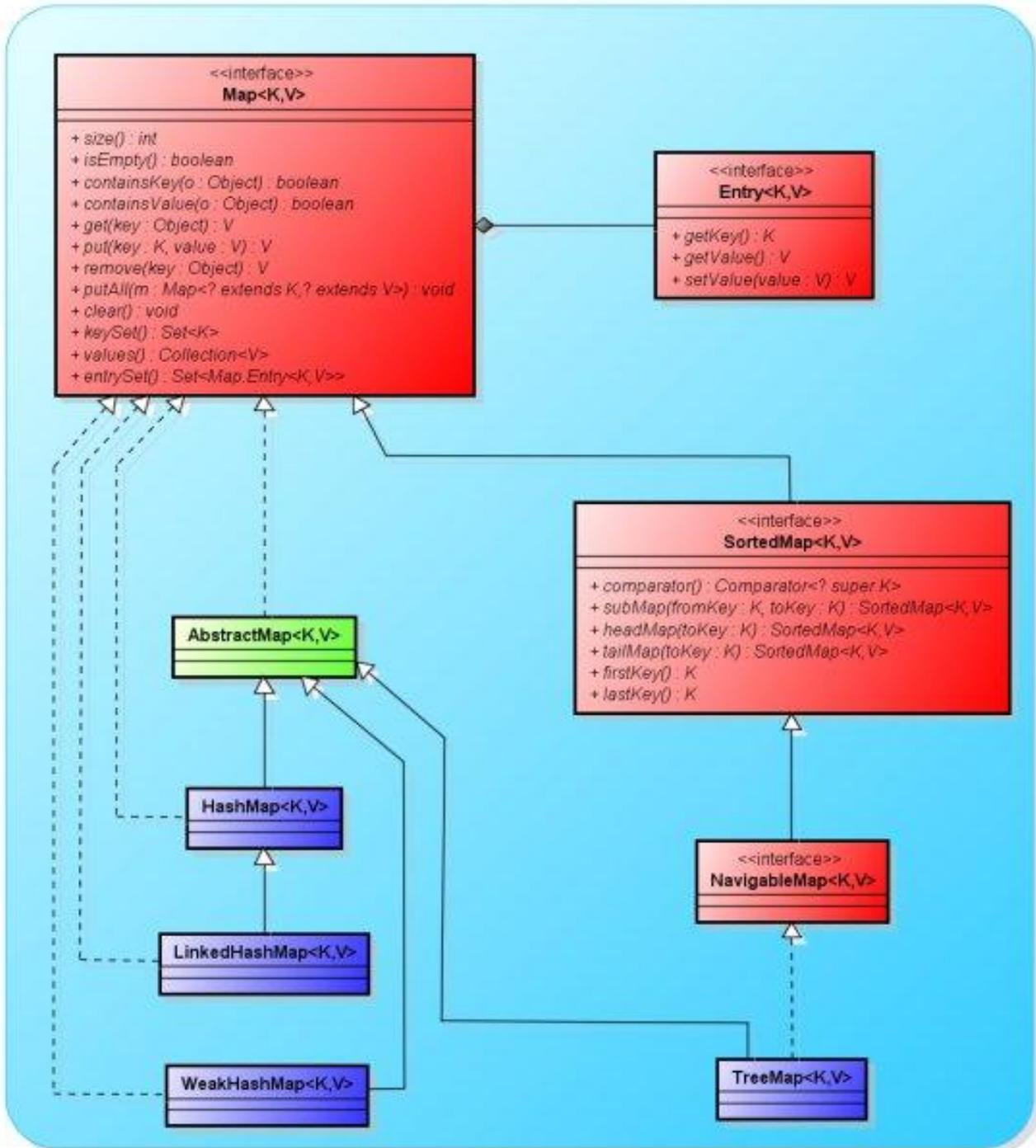
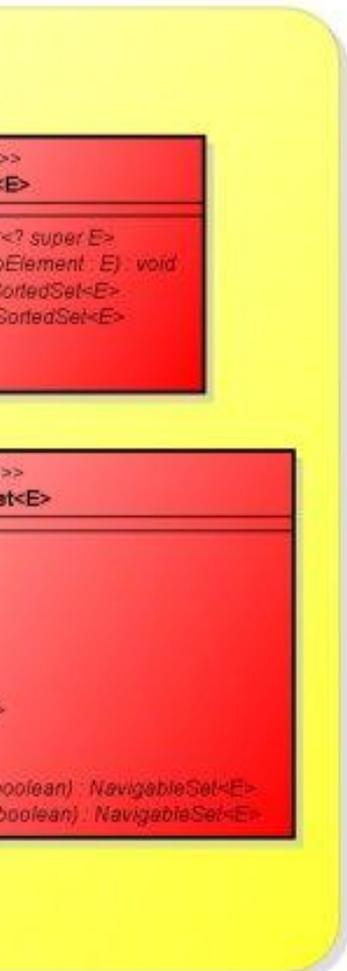


Collection interface









Преимущества JCF

- **Требует меньше усилий** — фреймворк располагает множеством готовых распространённых типов коллекций и полезных методов для манипуляции данными
- **Отличное качество реализации** — использование хорошо проверенных коллекций увеличивает качество программы
- **Повторное использование и совместимость**

Интерфейсы коллекций

Обратите внимание, что все интерфейсы являются generic, например `public interface Collection<E>`. Использование `<E>` — это указание типа объекта, который коллекция может содержать. Это помогает сократить ошибки времени выполнения с помощью проверки типов объектов во время компиляции.

Iterable interface

Итератор предоставляет методы для перебора элементов любой коллекции. Можно получить экземпляр итератора из коллекции с помощью метода `iterator()`. Итераторы позволяют удалить элементы из базовой коллекции во время выполнения итерации, либо просто перебрать все элементы.

Другие интерфейсы

- **Set** - набор представляет собой коллекцию, которая не может содержать повторяющиеся элементы
- **List** - список представляет собой упорядоченный набор элементов и может содержать повторяющиеся элементы. Можно получить доступ к любому элементу по индексу. Один из наиболее используемых типов коллекций

Другие интерфейсы

- **Queue** – элементы обычно располагаются по принципу FIFO
- **Deque** – поддерживается добавление и удаление элементов как с конца, так и с начала очереди
- **Map** – коллекция объектов-пар ключ-значение, причём обычно ключи повторяться не могут

Другие интерфейсы

- **SortedSet** - представляет собой множество, в котором элементы хранятся в порядке возрастания
- **SortedMap** - содержит элементы в порядке возрастания ключей. Является аналогом SortedSet. Используется для естественно упорядоченных пар ключ/значение, например, словарей и телефонных справочников

Сравнение коллекций

Коллекция	Упорядочивание	Random Access	Ключ-значение	Дубликат Элементы	Нулевой элемент	Потоко безопасность
ArrayList	Да	Да	Нет	Да	Да	Нет
LinkedList	Да	Нет	Нет	Да	Да	Нет
HashSet	Нет	Нет	Нет	Нет	Да	Нет
TreeSet	Да	Нет	Нет	Нет	Нет	Нет
HashMap	Нет	Да	Да	Нет	Да	Нет
TreeMap	Да	Да	Да	Нет	Нет	Нет
Vector	Да	Да	Нет	Да	Да	Да
Hashtable	Нет	Да	Да	Нет	Нет	Да
Properties	Нет	Да	Да	Нет	Нет	Да
Stack	Да	Нет	Нет	Да	Да	Да
CopyOnWriteArrayList	Да	Да	Нет	Да	Да	Да
ConcurrentHashMap	Нет	Да	Да	Нет	Нет	Да
CopyOnWriteArraySet	Нет	Нет	Нет	Нет	Да	Да

Что почитать о коллекциях

- <http://www.quizful.net/post/Java-Collections>
- <http://docs.oracle.com/javase/tutorial/collections/>
- <http://www.seostella.com/ru/article/2012/08/07/kollekcii-collections-v-java-vvedeniye.html>
- <http://habrahabr.ru/post/128017/>

Collection

http://www.tutorialspoint.com/java/java_collection_interface.htm

List

http://www.tutorialspoint.com/java/java_list_interface.htm

Set

http://www.tutorialspoint.com/java/java_set_interface.htm

SortedSet

http://www.tutorialspoint.com/java/java_sortedset_interface.htm

Queue

<http://tutorials.jenkov.com/java-collections/queue.html>

Deque

<http://tutorials.jenkov.com/java-collections/deque.html>

ArrayList, LinkedList

<https://git.io/vKTnJ>

List Iterator

<https://git.io/vKTIC>

http://www.tutorialspoint.com/java/java_using_iterator.htm

<http://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html>

<http://www.golovachcourses.com/collections-iterator/>

<http://habrahabr.ru/post/84184/>

HashSet, LinkedSet, TreeSet

<https://git.io/vKTnp>

ArrayDeque

<https://git.io/vKTcn>

EnumSet

<https://git.io/vKTWL>

User objects in collection

<https://git.io/vKTWR>

HashMap, TreeMap

<https://git.io/vKT8m>

<http://www.quizful.net/post/Using-Map-in-Java>

<http://developer.alexanderklimov.ru/android/java/map.php>

<http://developer.alexanderklimov.ru/android/java/hashmap.php>

TreeSet Comparator

<https://git.io/vKT8S>

Map Comparator

<https://git.io/vKT44>

Практика

Создать коллекцию `TreeSet` для объектов типа `Student`. Сделать несколько компараторов для сортировки студентов внутри коллекции по разным критериям (средний балл, имя по возрастанию, фамилия по убыванию). Применить компараторы (или лямбды).

Алгоритмы Collections

<https://git.io/vKT4V>

- Reverse order
- Sort
- Shuffle
- Min
- Max

Алгоритмы Arrays

- Sort
- Fill
- BinarySearch
- CopyOf
- Equals
- AsList
- ToString

Legacy Collections

- Vector
- Stack
- HashTable
- Dictionary
- Properties
- BitSet

Vector

<https://git.io/vKTBH>

Stack

<https://git.io/vKTRO>

Properties

<https://git.io/vKT03>

BitSet

<https://git.io/vKl9v>

Коллекции в Android

<https://developer.android.com/reference/java/util/Collections.html>

<https://proglib.io/p/data-structures/>

Guava

- <https://github.com/google/guava/wiki/Release19>
- <https://github.com/google/guava/wiki>
- Практика: качаем библиотеку, подключаем к проекту

java.util.concurrent

- <https://habrahabr.ru/company/luxoft/blog/157273/>
- <http://www.ibm.com/developerworks/ru/library/j-jtp07233/>
- <http://javacore.ru/topic/36-core-multithreading-synchronization.htm>

Уроки по Java

- <https://nuancesprog.ru/p/2334>
- https://www.youtube.com/playlist?list=PL0IO_mlqDDFUzG5WOCUVmqx4CBW2qIuIV

Нововведения в Java

- <https://tproger.ru/translations/java-9-features-examples/>
- <https://habr.com/post/424683/>
- <https://nuancesprog.ru/p/2234/>
- <https://youtu.be/v3dr4e54TLA>

- <https://habr.com/ru/post/253787/>