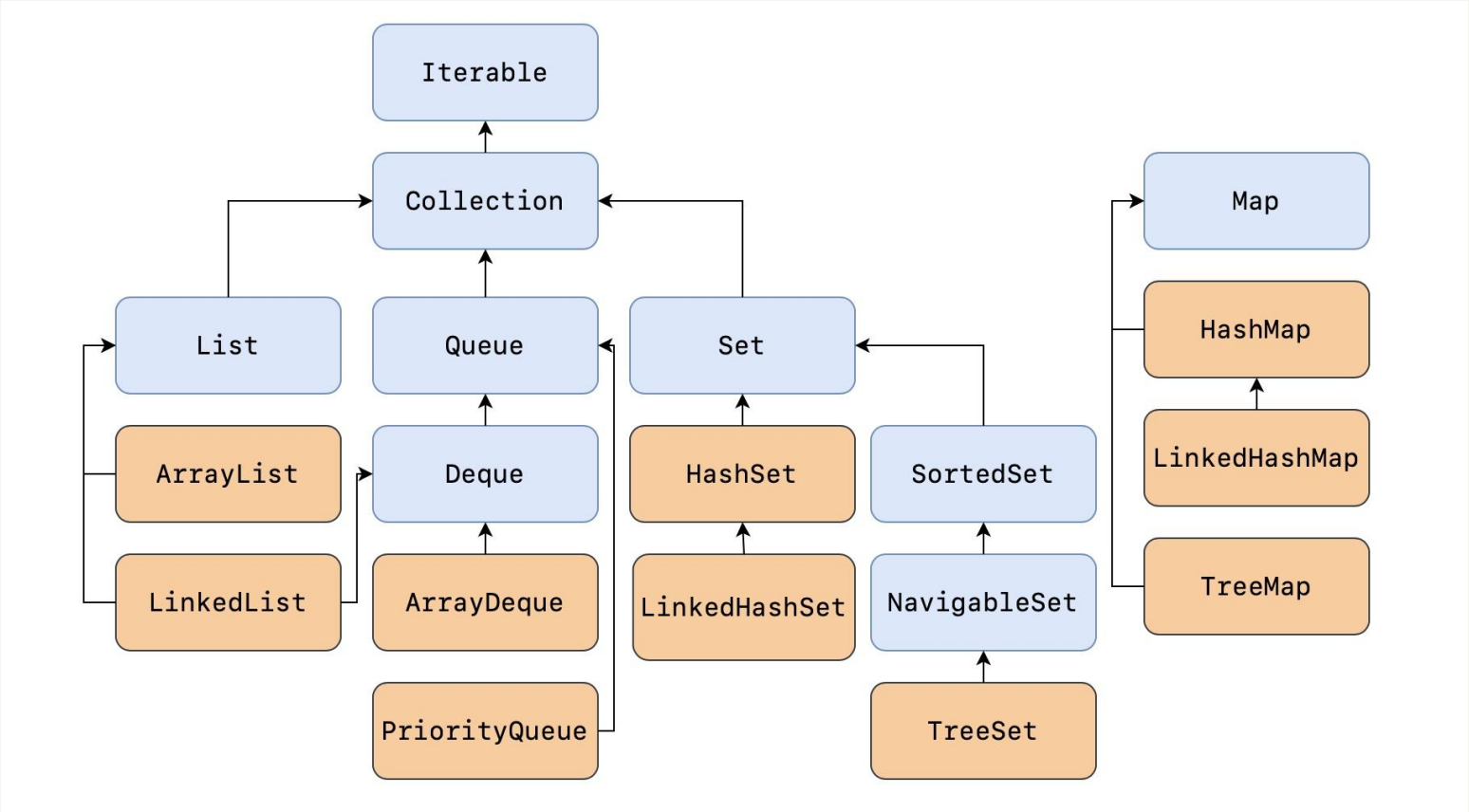
Хекслет

Коллекции в Java. Set. HashSet.

Наставник: Теплинская Мария Георгиевна
Группа: java-167
Дата: 08.02.2023

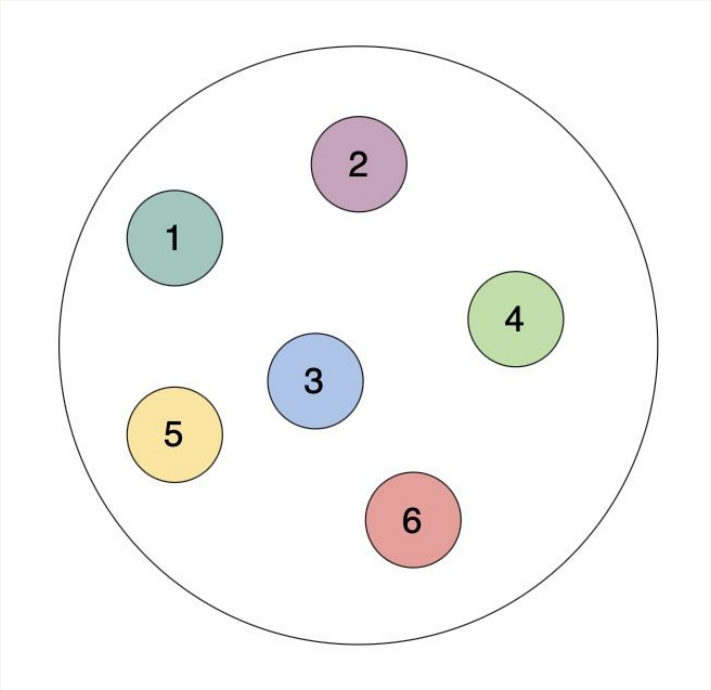
Иерархия коллекций



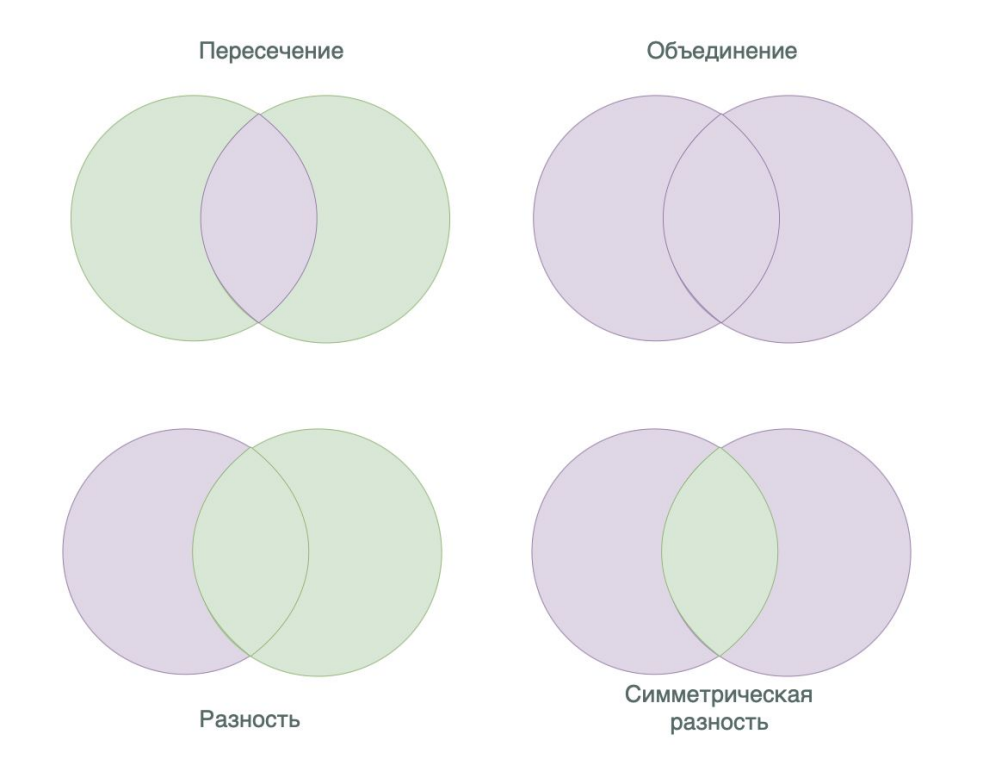
Что такое множество

Множество - это набор каких-либо (любых) объектов, которые являются элементами этого множества.

Примеры: множество букв русского алфавита, множество целых чисел и тд.



Операции над множествами



Определение Set

Set - коллекция, которая не содержит дубликатов. То есть она представляет из себя множество уникальных элементов.

Уникальность определяется таким образом, что каждая пара элементов из этого множества должны быть не равны по equals.

Set - это интерфейс, который имеет разные реализации. Методы этого интерфейса налагают дополнительные ограничения на методы интерфейса Collection.

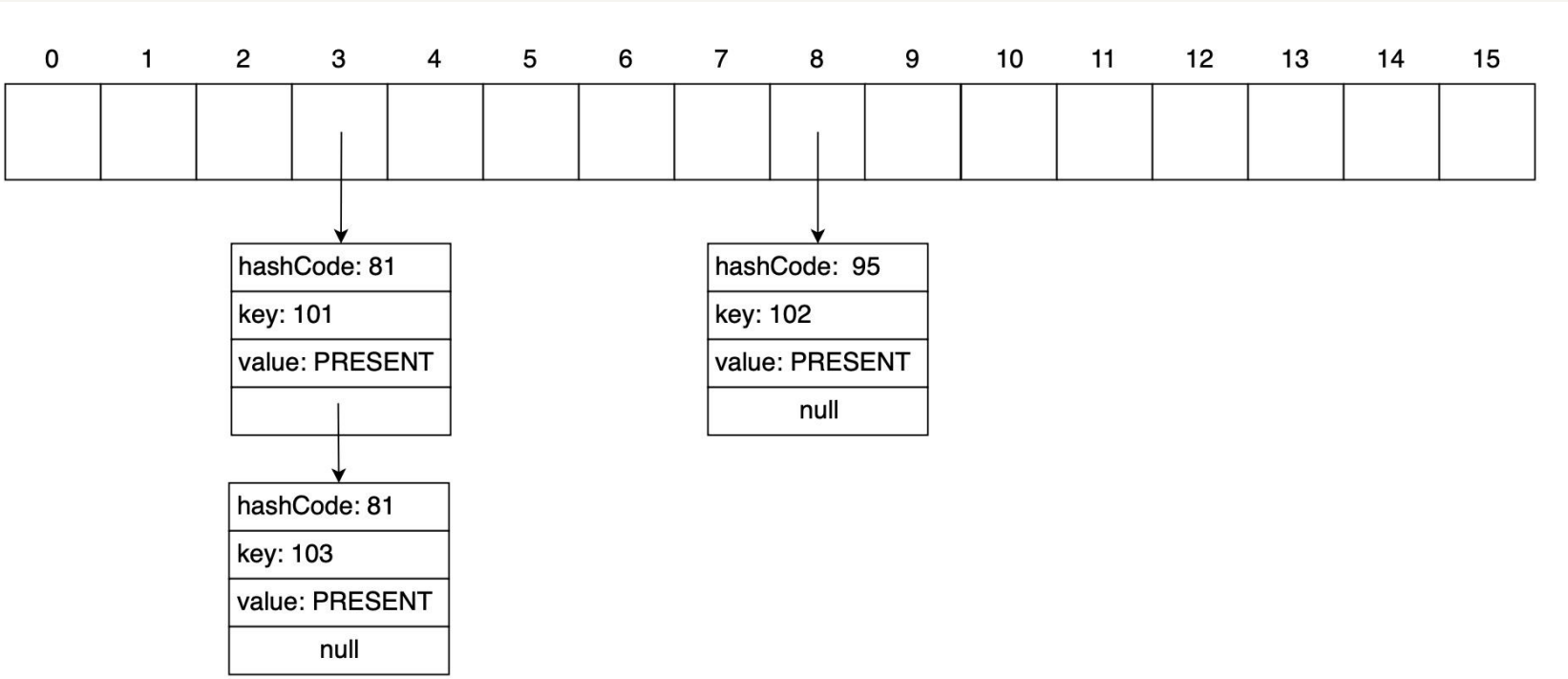
Определение HashSet

HashSet - это реализация интерфейса Set, основанная на хэш-таблице.

Внутри HashSet реализован через HashMap, ключами которой являются элементы множества, а значениями некоторая “пустышка”, константный объект PRESENT класса Object.

HashSet как и HashMap никак не гарантирует порядок элементов и как и HashMap позволяет добавлять null-элемент.

Структура HashSet



SortedSet и NavigableSet

SortedSet - коллекция (интерфейс), которая воплощает концепцию множества и гарантирует порядок элементов. Для того чтобы можно было установить порядок, элементы должны быть сравнимы между собой.
Обеспечить это можно двумя способами: через *Comparable* или *Comparator*.

NavigableSet - коллекция (интерфейс), которая расширяет SortedSet методами, позволяющими отыскать ближайший элемент к какому-то заранее заданному. Эту коллекцию можно обходить и по возрастанию, и по убыванию.

Определение TreeSet

TreeSet - коллекция, реализация интерфейса Set (SortedSet, NavigableSet).

Основана на коллекции *TreeMap*. Реализована через структуру данных красно-черное дерево.

Объявление Set

```
Set<Integer> set1 = new HashSet<>();
Set<Integer> set2 = new HashSet( initialCapacity: 32);
Set<Integer> set3 = new HashSet( initialCapacity: 32,  loadFactor: 0.98f);
Set<Integer> set4 = new HashSet(set3);

NavigableSet<Integer> nset1 = new TreeSet<>();
NavigableSet<Integer> nset2 = new TreeSet<>(Comparator.reverseOrder());
NavigableSet<Integer> nset3 = new TreeSet<>(nset1);
NavigableSet<Integer> nset4 = new TreeSet<>(new ArrayDeque());
```

Основные операции

```
Set<Integer> set = new HashSet<>();
set.add(1);
set.add(2);
set.add(3);
set.add(2);
System.out.println(set);
System.out.println(set.size());
System.out.println(set.contains(3));
System.out.println(set.contains(5));
set.remove( o: 3);
System.out.println(set);
```

```
NavigableSet<Integer> set = new TreeSet<>();
for (int i = 1; i <= 30; i++) {
    set.add(i);
}
System.out.println(set);
System.out.println(set.descendingSet());
System.out.println(set.first());
System.out.println(set.last());
set.remove( o: 15);
set.remove( o: 16);
System.out.println(set.ceiling( e: 15));
System.out.println(set.floor( e: 16));
```

Заключение

- Set - коллекция в Java, которая реализует множество
- Коллекция Set не содержит дубликатов
- HashSet - популярная реализация этой коллекции. Основана на hash-таблице
- Если в коллекции Set нужен порядок элементов, то можно использовать SortedSet.
- NavigableSet добавляет полезные методы к интерфейсу SortedSet
- TreeSet - реализация отсортированного множества, основанная на красно-черном дереве

Спасибо за внимание