

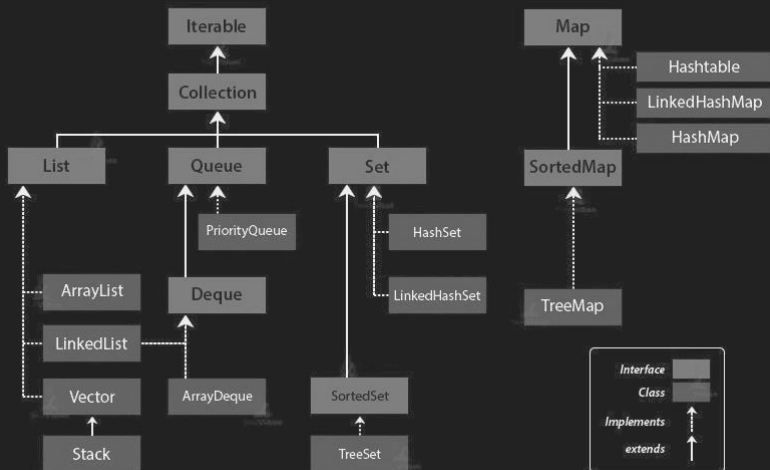


Java Collection III



Java Collection Framework

Collection Framework Hierarchy in Java



[Java Collection Framework](#) предоставляет архитектуру для хранения и манипулирования группой объектов.

Он включает в себя различные интерфейсы, реализации и алгоритмы для работы с коллекциями.

Еще одна статья покороче: [ссылка](#)



Set и HashSet

```
package 136.slides.ex1;

import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        Set<String> fruitSet = new HashSet<>();
        // Добавляем "Apple" в Set
        fruitSet.add("Apple");
        // Добавляем "Banana" в Set
        fruitSet.add("Banana");
        // Попытка добавить "Apple" снова - без эффекта
        fruitSet.add("Apple");

        // Вывод: [Apple, Banana]
        System.out.println(fruitSet);
    }
}
```

Set в Java представляет собой коллекцию, которая не содержит дубликатов элементов.

HashSet является одной из наиболее используемых реализаций интерфейса Set. Она использует хеш-таблицу для хранения элементов.

HashSet позволяет выполнять базовые операции, такие как добавление, удаление и проверка наличия элемента, с достаточно высокой производительностью.



Set и HashSet

`.contains(String value)`

`.remove(String value)`

`.add(String value)`

`.isEmpty()`

`.clear()`

`.isEmpty()`



Нет индексов

Значения все уникальные

Методы как у Collection

Операции множеств - объединение, пересечение, разность множеств

Зачем нужно:

- нужен быстрый поиск с относительно быстрой вставкой
- нужно только проверять наличие объекта в коллекции
- нужна коллекция без дублирований



Queue

```
package 136.slides.ex3;

import java.util.LinkedList;
import java.util.Queue;

public class Main {
    public static void main(String[] args) {
        Queue<String> queue = new LinkedList<>();
        // Добавление элементов в очередь
        queue.offer("Apple");
        queue.offer("Banana");

        // Удаление и возврат первого элемента ("Apple")
        System.out.println(queue.poll());
        // Просмотр (но не удаление)
        // первого элемента ("Banana")
        System.out.println(queue.peek());
    }
}
```

Queue — это коллекция, предназначенная для хранения элементов перед их обработкой.

В Java Queue интерфейс предоставляет операции для добавления, удаления и проверки элементов.

LinkedList как Queue: LinkedList реализует Queue и может быть использован как очередь для FIFO (First-In-First-Out) операций.

Дублирование допустимо.



Queue

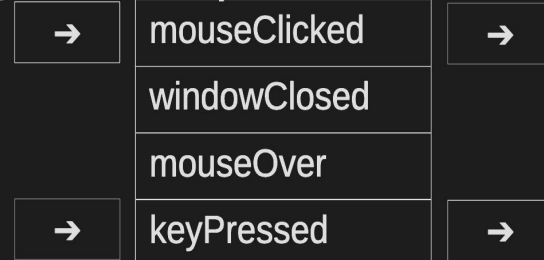
- List

0	яблоко
1	апельсин
2	банан
3	киви

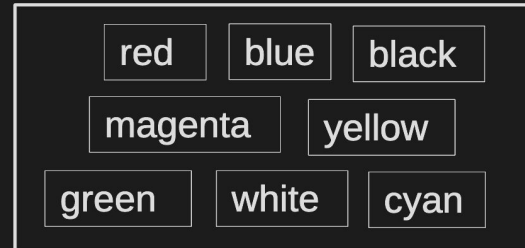
Map

и	37153142
т	31620970
м	16203060
о	55414481

Queue/Deque/Stack



Set





Arrays & Collections

класс Arrays:

Методы для работы с массивами (обычными):

- сортировка, поиск, копирование, заполнение
- и еще много полезного
- `public static <T> List<T> asList(T... a)`
 - возвращает список на основе массива `a`
 - элементы менять можно
 - размер менять нельзя

класс Collections:

алгоритмы для List

- `sort()`, `shuffle()`, `reverse()`,
`fill()`, `swap()`, `binarySearch()`

фабрики коллекций

- `emptySet`, `emptyList`, `emptyMap`,
- `singleton`, `singletonList`, `singletonMap`

и еще много полезного...



Контрольная точка

- Понятно ли?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



Домашнее задание

Поместите часть бобров из 35 домашки в мешок (в `hashset`) (надо переопределить `equals` и `hashCode` у них)

Проитерируйтесь по (старому) списку бобров и проверьте для каждого есть ли он в мешке



The end