

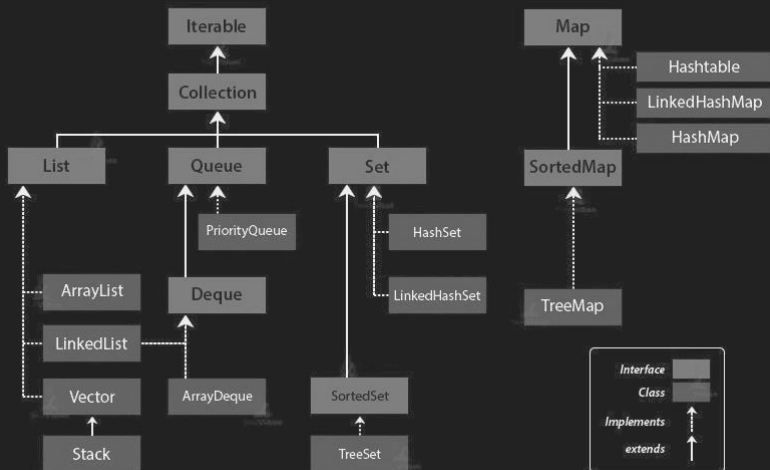


Java Collection IV



Java Collection Framework

Collection Framework Hierarchy in Java



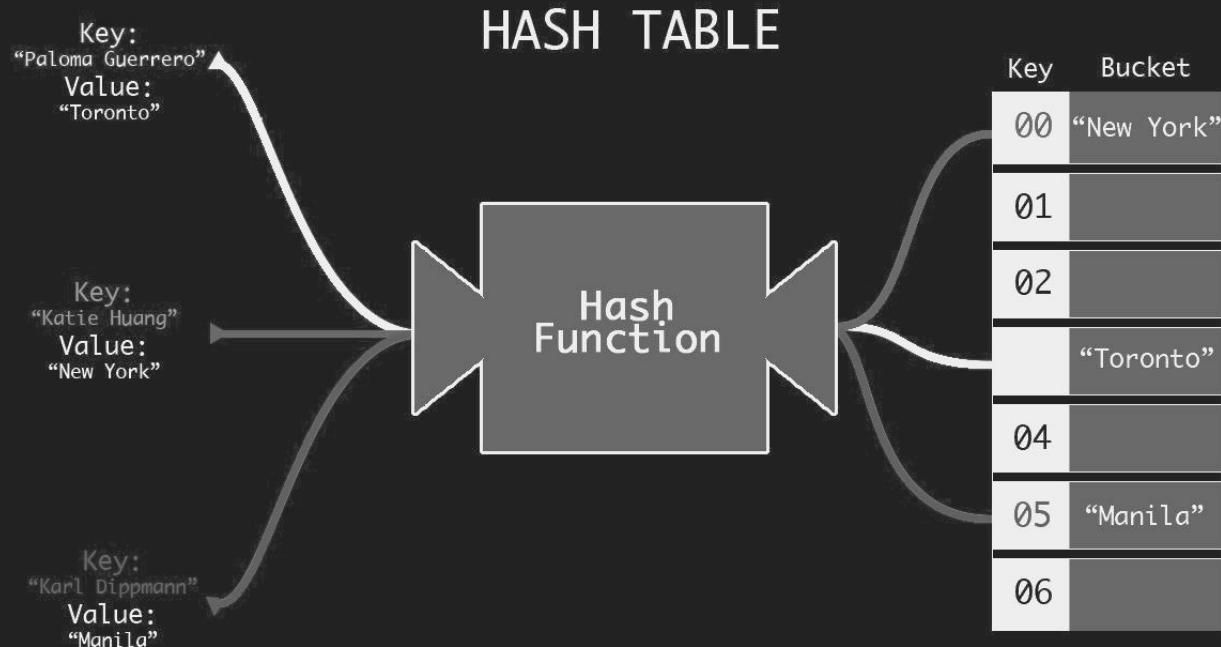
[Java Collection Framework](#) предоставляет архитектуру для хранения и манипулирования группой объектов.

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

Еще одна статья покороче: [ссылка](#)

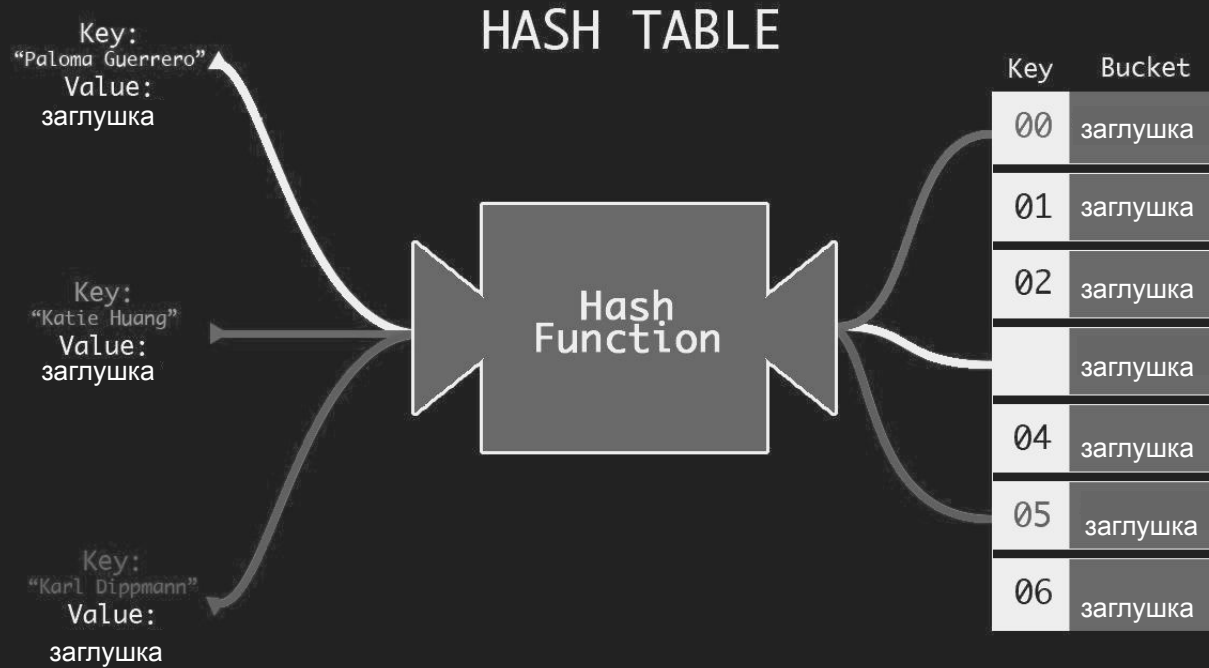


HashMap





HashSet





Iterable & Iterator

```
package l37.slides.ex1;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

class IterableIteratorExample {
    public static void main(String[] args) {
        List<String> fruits = new ArrayList<>();
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Cherry");

        // Получение Iterator
        Iterator<String> iterator = fruits.iterator();

        // Использование Iterator для перебора списка
        while (iterator.hasNext()) {
            String fruit = iterator.next();
            System.out.println(fruit);
        }
    }
}
```

Iterable — это интерфейс, позволяющий объекту возвращать Iterator, который может перебирать элементы. Iterator предоставляет методы для перебора коллекций.

Iterator тоже является интерфейсом.

В этом примере мы создаем список fruits и используем Iterator для перебора и вывода каждого фрукта.

Метод hasNext() проверяет наличие следующего элемента, а next() возвращает следующий элемент.



class Collections

```
package l37.slides.ex2;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

class CollectionsExample {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.add("Cherry");

        // Сортировка списка
        // Сортирует список в алфавитном порядке
        Collections.sort(list);
        System.out.println("Отсортированный список: " + list);

        // Перемешивание списка
        // Меняет порядок элементов в списке на случайный
        Collections.shuffle(list);
        System.out.println("Перемешанный список: " + list);

        // Реверс списка
        // Меняет порядок элементов в списке на обратный
        Collections.reverse(list);
        System.out.println("Список в обратном порядке: " +
list);

        // Заполнение списка одинаковыми элементами
        // Заполняет список одинаковыми значениями "Mango"
        Collections.fill(list, "Mango");
        System.out.println("Список после заполнения: " + list);

        // Добавление элементов для демонстрации swap
        list.add("Pineapple");
        list.add("Grape");

        // Обмен элементов
        // Меняет местами элементы с индексами 1 и 4
        Collections.swap(list, 1, 4);
        System.out.println("Список после обмена: " + list);

        // Бинарный поиск по списку
        int result = Collections.binarySearch(list,
"Pineapple");
        System.out.println("binarySearch: " + result);
    }
}
```



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