Списки

Составитель: Рощупкин Александр

Структуры данных

«Плохие программисты думают о коде. Хорошие программисты думают о структурах данных и их взаимосвязях», — Линус Торвальдс, создатель Linux

* **Структура данных** — это контейнер, который хранит данные в определенном макете. Этот «макет» позволяет структуре данных быть эффективной в некоторых операциях и неэффективной в других

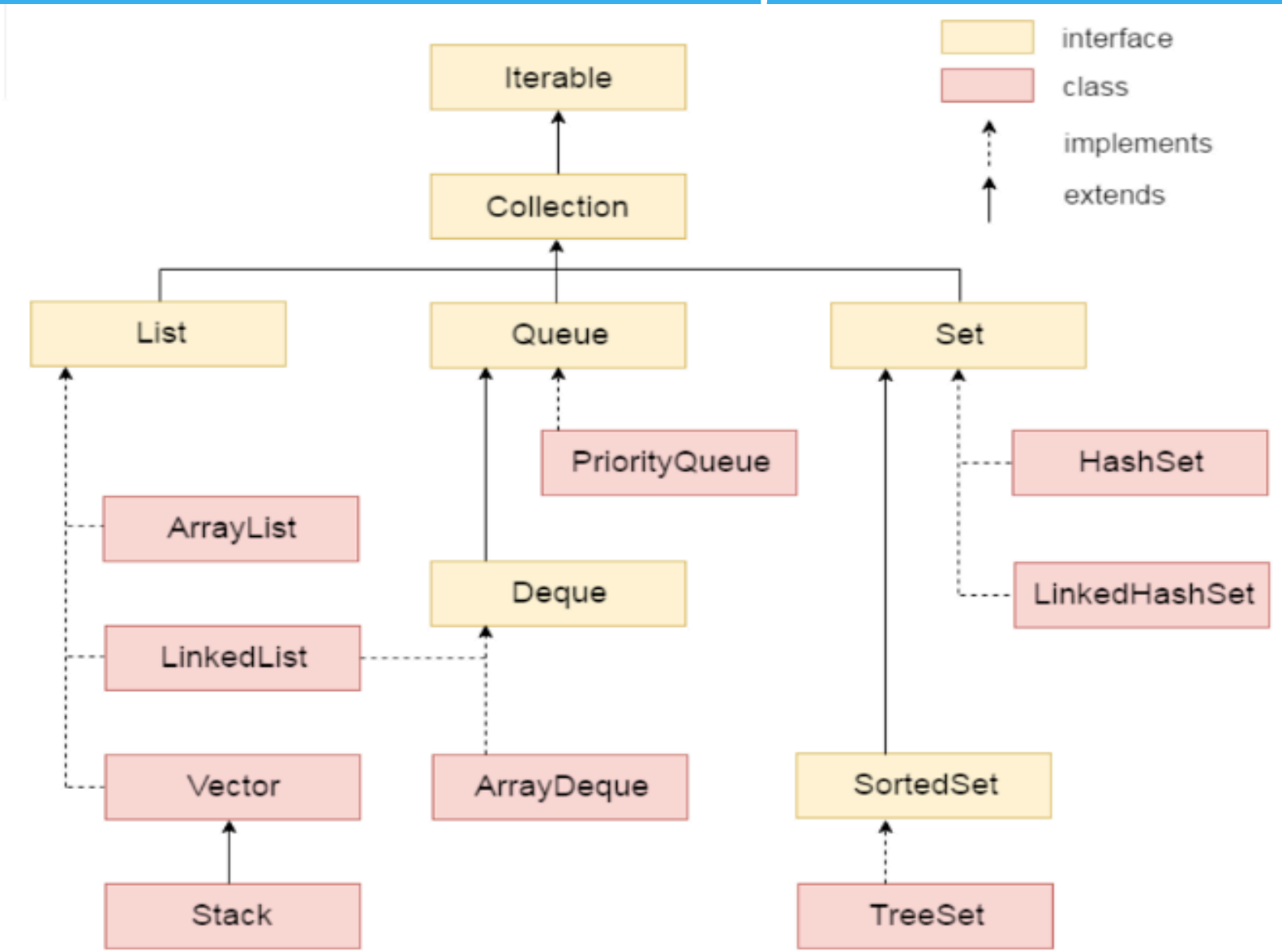
Основные структуры данных

* Массивы
* Множества
* Очереди
* Списки
* Связанные списки
* Графы
* Деревья
* Хэш таблицы

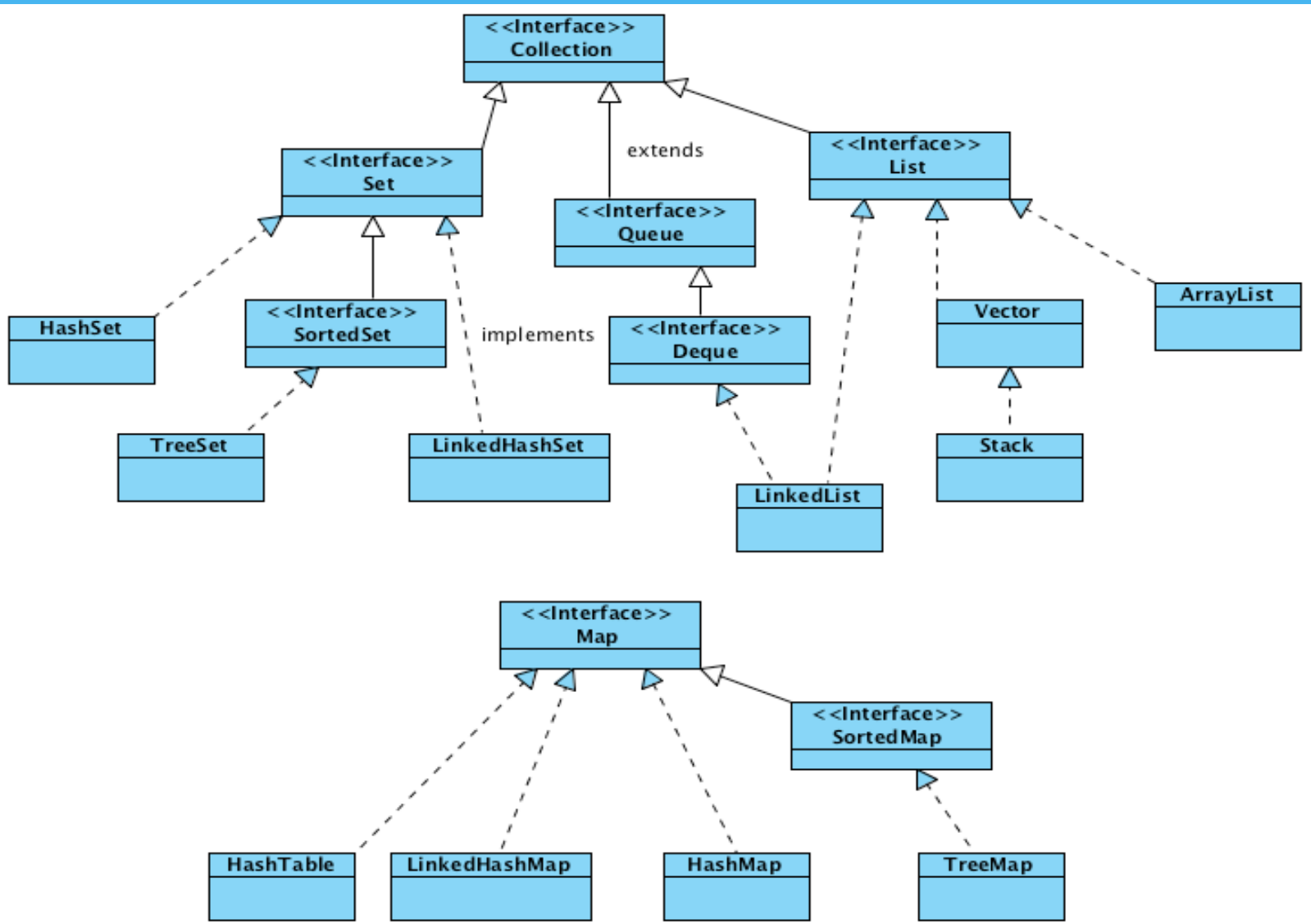
Коллекции

* Коллекции – это подбиблиотека в рамках стандартной библиотеки java, содержащая основные структуры данных и алгоритмы работы с ними

Коллекции


interface
class
implements
extends
Iterable
Collection
List
Queue
Set
PriorityQueue
HashSet
ArrayList
Deque
LinkedHashSet
LinkedList
Vector
ArrayDeque
SortedSet
Stack
TreeSet

Коллекции и карты



Основные элементы коллекций

* **Collection:** базовый интерфейс для всех коллекций и других интерфейсов коллекций
* **Queue:** наследует интерфейс Collection и представляет функционал для структур данных в виде очереди
* **Deque:** наследует интерфейс Queue и представляет функционал для двунаправленных очередей
* **List:** наследует интерфейс Collection и представляет функциональность простых списков
* **Set:** также расширяет интерфейс Collection и используется для хранения множеств уникальных объектов
* **SortedSet:** расширяет интерфейс Set для создания сортированных коллекций
* **NavigableSet:** расширяет интерфейс SortedSet для создания коллекций, в которых можно осуществлять поиск по соответствию
* **Map:** предназначен для созданий структур данных в виде словаря, где каждый элемент имеет определенный ключ и значение. В отличие от других интерфейсов коллекций не наследуется от интерфейса Collection

Список на основе массива

* Список – структура данных, хранит элементы последовательно
* Каждому элементу данных присваивается положительное числовое значение (индекс), который соответствует позиции элемента в массиве. Начальный индекс массива равен 0

Заготовка списка

```
public class VectorList {
    private int[] vector; // основной массив
    public static final int SIZE = 16; // размер массива
    private int length = 0; // виртуальная длинна

    public VectorList() {
        this.vector = new int[SIZE]; // создание реального массива
    }

    public int get(int index) {
        checkIndex(index); // проверка на выход за границы
        return vector[index];
    }

    private void checkIndex(int index) {
        if (index < 0 || index > length) {
            throw new IndexOutOfBoundsException(String.valueOf(index));
        }
    }
}
```

Дженерики

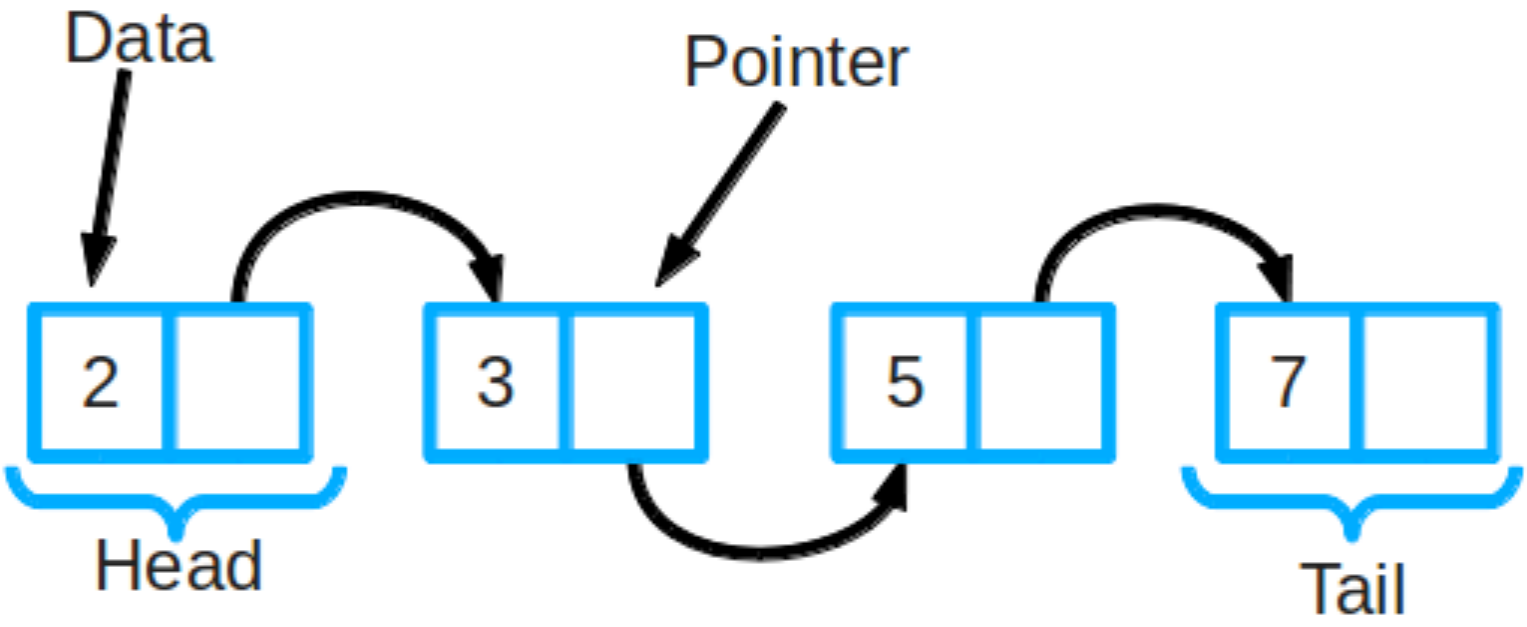
* Generics - переменная типа
* Создать класс пара, для хранения данных любого типа
* Generics
* - сырые типы, шаблоны, обобщения (приведение параметризированных и сырых типов)
* - переменные типа
* - дженерики и наследование
* - ограничение сверху (extends, можно A extends B & C & D, где C, D - интерфейсы)
* - ограничения снизу (? super T) тип должен быть либо предком, либо самим типом T

Параметризированный список

Связанный список

* Связный список состоит из группы узлов, которые вместе образуют последовательность. Каждый узел содержит две вещи: фактические данные, которые в нем хранятся (это могут быть данные любого типа) и указатель (или ссылку) на следующий узел в последовательности. Также существуют двусвязные списки: в них у каждого узла есть указатель и на следующий, и на предыдущий элемент в списке.

Связанный список



Параметризированный связанный список

Двусвязанный список