Хекслет

Коллекции в Java. ArrayList

Наставник: Теплинская Мария
Группа: java-167
Дата: 16.12.2022

Зачем нужны коллекции

В небольших приложениях программисту достаточно хранить данные в переменных или массивах. По мере усложнения программ этих конструкций становится недостаточно.

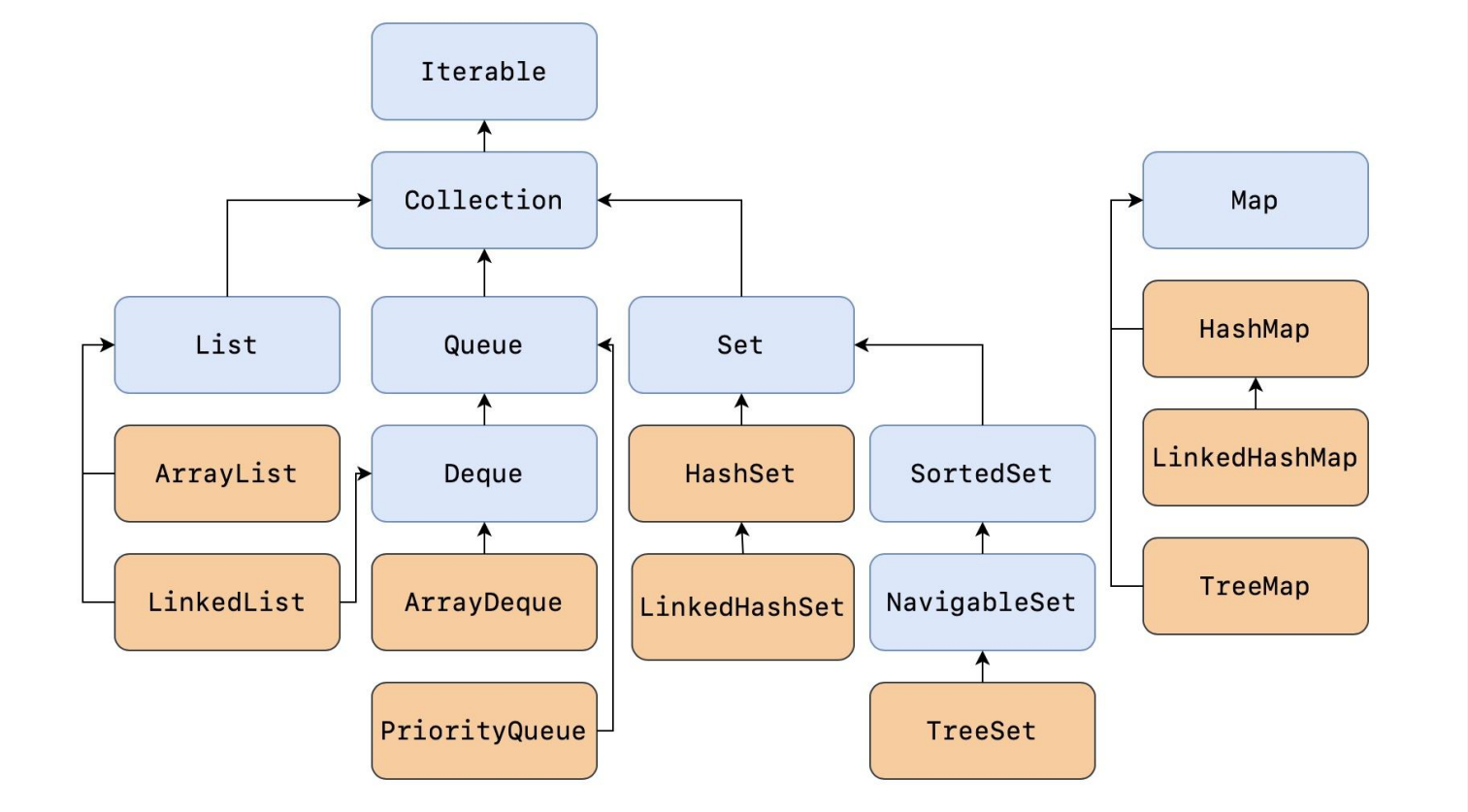
Данные, которыми оперируют большие программы, надо как-то организовывать. Для этого можно писать свои классы или использовать уже готовые. Идеологи и разработчики Java придумали и разработали свою библиотеку классов для работы с данными и назвали ее Java Collections Framework.

Что такое коллекции?

Java Collections Framework - это часть стандартной библиотеки Java (JDK), набор классов и интерфейсов, которые предоставляют реализацию самых популярных структур данных.

Коллекция - это некоторая группа однотипных элементов, организованная по определенным правилам. В Java это некая “надстройка” над структурами данных.

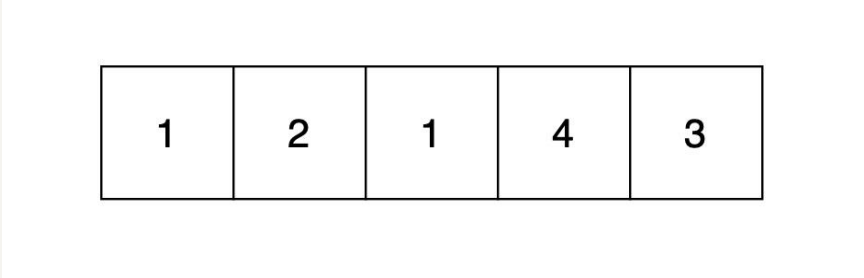
Иерархия коллекций



Проблематика

В чем заключается недостаток массива? Массив имеет *фиксированную* длину. Это может быть неудобно, если мы заранее не знаем, какое количество данных у нас будет.

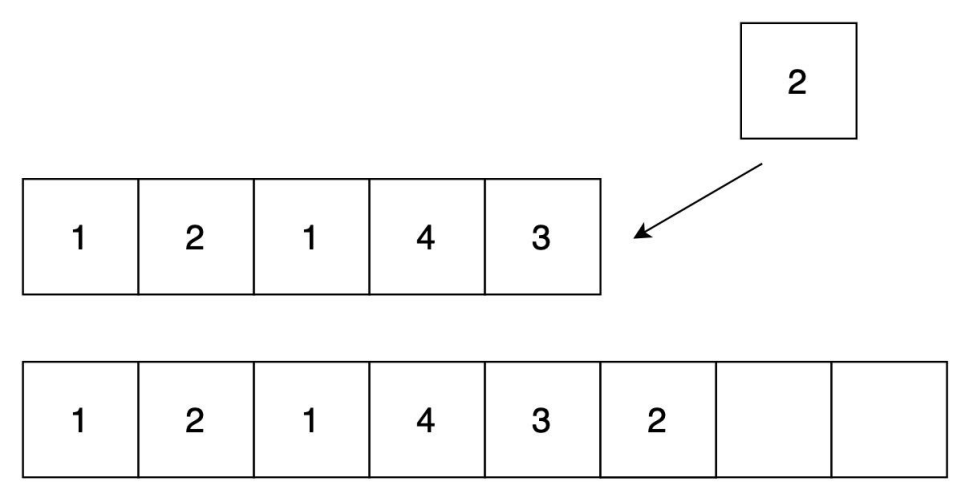
Массив - это структура данных довольно низкого уровня. Программисту самоятельно приходится следить за размером и поддерживать правильную последовательность элементов при вставке или удалении.



ArrayList

ArrayList - это класс (имплементация интерфейса List), который реализует динамический массив, то есть массив переменной длины.

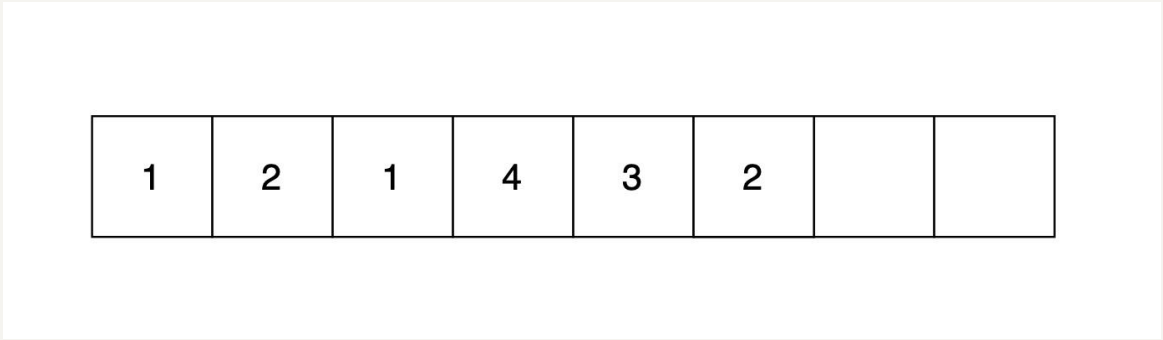
ArrayList может изменять свой размер в зависимости от количества элементов.



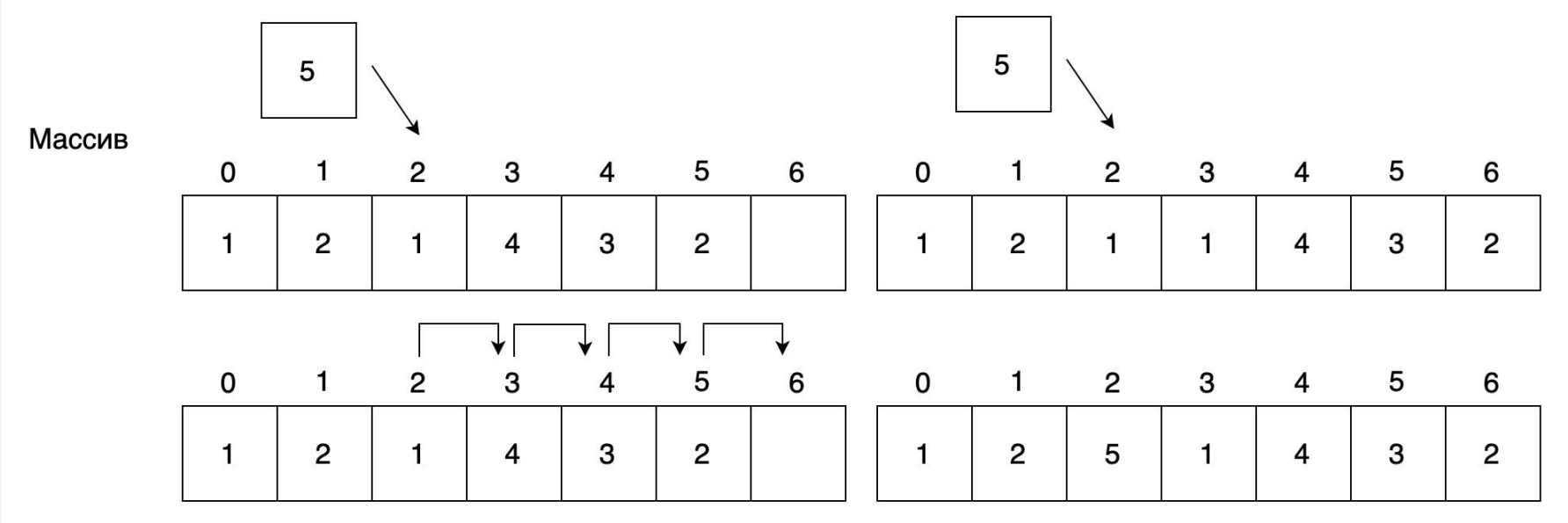
Структура ArrayList

Как следует из названия, ArrayList внутри, “под капотом”, представляет из себя обычный массив. У массива есть *емкость* (**capacity**) и *наполненность* (**size**).

Массив меняется, когда к ArrayList применяют операции модификации: добавление элемента, удаление элемента, увеличение или сжатие емкости.

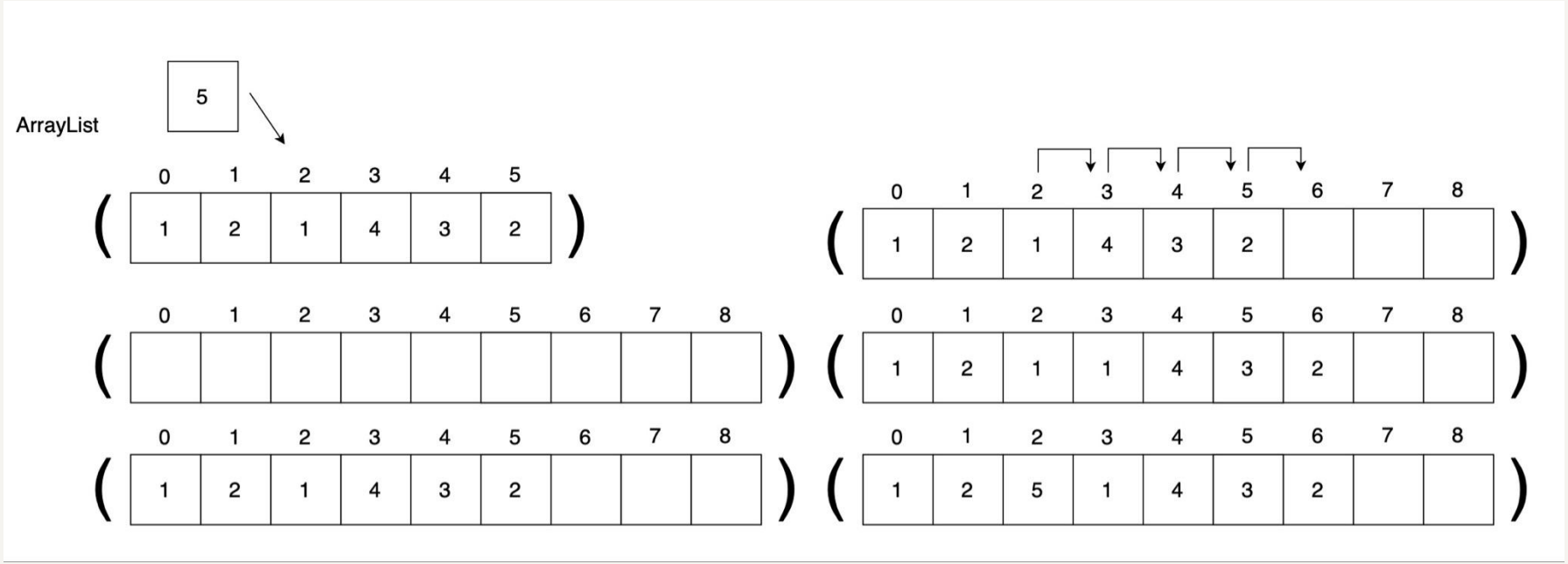


Добавление элемента в массив



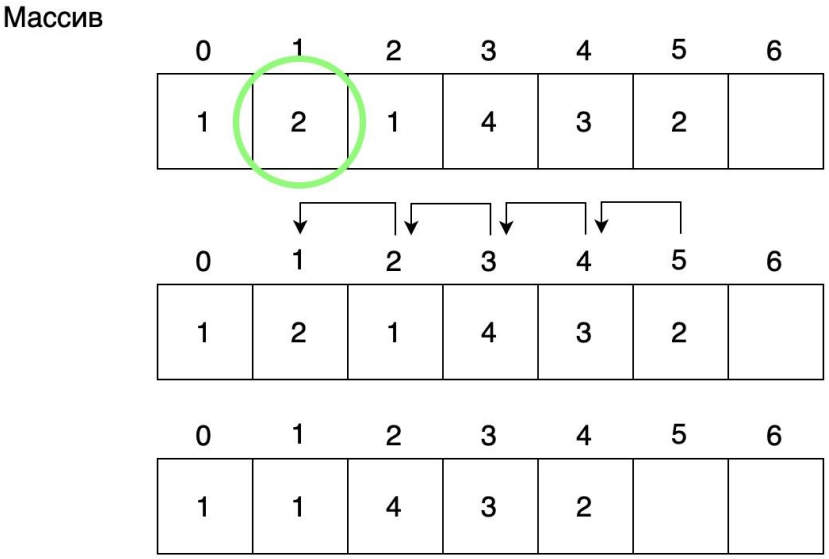
Сложность добавления элемента в обычный массив: O(n)

Добавление элемента в ArrayList



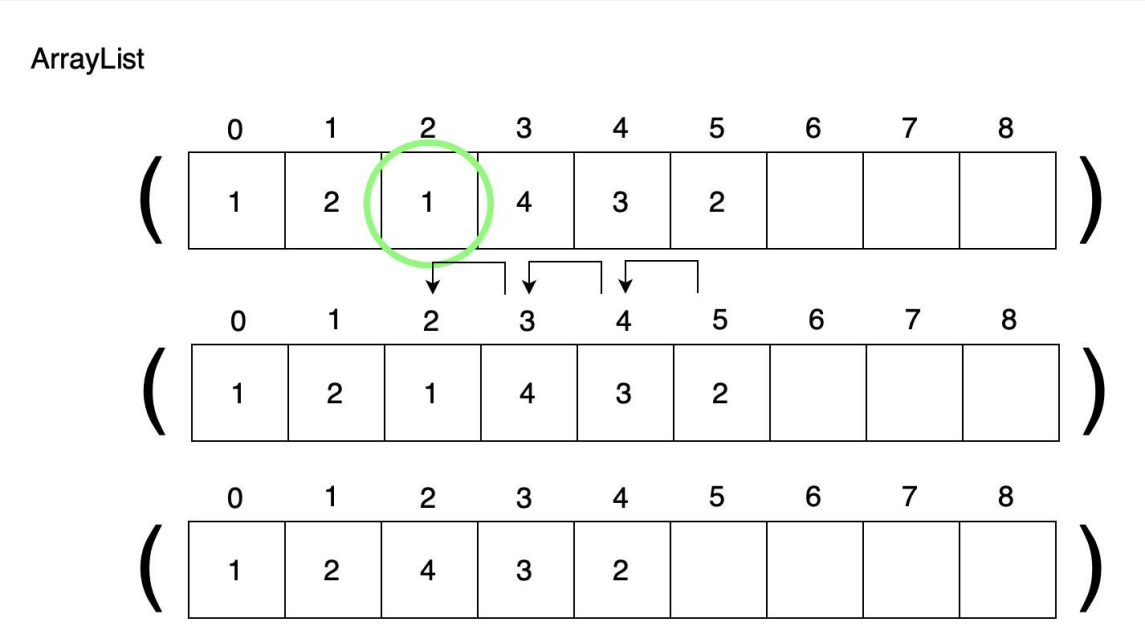
Сложность добавления элемента в ArrayList: O(1) + O(n) + O(n) + O(1) = **O(n)**

Удаление элемента из массива



Сложность удаления элемента из обычного массива: O(n)

Удаление элемента из ArrayList



Сложность удаления элемента из ArrayList: O(n)

После удаления ArrayList сам не сжимается. Для сжатия нужно использовать функцию trimtoSize(..)

Объявление ArrayList

```
List<Integer> list1 = new ArrayList<>();

List<Integer> list2 = new ArrayList<>( initialCapacity: 100);

List<Integer> list3 = new ArrayList<>(list2);
```

Основные операции

```
public boolean add(E e)
public boolean contains(Object o)
public E remove(int index)
public E get(int index)
public E set(int index, E element)
public int size()
public boolean isEmpty()
public boolean addAll(Collection<? extends E> c)
public boolean removeAll(Collection<?> c)
public boolean retainAll(Collection<?> c)
public void clear()
public Iterator<E> iterator()
public void ensureCapacity(int minCapacity)
public void trimToSize()
```

Итерация по ArrayList

```
for (int i = 0; i < list.size(); i++) {
    System.out.println(list.get(i));
}

for (Integer i: list) {
    System.out.println(i);
}

Iterator<Integer> it = list.iterator();
while (it.hasNext()) {
    System.out.println(it.next());
}
```

Сортировка ArrayList

```
for (int i = 0; i < list.size(); i++) {
    System.out.println(list.get(i));
}

Collections.sort(list);

for (Integer i: list) {
    System.out.println(i);
}
```

Связь с массивом

Методы:

```
public Object[] toArray()
public <T> T[] toArray(T[] a)
```

Использование:

```
Object[] listArray1 = list.toArray();


Integer[] numArray = new Integer[list.size()];
Integer[] listArray2 = list.toArray(numArray);
```

Связь с массивом (2)

Методы:

```
public static <T> List<T> asList(T... a)
```

Использование:

```
Integer[] numArray = {1, 2, 3, 4, 5};
List<Integer> numList = Arrays.asList(numArray);
```

Практическая задача

Есть текст. Надо заменить в этом тексте каждое слово

Заключение

Мы изучили одну из самых популярных коллекций *JCF* - ArrayList.

- ArrayList расширяет функционал обычного массива. Он более высокоуровневый, безопасный, и с ним приятно работать
- Он применяется практически повсеместно
- Предоставляет отличный уровень абстракции и хорош в плане быстродействия
- В ArrayList включено большое количество методов, которые могут решить любые задачи, возникающие при работе с данными

Полезные ссылки

- Java. Полное руководство - Герберт Шилдт
- Java. Справочник Разработчика - Эванс, Флэнаган (Интерфейс List)
- Алгоритмы. Построение и анализ - Корман, Лейзерсон, Ривест, Штайн
- Структуры данных в картинках. ArrayList: https://habr.com/ru/post/128269/
- Обзорная статья по структурам данных: https://habr.com/ru/post/696184/#ArrayList
- Документация: https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

Спасибо за внимание!