

Java Collections

Melvyn Ian Drag

November 4, 2019

Abstract

java.util provides many containers. These containers are widely used in Java programming and are implementations of the great data structures you hear about in data structures & algorithms classes. In today's lecture we'll have a look at a few of them and consider when we would want to use them.

1 Exam

Allow 30 minutes for students to comment their codes. Those who were unable to complete the midterm can comment a completed version for certain number of points.

2 Introduction

A java collection is a container you can use to store a bunch of values. For example, in your program you may need to store a bunch of ages - in this case you would

```
...  
int[] ageArray = {19, 25, 13, 41, 15};  
...
```

or you might create a class called 'Person' and when want to store a bunch of People. In this case you might

```
Person[] peopleArr = { new Person(), new Person(), ...};
```

in these two examples I am using arrays. While arrays are fine for simple collections of items, there are some distinct disadvantages of using arrays compared to using a more elegant container. There are some cases where using an array is extremely problematic. One major disadvantage is that the array size is fixed. Once you create an array of 5 elements , you cannot add a sixth person to the array.

There are many Java Collections. For example,

1. ArrayList
2. LinkedList
3. Vector

4. PriorityQueue
5. ArrayDeque
6. HashSet
7. LinkedHashSet
8. TreeSet
9. HashMap
10. TreeMap
11. LinkedHashMap
12. ConcurrentListSkipMap
13. WeakHashMap
14. Stack
15. etc.

We are going to focus on just a few of them today. Namely, we will compare

1. ArrayList
2. LinkedList

3 What is a list?

A list is an abstract idea - it's a bunch of objects in a row, like the arrays that we've seen so far. The list contains a bunch of items. You can delete items from it and add items to it. As I said before - big drawback of arrays is that the size of an array is fixed.

```
public class IntArrayCreator{
    public static void main(String[] args){
        // 1, 2, 3, 4
        int[] array1 = new int[]{1, 2, 3, 4};

        // 1, 2, 3, 4
        int[] array2 = {1, 2, 3, 4};

        // 0,0,0,0
        int[] array3 = new int[4];
    }
}
```

Note that all of the above arrays have four elements in them, and that cannot be changed! Sometimes that's what you want, sometimes that's not. Your call. If you want a variable number of elements in your container, use a List not an array.

4 What is an ArrayList?

4.1 overview

An ArrayList is a container like an array, but the size isn't fixed. You can put some elements of a specified type in them. See this example to create ArrayLists of different types:

```
import java.util.ArrayList;

public class ArrayListDemo{
    public static void main(String[] args){
        ArrayList<Integer> intArrayList = new ArrayList<Integer>();
        intArrayList.add(1);
        intArrayList.add(100);
        for( int i : intArrayList ){
            System.out.println(i);
        }

        ArrayList<MyClass> classList = new ArrayList<MyClass>();
        classList.add( new MyClass() );
        classList.add( new MyClass() );
        classList.add( new MyClass() );

        for( MyClass mc : classList ){
            System.out.println( mc.x );
        }
    }

    class MyClass{
        public int x = 1;
    }
}
```

WARNING! This code won't work:

```
...
ArrayList<int> intArrayList = new ArrayList<int>();
...
```

because you can't put primitive types in an ArrayList - only reference types (i.e. everything except primitives) You can't put any of these in an ArrayList

- byte
- float
- int
- double

but the "Byte", "Float", "Int" and "Double" types are okay, because these aren't primitives, they are classes.

5 Exercise: Create 5 different ArrayLists, each holding a different type.

Allow 10 minutes to ensure that everyone can use the arraylists successfully.

6 What is a LinkedList?

Draw a linked list on the board

7 The Collections Interface

Here is a good picture I found to explain the collections interface:

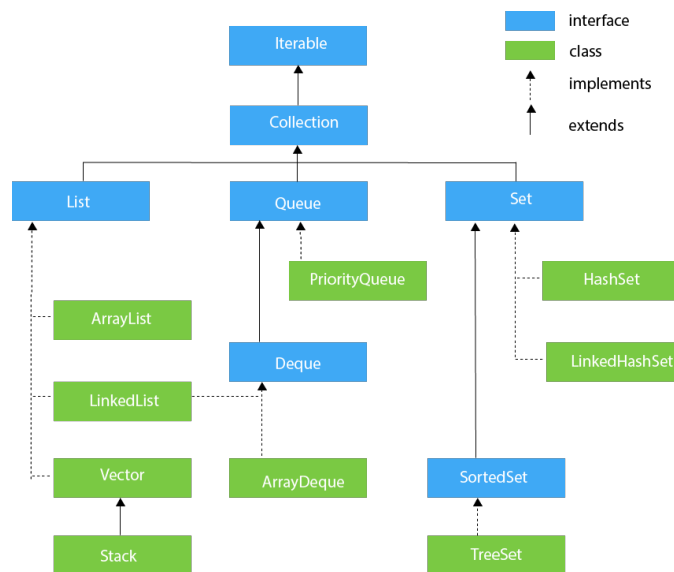


Figure 1: Illustration of a variety of classes which implement the Collection class.

7.1 Reference

<https://www.javatpoint.com/collections-in-java>

8 Initializing a List<T> with an ArrayList<T> or LinkedList<T>

Weird thing that Java programmers do. Java programmers use this pattern to achieve a bunch of complex things that you won't learn about in this class. Grab a good book and learn to write a Java Web App or Android App and you will see this pattern used in certain ways to achieve particular ends.

```
List<String> l = new ArrayList<String>(){
```

9 Timing Comparison

1. How to time code 2. Compare the following operations:

- add
- delete
- Check inclusion
-

10 Why Are Collections Useful?

In case I haven't made it abundantly clear through examples by now, collections are very useful and interesting. Here is the official pitch from Oracle about why Java collections are useful: <file:///home/melvyn/Desktop/JavaFall2019ClassRepo/ReadingMaterials/tutorial/collections/intro/index.html>

10.1 highlight these points

- reduces effort - you never need to code up a dynamically resizing array, because Java has one, for example
- increases quality - for example, the linked list implementation you are likely to write will be buggy and slow. The one in the collections framework has been tweaked and perfected by experts for a long time.
- Interoperability between different APIS - since we all agree to use standard collections, all our code can interact. If everyone used a different linked list implementation you couldn't pass data between different peoples' codes.

BTW If you don't know what an **API** is, it means **Application Programming Interface**. It pretty much means the publicly facing part of your code that other people can access. That may not be helpful - an example is warranted here.

There is a good chance that you know next to nothing about linked lists outside of what I've told you today. Even if you have studied algorithms and data structures, you probably implemented a very simplistic linked list. Nevertheless, today you learned how to add, and remove elements from a linked list. That is because it has a good API. You, the **Application Programmer**, only have to know a few functions, you don't need to understand the guts of the whole linked list code to use it. The API is the set of publicly accessible functions that application programmers will use.