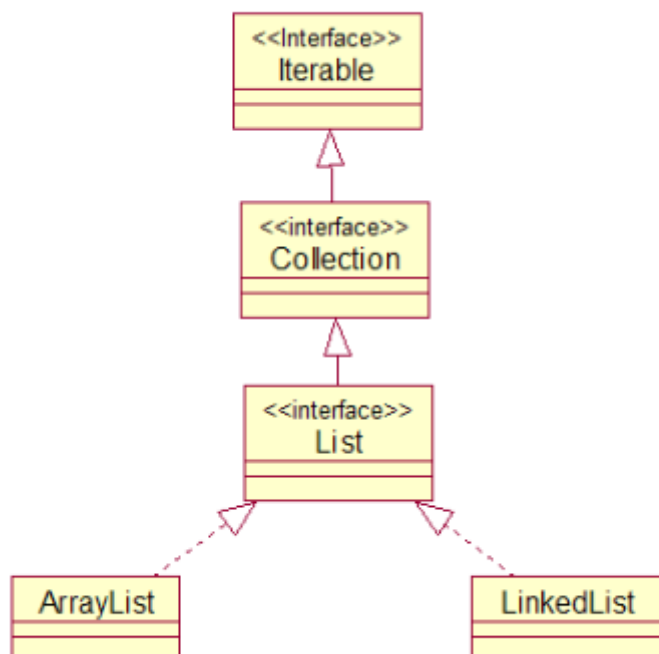


首先了解一下ArrayList和LinkedList在JDK中所在位置



ArrayList和LinkedList都是List接口的实现类，因此都实现了List的所有未实现的方法，只是实现的方式有所不同，这就是接口的好处。

ArrayList实现了List接口，它是以数组的方式来实现的，数组的特性是可以使用索引的方式来快速定位对象的位置，因此对于快速的随机取得对象的需求，使用ArrayList实现执行效率上会比较好。

Java代码

```
public class ArrayListDemo {

    public static void main(String[] args) {

        List<String> userlist = new ArrayList<String>();
        userlist.add("yulong");
        userlist.add("xiaoyun");
        userlist.add("羽龙共舞");
        System.out.println("使用普通for循环:");
        for(int i=0; i<userlist.size(); i++){
            System.out.print(userlist.get(i)+" ");
        }
        System.out.println();
        System.out.println();
        System.out.println("使用Iterator迭代器:");
        Iterator it = userlist.iterator();
        while(it.hasNext()){
            System.out.print(it.next()+" ");
        }
        System.out.println();
        System.out.println();
        System.out.println("使用增强for循环:");

        for(String s : userlist){
            System.out.print(s+" ");
        }
    }
}
```

第一种，是for循环，用get方法获取。第二种，用Iterator迭代器，使用next方法遍历。第三种是增强for循环，第三种方法是最方便，最简洁的。

LinkedList是采用链表的方式来实现List接口，它本身有自己特定的方法，如addFirst(), addLast(), getFirst(), removeFirst()等。由于采用的是链表实现的，因此在进行insert和remove动作在效率上要比ArrayList好得多。适合用来实现堆栈和队列，前者先进后出，后者先进先出。

在删除可插入对象的动作时，为什么ArrayList的效率会较低？？？

解析：因为ArrayList是使用数组实现的，若要从数组中删除或者插入一个对象，需要移动后段的数组元素，从而会重新调整索引顺序，调整索引顺序会消耗一定的时间，所以速度上就会比LinkedList要慢许多，相反，LinkedList是使用链表实现的，若要从链表中删除或者插入某一个对象时，只需要改变前后对象的引用就可以了。