CS206

ArrayList
Array

- An array is a sequenced collection of homogenous variables (elements)
- Each element of an array has an index
- The entire array is contiguous in memory
- The length of an array is fixed and cannot be changed
Array/List

- Dynamically-sized array
- Stores an ordered sequence of objects
  - Not sorted, ordered in the sense that arrays are ordered
- Can grow and shrink when items are added/removed
- Standard array features all supported, but with different syntax
Array/List

- ArrayList is implemented with an array
- A variable keeps track of the current size
  - initially it is equal to the actual size
  - deletion
    - elements are shifted to the left and size is decremented
  - addition, if not enough space
    - Create new, bigger array
    - Copy elements of old array into new one
Insertion

- In an operation \texttt{add}(i, o), we make room for the new element by shifting forward/to the right the elements \(A[i], \ldots, A[n - 1]\)
Deletion

• In an operation `remove(i)`, we fill the hole by shifting backward/to the left the elements $A[i + 1], ..., A[n - 1]$
### methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add(o)</code></td>
<td>appends o at the end of list</td>
</tr>
<tr>
<td><code>add(index, o)</code></td>
<td>inserting given o at index, shifting list to the right</td>
</tr>
<tr>
<td><code>get(index)</code></td>
<td>returns the object found at index</td>
</tr>
<tr>
<td><code>remove(index)</code></td>
<td>removes the object found at index and returns it, shifting list to the left</td>
</tr>
<tr>
<td><code>set(index, o)</code></td>
<td>replaces object at given index with o</td>
</tr>
<tr>
<td><code>size()</code></td>
<td>returns the number of elements in list</td>
</tr>
<tr>
<td><code>indexOf(o)</code></td>
<td>returns the first index where o is found, or -1</td>
</tr>
<tr>
<td><code>lastIndexOf(o)</code></td>
<td>returns the last index where o is found, or -1</td>
</tr>
<tr>
<td><code>clear()</code></td>
<td>removes all</td>
</tr>
</tbody>
</table>
Creation with Type Parameters

• When constructing an `ArrayList`, you must specify the type of elements via `<>

```java
ArrayList<String> l1 = new ArrayList<>();
ArrayList<Integer> l2 = new ArrayList<>();
```
Example usage

• Write a program to collect then print all unique words in a file

• Problem: you do not know the number of distinct words!
  • Solution
    • allocate a really big array
    • Use ArrayList!
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class WordCounter{
    public static void main(String[] args) {
        WordCounter wc = new WordCounter();
        wc.count("README");
    }
    void count(String filename){
        Arra206List<String> arrList = new Arra206List<>();
        try (Scanner input=new Scanner(new File(filename))){
            while (input.hasNextLine()) { // test if there is a line to read
                Scanner s2 = new Scanner(input.nextLine());
                while (s2.hasNext()) {
                    String word= s2.next();
                    if ( ) // word not in arrList
                        { arrList.add(word); }
                }
                s2.close();  }
        for (int i=0; i<arrList.size(); i++) System.out.println(arrList.get(i));
        } catch (FileNotFoundException e) {
            System.out.println("Error in opening the file:" + filename);
            System.exit(1);
        }}
}
java.util.ArrayList

- Implemented exactly as ours
- Part of Java collections framework
- import java.util.ArrayList
- Use this one rather than ours for Homework
Collections