ArrayList

James Brucker

Limitations of Arrays

In Java you can allocate space for an array at runtime:

```
numWords = console.nextInt();

String [ ] words = new String[numWords];
```

What if you don't know the **size of data** in advance?

Example: reading words from a file, but you don't know how many words are in the file?

After you create an array, you cannot change the size.

ArrayList

ArrayList is an alternative for variable size data

- ArrayList is an ordered collection of elements
- ArrayList grows and shrinks as needed!
- can add, delete, replace objects anywhere!
- **ArrayList** is a class in Java

```
ArrayList food = new ArrayList( );
food.size();  // returns 0.  Its empty
food.add("Apple");
food.add("Banana");
food.size();  // returns 2
System.out.println( food.get(0) ); // Apple
System.out.println( food.get(1) ); // Banana
```

List and ArrayList

List is a basic data type (not a class).

- in Java, *List* is an interface, not a class
- you cannot create "List" objects

ArrayList is a class that behaves like a *List*

- you can ignore "List" for now

Copying ArrayList to Array

- Use an ArrayList to save data when you don't know how big the data set is.
- After saving all data, copy to an array

```
ArrayList<String> list = new ArrayList<String>( );

... read all the data and save in list

// create an Array large enough to store the data
String [ ] words = new String[ list.size( ) ];
// copy ArrayList to Array
list.toArray( words );
```

Useful ArrayList Methods

- int size( )   returns actual size of ArrayList
- add(Object obj)   add an object to ArrayList
- add(int k, Object obj) add object at specific position
- Object get(int index)   get object at given index
- Object remove(int i)   get object and delete from ArrayList
- clear( )   remove all objects from ArrayList
- set(int index, Object obj) replace object at index
- contains(Object obj)   "true" if obj is in ArrayList
- ensureCapacity(int size) make sure ArrayList can hold at least this many elements without reallocation

ensureCapacity( ) is used to improve efficiency when you are adding a lot of items to an Arraylist.

Merging Data (Union)

- Read words from the input and form the union of all data.
- You must eliminate duplicate words.
- Sort the list of words and output them

```
ArrayList wordlist = new ArrayList( );
BufferedReader input = openFile("wordlist.txt");
while( input.ready( ) ) {
    String s = input.readLine( ).trim( );
    if ( ! wordlist.contains( s ) ) wordlist.add( s );
}
// copy to an array so we can sort
String [ ] words = new String [ wordlist.size( ) ];
wordlist.toArray( words ); /* copy to array */
Arrays.sort( class );
```

Sorting

To sort an ArrayList use the java.util.Collections class

- Collections.sort(anyList)
- list must contain objects that are *Comparable*
  - String, Double, Long, Int, Date…
  - any class that has a compareTo method

```
ArrayList<String> alist = new ArrayList<String>( );


Collections.sort( alist );
```

More Information

- *Core Java, Volume 1.*
- Java API documentation.
- Java Tutorial