Array List

Array List

Stony Brook University

The ArrayList Class

- ArrayList is a Java ***collection*** that removes some of the limitations of arrays.

- An array’s size is set at the time it’s created; an ArrayList can grow to hold more items.

- An ArrayList still maintains constant-time insertion(update) and access.

- Java collection framework (JCF) is set of classes and interfaces that implement commonly reusable collection data structures.

The ArrayList Class contd..

- $ArrayList<T>$ ()
  - T must be an object type!
  - Ex. $ArrayList<String>$()

- $ArrayList<T>$ ($int\ n$)
  - Creates an $ArrayList$ of type T with space for n elements

Examples

- $ArrayList < String > \ city \ = \ new \ ArrayList < String > ();$

- $ArrayList < String > \ bush \ = \ new \ ArrayList < String > \ (20);$

- $ArrayList < Integer > \ number \ = \ new \ ArrayList < Integer > ();$

Adding Elements

- $add\ (T\ v)$

  appends v to the end of the $ArrayList$

- $add\ (int\ i, T\ v)$

  Inserts v at position i in the $ArrayList$ Shifts elements at i ... down one space

Insertion Examples

- $city.add(“Madrid”);$
- $city.add(“Cairo”);$
- $bush.add(“Forsythia”);$
- $bush.add(“Azalea”);$
- $number.add(2);$
- $number.add(0,3);$ // puts 3 in position 0

Access Methods

- $get\ (int\ i)$
  - Returns the value at position i
  - Throws exception if i is not valid

- $remove\ (int\ i)$
  - Removes and returns value at position i
  - Shifts elements $i + 1$ ... left to fill space

Access Example

- $String\ value\ =\ city.get(1);$
- $System.out.print(city.get(0));$
- $int\ temp\ =\ 2\ *\ number.get(0).intValue();$
- $number.remove(1);$

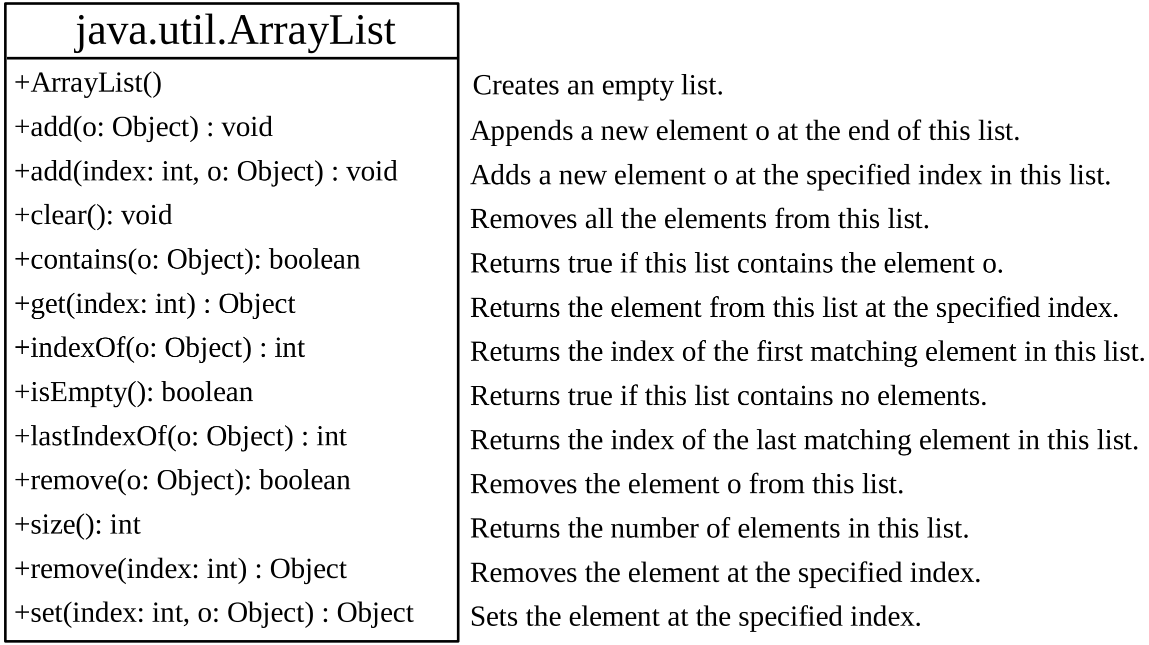
Other Useful Methods

- $size()$
  - Returns # of elements in the $ArrayList$

- $clear()$
  - Removes all elements from the $ArrayList$

The ArrayList Class

- You can create arrays to store objects - But the array’s size is fixed once the array is created. **Java provides the <u>java.util.ArrayList</u>** class that can be used to store an unlimited number of objects:



java.util.ArrayList	
+ArrayList()	Creates an empty list.
+add(o: Object) : void	Appends a new element o at the end of this list.
+add(index: int, o: Object) : void	Adds a new element o at the specified index in this list.
+clear(): void	Removes all the elements from this list.
+contains(o: Object): boolean	Returns true if this list contains the element o.
+get(index: int) : Object	Returns the element from this list at the specified index.
+indexOf(o: Object) : int	Returns the index of the first matching element in this list.
+isEmpty(): boolean	Returns true if this list contains no elements.
+lastIndexOf(o: Object) : int	Returns the index of the last matching element in this list.
+remove(o: Object): boolean	Removes the element o from this list.
+size(): int	Returns the number of elements in this list.
+remove(index: int) : Object	Removes the element at the specified index.
+set(index: int, o: Object) : Object	Sets the element at the specified index.

Arrays Vs ArrayList

Operation	*Array*	*ArrayList*
Creating an array/ArrayList	`String[] a = new String[10]`	`ArrayList<String> list = new ArrayList<>();`
Accessing an element	`a[index]`	`list.get(index);`
Updating an element	`a[index] = "London";`	`list.set(index, "London");`
Returning size	`a.length`	`list.size();`
Adding a new element		`list.add("London");`
Inserting a new element		`list.add(index, "London");`
Removing an element		`list.remove(index);`
Removing an element		`list.remove(Object);`
Removing all elements		`list.clear();`

Array Lists from/to Arrays

- Creating an ArrayList from an array of objects:
  - $String[]\ array\ =\ \{\text{"red"}, \text{"green"}, \text{"blue"}\};$
  - $ArrayList < String >\ list\ =\ \boldsymbol{new}\ ArrayList <> (Arrays.asList(array));$
- Creating an array of objects from an ArrayList:
  - $String[]\ array1\ =\ \boldsymbol{new}\ String[list.size()];$
  - $list.toArray(array1);$

max and min in an Array List

- String[] array = {**"red"**, **"green", "blue"**};
- System.out.pritnln(java.util.Collections.max(new ArrayList<String>(Arrays.asList(array)));

```
String[] array = {"red", "green", "blue"};

System.out.pritnln(java.util.Collections.min(new
    ArrayList<String>(Arrays.asList(array)));
```

Shuffling an Array List

- Integer[] array = {**3**, **5**, **95**, **4**, **15**, **34**, **3**, **6**, **5**};
- ArrayList<Integer> list = **new** ArrayList<>(Arrays.asList(array));
- java.util.Collections.shuffle(list);
- System.out.println(list);

Stony Brook
University