

Lesson Java: What is an ArrayList?

Arraylists

- An **arraylist** is a **dynamic** data structure. It is dynamic because its size is not fixed and it changes according to the number of elements it holds. The array is a **static** data structure.
- A second important difference between an array and an arraylist is that the array holds elements of the same type while the arraylist can hold elements of different types.
- The elements of an arraylist are objects. We cannot store a primitive value in an arraylist e.g. if we add an integer in an arraylist the added element will be of type **Integer**.
 - An Integer (note: uppercase I) is an object containing only one value which is of type int.
- As in arrays the indices of an arraylist start from zero.

Some Basic Arraylist Functions

- **add**
 - Assume al is an arraylist having the elements ("Mark", 8, 3, "Stephanie"). The statement al.add (-5) will add an element at the end of the arraylist. The new al is ("Mark", 8, 3, "Stephanie", -5)
 - Now suppose we now perform al.add (1, "George"). This will add "George" at index 1 moving all the necessary elements by one position. The new arraylist al is now ("Mark", "George", 8, 3, "Stephanie", -5).
- **remove**
 - continuing on the previous arraylist, al.remove (2) will eliminate the element with index 2. So the new formed al will consist of ("Mark", "George", 3, "Stephanie", -5).

- al.remove ("Stephanie") will remove the first occurrence of "Stephanie" (i.e. it will only remove the value with the least index). So now al consists of ("Mark", "George", 3, -5).
- **size**
 - This is a function that gives you the number of elements in an arraylist.
- **get**
 - This function is used to read an element from an arraylist.

A class that uses an ArrayList

```

1  import java.util.ArrayList;
2
3  public class AboutArraylists
4  {
5      static void main (String[ ] args)
6      {
7          ArrayList ListTest = new ArrayList();
8
9          ListTest.add ("James");
10         ListTest.add ("Anne");
11         ListTest.add ("Isabel");
12         ListTest.add (7);
13         ListTest.remove (2);
14         ListTest.add ("Paul");
15         ListTest.add (1, "Mary");
16         ListTest.remove ("Anne");
17         ListTest.remove ("Paul");
18         int x;
19         x = (int) ListTest.get (2); // (int) is called casting
20         System.out.println (x+5);
21     }
22
23     static void ViewArraylist (ArrayList al)
24     {
25         int l = al.size();
26         for (int i = 0; i<l; i++)
27             System.out.print (al.get(i) + " ");

```

```
28     System.out.println();
29     }
30 }
```

Notes on the above program

- Line 1: the use of arraylists requires the import shown in this line
- Line 7: this line shows the **declaration** and **creation** of an arraylist.
- Line 19: here **casting** is used. Casting changes the type of a value before it is assigned to a variable.

Some Other ArrayList Functions

- void **clear**(): Removes all of the elements from this list.
- boolean **contains** (Object o): Returns true if the arraylist contains the specified element.
- int **indexOf** (Object o): Returns the index of the first occurrence of the specified element, or -1 if the List does not contain this element.
- int **lastIndexOf** (Object o): Returns the index of the last occurrence of the specified element, or -1 if the list does not contain this element.
- Object **set** (int index, Object element): Replaces the element at the specified position in this list with the specified element.

Uni-type Arraylists

The following program shows how it is possible to create an arraylist that accepts only elements of one type.

Class that uses an arraylist with elements of the same type	
1	import java.util.ArrayList;
2	
3	public class ArrayListOther
4	{
5	public static void main (String[] args)
6	{
7	ArrayList <String> members = new ArrayList <String> ();

```

8
9 members.add("Graham");
10 // the statement members.add(8) is not accepted
11
12 ArrayList <Integer> memNum = new ArrayList<Integer>();
13 memNum.add(7);
14 // the statement memNum.add("Java") is not accepted
15 // the statement memNum.add(3.6) is not accepted
16 double r = 3.6;
17 memNum.add( (int) r );
18 int i;
19 i = memNum.get(0);
20 }
21 }

```

Exercise

- 1) The following statements have been applied on an arraylist called myList (initially empty). What will myList contain at the end of the execution of the statements?
 - a) myList.add ("woman");
 - b) myList.add (8);
 - c) myList.add ("dress");
 - d) myList.add ("car");
 - e) myList.add (6);
 - f) myList.remove(0);
 - g) myList.add ("dress");
 - h) myList.add (2, "mobile");
 - i) myList.remove ("dress");
 - j) int x;
 - k) x = (int)myList.get(3);
 - l) myList.add(x*5);

- 2) Write a method that given a string and a string arraylist as parameters it adds the number of occurrences of the string in the arraylist.

- 3) Write a method that given a string and a string arraylist as parameters deletes all occurrences of the string in the arraylist.