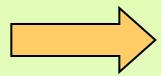


Outline



ArrayList

The ArrayList Class

- An **ArrayList** object stores a list of objects
- An **ArrayList** object grows and shrinks as needed, adjusting its capacity as necessary
- The **ArrayList** class is part of the `java.util` package
- You can reference each object in the list using a numeric index just like the arrays

The ArrayList Class

- Index values of an ArrayList begin at 0 (not 1):

0	"Bashful"
1	"Sleepy"
2	"Happy"
3	"Dopey"
4	"Doc"

- Elements can be inserted and removed
- The indices of the elements adjust accordingly

The ArrayList Class

```
ArrayList<String> cityList = new ArrayList<String>();  
  
//Add some cities in the list  
cityList.add("London");  
// cityList now contains [London]  
cityList.add("Denver");  
// cityList now contains [London, Denver]  
cityList.add("Paris");  
// cityList now contains [London, Denver, Paris]  
cityList.add("Ankara");  
// cityList now contains [London, Denver, Paris,  
Ankara]
```

- See [TestArrayList.java](#)

ArrayList Methods

- **ArrayList()**
Creates an empty list.
- **add(o: Object) : boolean**
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.

ArrayList Methods

- `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.

ArrayList Methods

- `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): boolean`
removes the element at the specified index.
- `set(index: int, o: Object): Object`
sets the element at the specified index.

The ArrayList Class

- The type of object stored in the list is established when the ArrayList object is created:

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

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

- An ArrayList object cannot store primitive types, but wrapper classes are used instead.
- See [Beatles.java](#)

```
//*****  
//  Beatles.java      Author: Lewis/Loftus  
//  
//  Demonstrates the use of a ArrayList object.  
//*****  
  
import java.util.ArrayList;  
  
public class Beatles  
{  
    //-----  
    // Stores and modifies a list of band members.  
    //-----  
    public static void main (String[] args)  
    {  
        ArrayList<String> band = new ArrayList<String>();  
  
        band.add ("Paul");  
        band.add ("Pete");  
        band.add ("John");  
        band.add ("George");  
    }  
}
```

continue

continue

```
System.out.println (band);
int location = band.indexOf ("Pete");
band.remove (location);

System.out.println (band);
System.out.println ("At index 1: " + band.get(1));
band.add (2, "Ringo");

System.out.println ("Size of the band: " + band.size());
int index = 0;
while (index < band.size())
{
    System.out.println (band.get(index));
    index++;
}
}
```

continue

```
System.out.println(band);
int location = band.indexOf("John");
band.remove(location);

System.out.println(band);
System.out.println("At index " + location + ": " + band.get(location));
System.out.println("Size of the band: " + band.size());

System.out.println(band);
band.add("Ringo");
System.out.println(band);
System.out.println("Size of the band: " + band.size());

int index = 0;
while (index < band.size())
{
    System.out.println(band.get(index));
    index++;
}
}
```

Output

[Paul, Pete, John, George]

[Paul, John, George]

At index 1: John

Size of the band: 4

Paul

(1));

John

Ringo

George

nd.size());

Examples

- See [ReverseFile.java](#)
- See [CountVowels.java](#)
- See [CountWords.java](#)