ArrayList Operations

Introduction to OO Programming

ArrayList Basics

- The `ArrayList` class is designed to store a series of objects of the same type.
- An `ArrayList` is a resizable container or collection class which provides methods for the manipulation and storage of its contents.
- An `ArrayList` can only contain objects. It cannot contain primitives.

ArrayList Basics

- Like an array, an `ArrayList` has an index which starts at 0. This allows you to refer to any element in an `ArrayList`.
- The index in an `ArrayList` goes from 0 to n – 1 where n is the number of elements in the `ArrayList`.
- To declare `ArrayList` objects, you need to import the class from the package `java.util`.

```
import java.util.ArrayList;
ArrayList<BankAccount> accounts=new ArrayList<BankAccount>();
```

Syntax :
The "for each" Loop or ***enhanced for*** loop

```
for (Type variable : collection)
    statement;
```

Example 1: Array

```
for (double d : data)
    sum = sum + d;
```

Example 2: ArrayList

```
for (BankAccount b : accounts)
      sum = sum + b.getBalance();
```

Purpose:

To execute a loop for each element in the collection. In each iteration, the variable is assigned the next element of the collection. Then the statement is executed.

To remove an element from ArrayList

- We do ***not*** need a for loop to remove an object
- However if we wish to find an object, then remove it, we need a for loop to find the object first.

```
System.out.print("Enter employee ID: ");
targetIdNo = keyIn.nextInt();
//must find employee first - then remove
for(int i = 0; i < empList.size(); i++)
{
        e = empList.get(i);
        if(targetIdNo == e.getEmpNo())
        {
                empList.remove(i);
                break; //NB -break out of loop when found
        }
}
```

To remove an element from ArrayList

- Using for each

```
System.out.print("Enter employee Id no: ");
int targetIdNo = keyIn.nextInt();

//for each employee in list
for(Employee e: empList)
{
   if(e.getEmpNo() == targetIdNo)//if i find a match
   {
       empList.remove(e);
       break; //NB - break out of loop when match found
    }
}
```

Simple Algorithms

- The same algorithms used with arrays for counting, searching and finding highest and lowest can also be applied to `ArrayList`

Counting occurrences

- Count all words beginning with a given letter.

- Algorithm

```
FOR each word in wordList
    IF word starts with 'letter'
        ADD 1 to count
DISPLAY count
```

```
System.out.print("Enter a letter : " );
letter = keyIn.next().charAt(0) ;
count = 0;  //reset count
for(String s : wordList) //for each word in wordlist
{
     if(s.charAt(0) == letter)
     {
          count++;
     }
}
System.out.print(count +" words begin with "
          +letter +". ");
```

This will not handle case differences

```
System.out.print("Enter a letter : " );
letter = keyIn.next().charAt(0) ;
count = 0;  //reset count
for(String s : wordList) //for each word in wordlist
{
     if(s.toLowerCase().charAt(0) ==
          Character.toLowerCase(letter))
     {
          count++;
     }
}
System.out.print(count +" words begin with "
          +letter +". ");
```

This will handle case differences

Counting occurrences

- Count all bank accounts with a balance greater than some amount
- Algorithm

FOR each BankAccount b in accountList
IF b's balance is greater than 'someAmount'
ADD 1 to count
DISPLAY count

Counting Occurrences

- Check all elements and count the occurrences until you reach the end of the array list.

```
int count= 0;
for (BankAccount b : accounts)
{
   if (b.getBalance() >= atLeast)
        count++;  // Found a match
}
```

Finding the Maximum or Minimum

- Find the longest word in wordList

Assign length of first word to ‘longestYet’
FOR each word in wordList
IF ‘longestYet’ < word’s length
‘longestYet’ is assigned word’s length
DISPLAY ‘longestYet’

```
//length of 1st word in list for comparisons
int longestYet = wordList.get(0).length();
for(String s : wordList) //for each string in list
{
     if(longestYet < s.length())
          longestYet = s.length();
}
System.out.print ("Longest word has length "
          +longestYet);
```

If I want to find longest word

```
//length of 1st word in list for comparisons
String longestYet = wordList.get(0);
for(String s : wordList) //for each string in list
{
      if(longestYet.length() < s.length())
            longestYet = s;
}
System.out.print (longestYet" is Longest
word has length "  +longestYet.length());
```

Another version – store index

```
int longIndex = 0;
for(int i = 0; i < wordList.size(); i++)//for each
{
      if(wordList.get(i).length() >
            wordList.get(longIndex).length())
                  longIndex = i; //remember i

}

System.out.println(wordList.get(longIndex) + " has "
+wordList.get(longIndex).length() + " letters"  );
```

Finding the Maximum or Minimum

- When processing elements in an array list we may want to see the details of the element that fulfills the condition
  - i.e
    - the name of the employee with the highest salary
    - The account details of the account with lowest balance
      - It may not be enough to see the lowest balance or the highest salary

Find max and min

```
//find highest and lowest
BankAccount acc = accounts.get(0);
double highest = acc.getBalance();
double lowest = highest;
//get each element in turn
for(BankAccount a: accounts)
{
      //check if a's balance is higher than highest
      if(a.getBalance() > highest)
            highest = a.getBalance();
      if(a.getBalance() < lowest)
            lowest = a.getBalance();
}
…
```

This will only tell us the highest and lowest balances, not who owns them

Finding the Maximum or Minimum

- Initialize an object reference (e.g. `largestYet)` with an element in the list
- Compare `largestYet` with remaining elements
- Update `largestYet` if you find a larger (or smaller) value

Finding the Maximum or Minimum

- Find the BankAccount with highest balance

```
Assign first BankAccount to 'largestYet'
FOR each BankAccount b in accountList
 IF largestYet's balance < b's balance
        'largestYet' is assigned b
DISPLAY 'largestYet'
```

Finding the Maximum or Minimum

- Example: find bank account with highest balance

```
BankAccount largestYet = accounts.get(0);
for (BankAccount a : accounts)
{
   if (a.getBalance() > largestYet.getBalance())
      largestYet = a;
}
```

`largestYet` is a BankAccount object

Finding the Maximum or Minimum

- Works only if there is at least one element in the array list
- If list is empty display error message ….

```
if (accounts.size() == 0)
      System.out.println("No accounts in bank");
else //go ahead and find largest
{
      BankAccount largestYet = accounts.get(0);
      . . .
}
```

`isEmpty()`

- Could also use `isEmpty()`
- If list is not empty find largest

```
if (accounts.isEmpty())
       System.out.println("No accounts in bank");
else //go ahead and find largest
{
       BankAccount largestYet = accounts.get(0);
       . . .
}
```

Searching

- Often want to find an object in `ArrayList` so that we can process it further
  - find an employee
    - to give a raise
    - to display details
    - to remove if leaving
  - find a bank account
    - to display balance
    - to withdraw/deposit funds
    - to delete/remove it if closed

Sample algorithm

FOR each element in collection/list

IF object found

PROCESS object

SET foundFlag to true

IF foundFlag is false

DISPLAY not found

Finding an object

- Check all elements until you have found a match. Should also determine if no match found

```
boolean found = false;  //assume not found
for (BankAccount a : accounts) //for each account in accounts
{
   if (a.getAccountNumber() == targetNo) // Found a match
   {
       a.deposit(1000) ; //some code here … deposit/withdraw..
       found = true;    //flag as found
   }
}
if(!found)  // IF not found - No match in the entire array list
     System.out.print("no match found");
```

Finding a Value

- If we need to know position of object in the list it may be more appropriate to use traditional for loop

```
pos = -1;  //set to invalid index value
for (int i = 0; i < accounts.size(); i++)
{
   BankAccount a = accounts.get(i);
   if (a.getAccountNumber() == targetNo) // Found a match
   {
      pos = i;
   }
}
if(pos == -1)  // No match in the entire array list
   System.out.print("no match found");
```

Other useful methods in ArrayList

boolean	**contains**(**Object** o)Returns true if this list contains the specified element.
void	**clear**()Removes all of the elements from this list.
int	**indexOf**(**Object** o)Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
boolean	**isEmpty**()Returns true if this list contains no elements.
Object[]	**toArray**()Returns an array containing all of the elements in this list in proper sequence (from first to last element).