ISSN 2278 – 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol. 5 , No.1 Pages : 18 - 21 (2016)

Special Issue of ICCEIT 2016 - Held on February 16, 2016 in The Solitaire Hotel, Bangalore, India

http://warse.org/IJATCSE/static/pdf/Issue/icceit2016sp05.pdf

Sharing of data between two different types of data collection in Java

Vijayakumar C M, Kumudavalli.M.V, Raniket Ram

Dept. of Computer Applications,
Dayananda Sagar College of Arts, Science and Commerce, Bangalore, India.

Abstract: It is an object that is used to collect multiple elements into a single unit. Data collection is used to store information on targeted variables in systematic manner. Sharing of data from an ArrayList to a two - dimensional array. Computer programs often manage many objects of the same type, e.g., a bank's accounting program must manage hundreds of customer accounts. It is inconvenient and usually impossible to declare distinctly named variables for each of the customer accounts; instead, one constructs a new form of object---a data structure---to collectively hold and name the customer accounts.

Key words :.

INTRODUCTION

There are many types of data collection technique in java such as Array, ArrayList, HashMap, Linked list and so on. Here my program is to establish a relationship between two different types of data collection i.e. Array and ArrayList. There are many different types of Array in java such as single-dimensional array, two-dimensional array and multi-dimensional array. They can store elements of same data type. But they are static in nature[1][2]. That means when you declare an array you declare them with its size that tells how many element an array can hold. When you add items to the array at run time and the size of the array exceeds it throws an exception (ArrayIndexOutOfBoundsException) or generates an error. When we encounter such kind of problems ArrayList comes in handy. There are many situations in application when we are not sure that how many elements we are going to store in our data collection. So java provides special class for re-sizable array named ArrayList. That means you can add or remove elements at run time unlike simple Array[3][4]. You declare ArrayList as an instance of a class so it's complex object.

IMPLEMENTATION

Here is the program that establishes a relationship between the ArrayList and a simple Array(two-dimensional Array). Firstly, all the data are stored into the ArrayList then all the data form ArrayList is stored into two – dimensional array.

```
import java.util.ArrayList;
import java.util.ListIterator;
import java.util.Scanner;

public class MainClass {
  public static void main(String[] args) {



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

        @SuppressWarnings("resource")

    Scanner info = new Scanner(System.in);



    System.out.println("## We are going to store data in
ArrayList ##");



//  Getting the size of ArrayList

    System.out.print("Enter the number of elements you
want  to store in ArrayList: ");

    int numberOfItems = info.nextInt();



    System.out.println();



//  Getting the elements for ArrayList

    for (int i = 0; i < numberOfItems; i++) {

      System.out.printf("Enter item %d:  ", i+1);

      String item = info.next();

      list.add(item);

    }
```

```
        System.out.println();

        //    Printing the ArrayList

        System.out.println("Here is your ArralList");

        System.out.println(list);



        System.out.println();

        System.out.println("## Now we are going to store data
    from ArrayList to Two-Dimentional Array ##");



        System.out.print("Enter the number of rows: ");

        int row = info.nextInt();

        System.out.print("Enter the number of columns: ");

        int column = info.nextInt();





    //    Declaring the ListIterator to  iterate over ArrayList

        ListIterator<String> listIterator = list.listIterator();



        //    Declaring Two-dimensional Array

        String array[][] = new String[row][column];



    //    Storing data into two-dimensional array from ArrayList

        for (int i = 0; i < row ; i++) {

          for(int j=0; j < column; j++){

            while (listIterator.hasNext()) {

            String string = (String) listIterator.next();

            array[i][j] = string;

            break;

            }

          }

        }
```

```
//    Printing the two-dimensional array

    for (int i = 0; i < row; i++) {

      System.out.println();

      for (int j = 0; j < column; j++) {

        System.out.print( "["+i+"]"+"["+j+"]" + array[i][j]
+ "   ");

      }

    }

  }
}
```

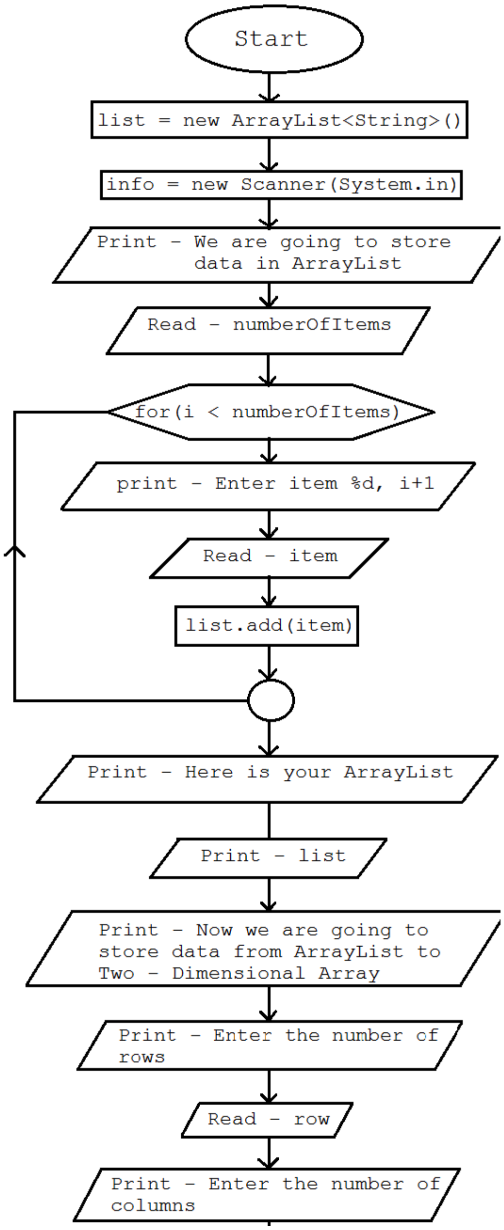
OUTPUT WOULD LOOK LIKE THIS

```
## We are going to store data in ArrayList ##
Enter the number of elements you want to store in ArrayList:
                              8

Enter item 1:  1
Enter item 2:  2
Enter item 3:  3
Enter item 4:  4
Enter item 5:  5
Enter item 6:  6
Enter item 7:  7
Enter item 8:  8

Here is your ArralList
[1, 2, 3, 4, 5, 6, 7, 8]

   ## Now we are going to store data from ArrayList to
              Two-Dimensional Array ##
Enter the number of rows: 2
Enter the number of columns: 4

[0][0]1   [0][1]2   [0][2]3   [0][3]4
[1][0]5   [1][1]6   [1][2]7   [1][3]8
```

METHODOLOGY

First the user is asked to enter the total number of elements that our ArrayList is going to store. Then a for loop is used to store the elements one by one into the ArrayList. And the ArrayList is printed to the console. Now the user is asked to provide the number of rows and number of columns to make two - dimensional Array[5][6]. A ListIterator is used to iterate over the ArrayList. To store data into two - dimensional array we have used two for loops and a while loop[7]. The for loops is used to iterate over the row and columns of the array and the while loop is used to iterate over the ArrayList elements. Inside these three loops a statement is used to store data into a string variable from the ArrayList using ListIterator object and in the next statement we are storing data from this string variable to a two - dimensional array. Then we are printing our two - dimensional array to the console using two for loops. Figure 1 represents flow of work.

FLOW OF WORK


Start
list = new ArrayList<String>()
info = new Scanner(System.in)
Print - We are going to store data in ArrayList
Read - numberOfItems
for(i < numberOfItems)
print - Enter item %d, i+1
Read - item
list.add(item)
Print - Here is your ArrayList
Print - list
Print - Now we are going to store data from ArrayList to Two - Dimensional Array
Print - Enter the number of rows
Read - row
Print - Enter the number of columns

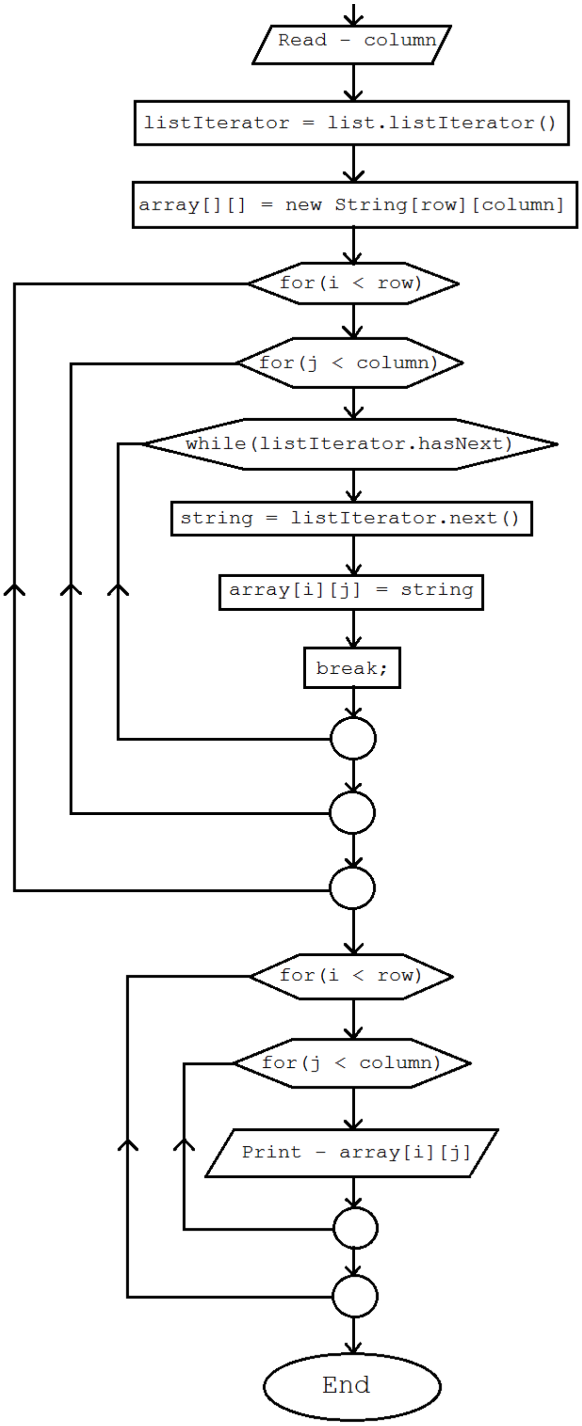

Read - column
listIterator = list.listIterator()
array[][] = new String[row][column]
for(i < row)
for(j < column)
while(listIterator.hasNext)
string = listIterator.next()
array[i][j] = string
break;
for(i < row)
for(j < column)
Print - array[i][j]
End

Figure 1: Flow Chart.

CONCLUSION

First the user is asked to enter the number of elements that they want to store in ArrayList. Here we have entered 8. After that they are asked to enter the elements one by one, here we have simply given the numbers 1, 2, 3, 4, 5, 6, 7 and 8. Then the elements in our ArrayList is displayed. Now the user is asked to provide number of rows and the number of columns to make the two – dimensional array, and it is initialized with 2 numbers of rows and 4 numbers of columns and elements from ArrayList is stored in our two – dimensional array one by one and displayed to the console.

Here we can notice that the two different type of data collections ArrayList and Array are communicating and data is shared from ArrayList to a two – dimensional Array.

REFERENCES

[1] S. Jain, R.C. Shah, S. Roy and W. Brunette, (June 2006) “Exploiting mobility for energy efficient data collection in wireless sensor Networks,"Proc. IEEE Workshop on Modeling and Optimization in Mobile Ad hoc and Wireless Networks (WiOpt’06), vol.11, issue 3, pp.327–339.

[2] R.C. Shah, S. Roy, S. Jain, and W. Brunette (May 2003), “Data MULEs: modeling a three-tier architecture for sparse sensor networks,” Proc. Int. Workshop on Sensor Network Protocols and Applications (SNPA’03), pp 30–41.

[3] S.H. Yoon, and C. Shahabi (May 2005),“Exploiting spatial correlation towards an energy efficient clustered aggregation technique (CAG)”, Proc. IEEE Int. Conf. Communications (ICC’05), pp.82–98.

[4] Xiaobing Wu , Nanjing Univ., Nanjing Guihai Chen (Aug. 2007),“Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks “,Computer Communications and Networks, ICCCN 2007, pp 1297 – 1302.

[5] Anastasi, G., Conti, M., Di Francesco, M. (July 2008),”Data collection in sensor networks with data mules: An integrated simulation analysis “Computers and Communications, ISCC 2008. , pp 1096 – 1102.

[6] Xu Xu , Sch. of Comput. Sci., Australian Nat. Univ., Canberra, ACT, Australia, Weifa Liang;
Wark, T (June 2010), “Data Quality Maximization in Sensor Networks With a Mobile Sink”, “Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 ”,pp 1-8.

[7] Seino, W., Sakashita, S., Yoshihisa, T., Hara, T. and Nishio, S.(Nov. 2010), “A Communication Protocol to Improve Fairness and Data Amount on Sensor Data Collection with a Mobile Sink,” Proc. Int. Conf. on Broadband, Wireless Computing, Communication and Applications (BWCCA’2010), pp 33–40.