

Concurrent and Parallel Programming Lab

Section – A

Assignment - 2

14-Aug-2019

1. Implement the unbounded array data structure in Java (Extend the program we developed in the class). Define following methods in UnboundedArray class.

- `public void delete();`
To delete the first element in the array.
- `public void delete(int index);`
To delete an element at a specific index. Array elements must be adjusted against the free space created after deletion
- `public void delete(int startIndex, int endIndex);`
To delete all the elements pointed from `startIndex` to `endIndex`. Array elements must be adjusted against the free space created after deletion.
- `public void modify(int index, int newElement);`
To replace an element at a specific index with `newElement`.

2. Define following threads which operate on a shared object of UnboundedArray and observe the race conditions.

- Insert_Thread: inserts three elements and sleeps for a second.
- Delete_Thread-1: deletes 2 elements and sleeps for a second.
- Delete_Thread-2: deletes all the elements from index $n/4$ to $3n/4$ and sleeps for a second.
- Modify_Thread: replaces the elements a random location every second.

Note: Each operation should handle exceptional cases like array overflow and array underflow with appropriate error messages.

All the operations must be automated (No input is given manually either using literals or keyboard input) and indices and elements are generated randomly. Please refer to the generate method written in driver class that can generate random numbers of desired range when called. You can generate random number in a specific range by calling

```
ArrayThreadDriver.generate(min,max);
```

Skeleton code for the Insert thread behavior and driver program is given below. In the similar way, define code for all other threads.

Concurrent and Parallel Programming Lab

Section – A

Assignment - 2

place this code in InsertThread.java

```
public class InsertThread extends Thread {
    private UnboundedArray array;

    public InsertThread(String name, UnboundedArray array) {
        // TODO Auto-generated constructor stub
        super(name);
        this.array = array;
    }

    public void run() {
        // Write your code to implement the job of Insert_Thread
    }
}
```

//Driver Program ArraythreadDriver.java

```
public class ArrayThreadDriver {
    /**
     * Generates a random number in the range of min to max both inclusive.
     * @param min starting value for the range
     * @param max ending value for the range
     * @return a value in the range of min to max
     */
    public static int generate(int min, int max) {
        return min + (int) (Math.random() * ((max - min) + 1));
    }

    public static void main(String[] args) {
        int initialCapacity = generate(1, 10);
        UnboundedArray array = new UnboundedArray(initialCapacity);
        InsertThread insertThread = new InsertThread("Insert_Thread",
            array);
    }
}
```

Concurrent and Parallel Programming Lab

Section – A

Assignment - 2

```
DeleteSingleThread deleteSingleThread = new DeleteSingleThread(  
    "DeleteSingle_Thread", array);
```

```
DeleteRangeThread deleteRangeThread = new DeleteRangeThread(  
    "DeleteRange_Thread", array);
```

```
ModifyThread modifyThread = new ModifyThread("Modify_Thread",  
array);
```

```
insertThread.start();
```

```
deleteSingleThread.start();
```

```
deleteRangeThread.start();
```

```
modifyThread.start();
```

```
}
```

```
}
```