DSA LabExam

Name : Lila Jadhav

PRN: 31

Q1.

a. **Implement circular queue using arrays.**

```
package CQueue;
import java.util.ArrayList;

class CircularQueue{
private int size, front, rear;
private ArrayList<Integer> queue = new ArrayList<Integer>();
CircularQueue(int size)
{
        this.size = size;
        this.front = this.rear = -1;
}
public void enQueue(int data)
{
        if((front == 0 && rear == size - 1) ||
        (rear == (front - 1) % (size - 1)))
        {
                System.out.print("Queue is Full");
        }

        else if(front == -1)
        {
                front = 0;
                rear = 0;
                queue.add(rear, data);
        }

        else if(rear == size - 1 && front != 0)
        {
                rear = 0;
                queue.set(rear, data);
        }

        else
```

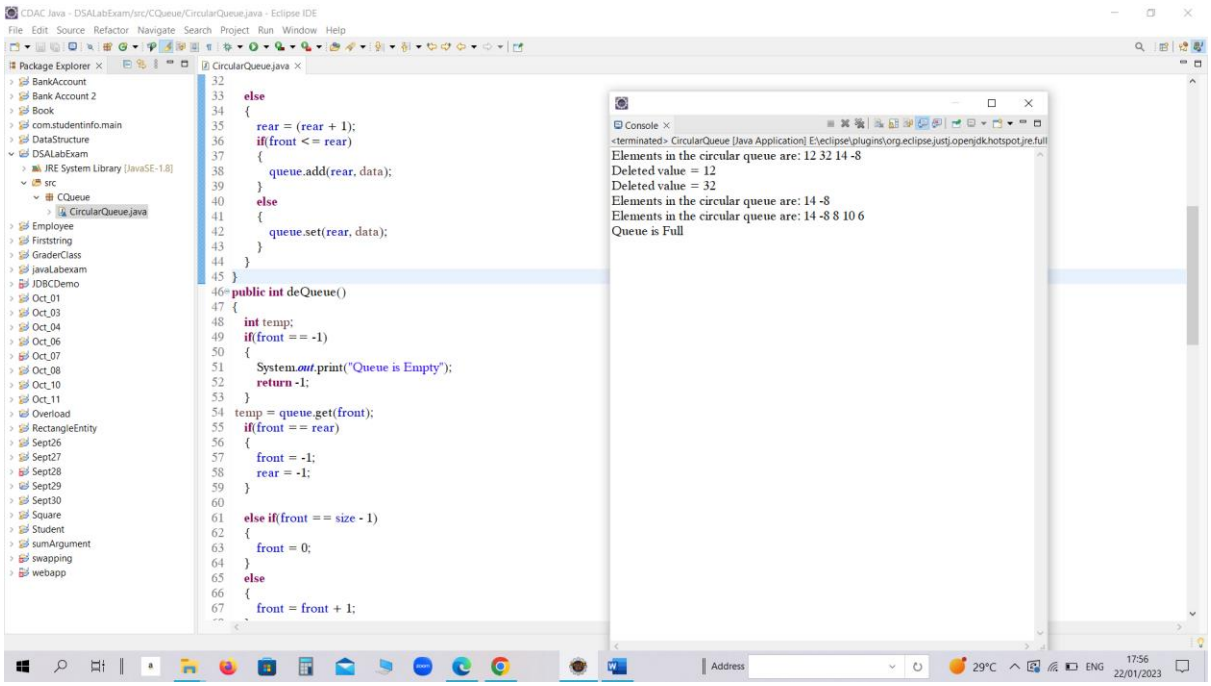
```
        {
                rear = (rear + 1);
                if(front <= rear)
                {
                        queue.add(rear, data);
                }
                else
                {
                        queue.set(rear, data);
                }
        }
}
public int deQueue()
{
        int temp;
        if(front == -1)
        {
                System.out.print("Queue is Empty");
                return -1;
        }
 temp = queue.get(front);
        if(front == rear)
        {
                front = -1;
                rear = -1;
        }

        else if(front == size - 1)
        {
                front = 0;
        }
        else
        {
                front = front + 1;
        }
        return temp;
}
public void displayQueue()
{
        if(front == -1)
        {
                System.out.print("Queue is Empty");
                return;
        }
        System.out.print("Elements in the " +
                                        "circular queue are: ");
```

```
        if(rear >= front)
        {
                for(int i = front; i <= rear; i++)
                {
                        System.out.print(queue.get(i));
                        System.out.print(" ");
                }
                System.out.println();
        }
        else
        {
                for(int i = front; i < size; i++)
                {
                        System.out.print(queue.get(i));
                        System.out.print(" ");
                }
                for(int i = 0; i <= rear; i++)
                {
                        System.out.print(queue.get(i));
                        System.out.print(" ");
                }
                System.out.println();
        }
}
public static void main(String[] args)
{
        CircularQueue q = new CircularQueue(5);

        q.enQueue(12);
        q.enQueue(32);
        q.enQueue(14);
        q.enQueue(-8);
        q.displayQueue();
   int x = q.deQueue();
        if(x != -1)
        {
                System.out.print("Deleted value = ");
                System.out.println(x);
        }
  x = q.deQueue();
  if(x != -1)
        {
                System.out.print("Deleted value = ");
                System.out.println(x);
        }
  q.displayQueue();
        q.enQueue(8);
```

```
        q.enQueue(10);
        q.enQueue(6);
        q.displayQueue();
        q.enQueue(10);
}
}
```



b. Perform quick sort to arrange given set of elements

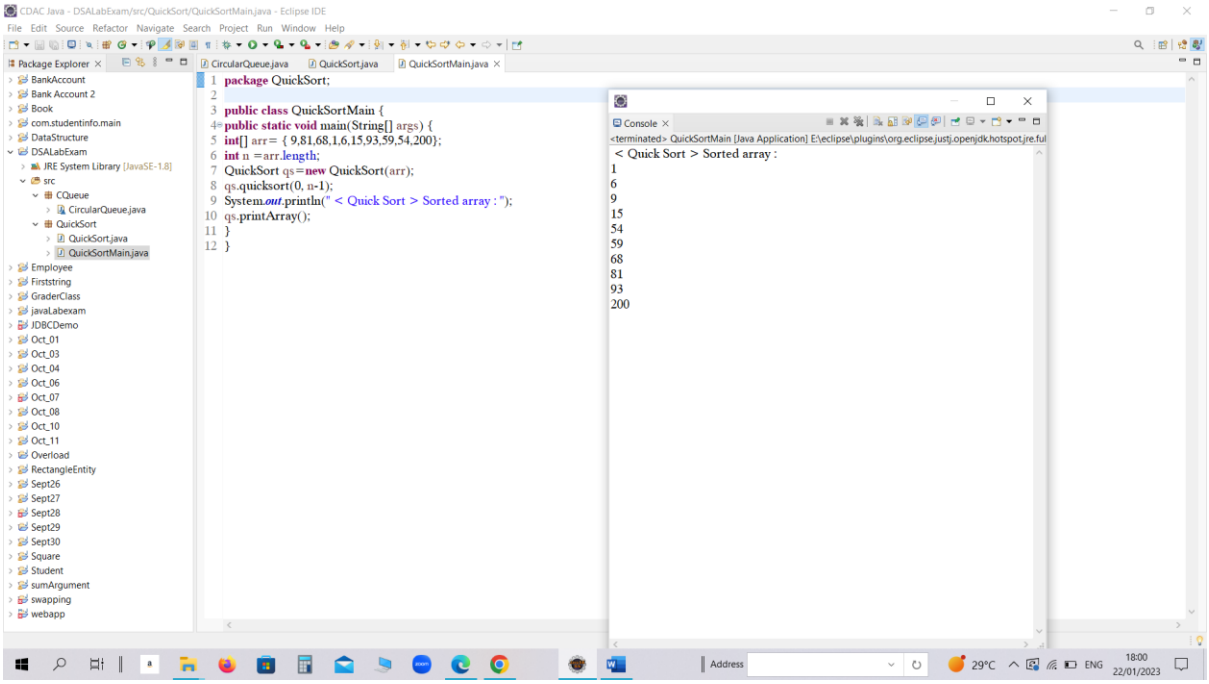
Main:

```
package QuickSort;

public class QuickSortMain {
public static void main(String[] args) {
int[] arr= { 9,81,68,1,6,15,93,59,54,200};
int n =arr.length;
QuickSort qs=new QuickSort(arr);
qs.quicksort(0, n-1);
System.out.println(" < Quick Sort > Sorted array : ");
qs.printArray();
}
}
```

```
package QuickSort;

public class QuickSort {
private int[] arr;
public QuickSort(int[] arr) {
this .arr=arr;
}
void swap(int i,int j) {
int temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
}
int partition(int start ,int end) {
int pivot=arr[end];
int i=(start-1);
for(int j =start; j <=end-1;j++) {
if(arr[j]<pivot)
{
i++;
swap(i,j);
}
}
swap(i+1,end);
return(i+1);
}
void quicksort(int start,int end)
{
if(start<end)
{
int i = partition(start,end);
quicksort(start, i-1);
quicksort(i+1, end);
}
}
void printArray()
{
int size = arr.length;
for(int i=0;i<size;i++)
System.out.println(arr[i]+" ");
System.out.println( );
}
}
```


CDAC Java - DSALabExam/src/QuickSort/QuickSortMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
BankAccount
Bank Account 2
Book
com.studentinfo.main
DataStructure
DSALabExam
JRE System Library [JavaSE-1.8]
src
CQueue
CircularQueue.java
QuickSort
QuickSort.java
QuickSortMain.java
Employee
Firststring
GraderClass
javaLabexam
JDBCDemo
Oct_01
Oct_03
Oct_04
Oct_06
Oct_07
Oct_08
Oct_10
Oct_11
Overload
RectangleEntity
Sept26
Sept27
Sept28
Sept29
Sept30
Square
Student
sumArgument
swapping
webapp
CircularQueue.java
QuickSort.java
QuickSortMain.java
1 package QuickSort;
2
3 public class QuickSortMain {
4 public static void main(String[] args) {
5 int[] arr= { 9,81,68,1,6,15,93,59,54,200};
6 int n =arr.length;
7 QuickSort qs=new QuickSort(arr);
8 qs.quicksort(0, n-1);
9 System.out.println(" < Quick Sort > Sorted array : ");
10 qs.printArray();
11 }
12 }
Console
<terminated> QuickSortMain [Java Application] E:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.ful
< Quick Sort > Sorted array :
1
6
9
15
54
59
68
81
93
200
Address
29°C
ENG
18:00
22/01/2023