Selected files

4 printable files

```
CustomerQueueSimulation.java
FirstLastList.java
Link.java
LinkQueue.java
```

CustomerQueueSimulation.java

```java
import java.util.Random;

public class CustomerQueueSimulation {
    public static void main(String[] args) {
        LinkQueue customerQueue = new LinkQueue();
        Random random = new Random();

        int totalCustomers = 10; // Number of customers in the simulation
        int serveTimeMin = 1; // Minimum serve time in seconds
        int serveTimeMax = 5; // Maximum serve time in seconds

        // Simulate customers joining the queue
        for (int i = 0; i < totalCustomers; i++) {
            customerQueue.insert(i + 1); // Insert customers with IDs 1, 2, 3, etc.
            System.out.println("Customer " + (i + 1) + " joined the queue.");
        }

        // Serve customers and observe the effect on queue size
        System.out.println("\nStarting customer service simulation:");
        while (!customerQueue.isEmpty()) {
            int serveTime = serveTimeMin + random.nextInt(serveTimeMax - serveTimeMin + 1);
            System.out.println("Serving a customer for " + serveTime + " seconds...");
            try {
                Thread.sleep(serveTime * 1000L); // Simulate service time
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }

            long servedCustomer = customerQueue.remove();
            if (servedCustomer != -1) {
                System.out.println("Customer " + servedCustomer + " has been served and 
removed from the queue.");
            } else {
                System.out.println("Service attempt, but customer not removed (N calls not 
reached).");
            }

            System.out.println("Current queue size: " + customerQueue.size());
            customerQueue.displayQueue();
        }
        System.out.println("\nAll customers have been served.");
    }
}
```

```
```

FirstLastList.java

```java
public class FirstLastList {
    private Link first;
    private Link last;
    private int size = 0; // Track the size of the list

    public FirstLastList() {
        first = null;
        last = null;
    }

    public boolean isEmpty() {
        return first == null;
    }

    public void insertLast(long dd) {
        Link newLink = new Link(dd);
        if (isEmpty()) {
            first = newLink;
        } else {
            last.next = newLink;
        }
        last = newLink;
        size++;
    }

    public long deleteFirst() {
        long temp = first.dData;
        if (first.next == null) {
            last = null;
        }
        first = first.next;
        size--;
        return temp;
    }

    public int getSize() {
        return size;
    }

    public void displayList() {
        Link current = first;
        while (current != null) {
            current.displayLink();
            current = current.next;
        }
        System.out.println();
    }
}

```

Link.java

```java
public class Link {
    public long dData;
    public Link next;

    public Link(long d) {
        dData = d;
    }

    public void displayLink() {
        System.out.print(dData + " ");
    }
}

```

LinkQueue.java

```java
public class LinkQueue {
    private FirstLastList theList;
    private int removeCallCount = 0; // Counter for remove() calls
    private final int N = 3; // Remove only after N calls

    public LinkQueue() {
        theList = new FirstLastList();
    }

    public boolean isEmpty() {
        return theList.isEmpty();
    }

    public void insert(long j) {
        theList.insertLast(j);
    }

    public long remove() {
        removeCallCount++;
        if (removeCallCount == N) { // Only remove if N calls have been made
            removeCallCount = 0; // Reset counter after removal
            return theList.deleteFirst();
        }
        return -1; // No removal
    }

    public int size() {
        return theList.getSize();
    }

    public void displayQueue() {
        System.out.print("Queue (front-->rear): ");
        theList.displayList();
    }
}

```