

Gebze Technical University  
Computer Engineering

CSE 222  
2017 Spring

HOMEWORK 4 REPORT

AYDIN ÇALIKOĞLU  
141044078

Course Assistant: Nur Banu Albayrak

## 1. System Requirements

Q1 :

A Csv file

Q2 :

A Csv file

## 2. Problem Solutions Approach

Q1:

Stack A:

Stack interface imlement, ArrayList'i extend ederek bir yapı oluşturduk.

Bu durumda StackInterface fonksiyonları kullanılarak **StackA is a ArrayList** ilişkisi oluştu.

Stack B:

Stack interface imlement ettik ve ArrayList ile has a ilişkisi oluşturduk.

Bu durumda StackInterface fonksiyonları kullanılarak **StackA has a ArrayList** ilişkisi ve bu ilişki **Composition** olarak kullanıldı. Bu method Hem atama hem de Atama olarak kullanıldığından güçlü bir method.

Stack C:

Bir Node Class tanımladık ve bu class kendi içinde **Aggregation** yapısı olarak kullandık.

StackC ile Own (**Composition** ) has Node ilişkisini sağlayarak kullandık.

Stack D:

Bir Queue yapısı tanımladık ve LinkedList kullanarak Declare ettik.

StackD ile Own (**Composition** ) has Queue ilişkisini sağlayarak kullandık.

Q2:

1. reverse method için, ListIterator ile add ve remove fonksiyonları kullandık listenin sonundan başına doğru eleman silip tekrar ekledik, böylece que yapısının tersini aldık.
2. reverseQueue method u bir recursive yapı oluşturduk, bu yapıda listeden eleman çıkartıp, elemanı sakladık ve yapıyı 1 eksik queue ile tekrar çağırdık bu queue eleman sayısı 0 olduktan sonra tekrar sakladığımız elemanları ekledik.

Q3:

PriorityQueue için sıralı bir ekleme işlemini yaptık. Highest Priorty elaman ilk sırada eleman olduğundan ekleyip çıkardığımız tüm elemanlar sıralanıyor.

### 3. Test Cases for main

Q1:

Tüm Stack'leri tanımladıktan sonra file dan okuduğumuz her satır içindeki verileri StackA,StackB,StackC,StackD içersine ekledik ve Bunları StackInterface tutan bir ArrayList İçerisinde tuttuk.  
Dosya'ya yazdırmadan önce Tüm Stack verileri pop ile çıkardık ve verileri tersine çevirdik.

Q2:

`listQueueForReverseQueue` and `listQueueForReverse` iki adet iç içe Queue yapısı oluşturduk ve dosyadan okuduğumuz verileri içerdeki Queue ya Scanner yardımı ile okuduğumuz verileri add methodu ile Ekledik.

```
myQueue<myQueue> listQueueForReverseQueue = new myQueue<myQueue>();
```

```
myQueue<myQueue> listQueueForReverse =new myQueue<myQueue>();
```

`Reverse` method için önce tüm satırların ters çevirdik. Ardından tüm içerikleri bir döngü yardımıyla ters çevirdik.

```
/* reverse method TEST */
```

```
listQueueForReverse.reverse();
```

```
int sizeQueueList = listQueueForReverse.size();
```

```
for (int i = 0; i < sizeQueueList; i++) {
```

```
listQueueForReverse.get(i).reverse();
```

```
}
```

`reverseQueue` methodu ile `listQueueForReverseQueue` yapısı reverse ettik.

```
/* 2=) reverseQueue TEST */
```

```
listQueueForReverseQueue.reverseQueue(listQueueForReverseQueue);
```

```
sizeQueueList=listQueueForReverseQueue.size();
```

```
for (int i = 0; i < sizeQueueList; i++) {
```

```
listQueueForReverseQueue.get(i).reverseQueue(listQueueForReverseQueue.get(i));
```

```
}
```

Ek olarak metodlarımız tekrar tersini alıyor mu, kontrol ettik

```
/* @TEST reverse Method Tekrar Tersini alıyor mu */  
  
listQueueForReverse.reverse();  
  
listQueueForReverse.reverse();  
  
/* @TEST reverseQueue Tekrar Tersini alıyor mu */  
  
listQueueForReverseQueue.reverseQueue(listQueueForReverseQueue);  
  
listQueueForReverseQueue.reverseQueue(listQueueForReverseQueue);
```

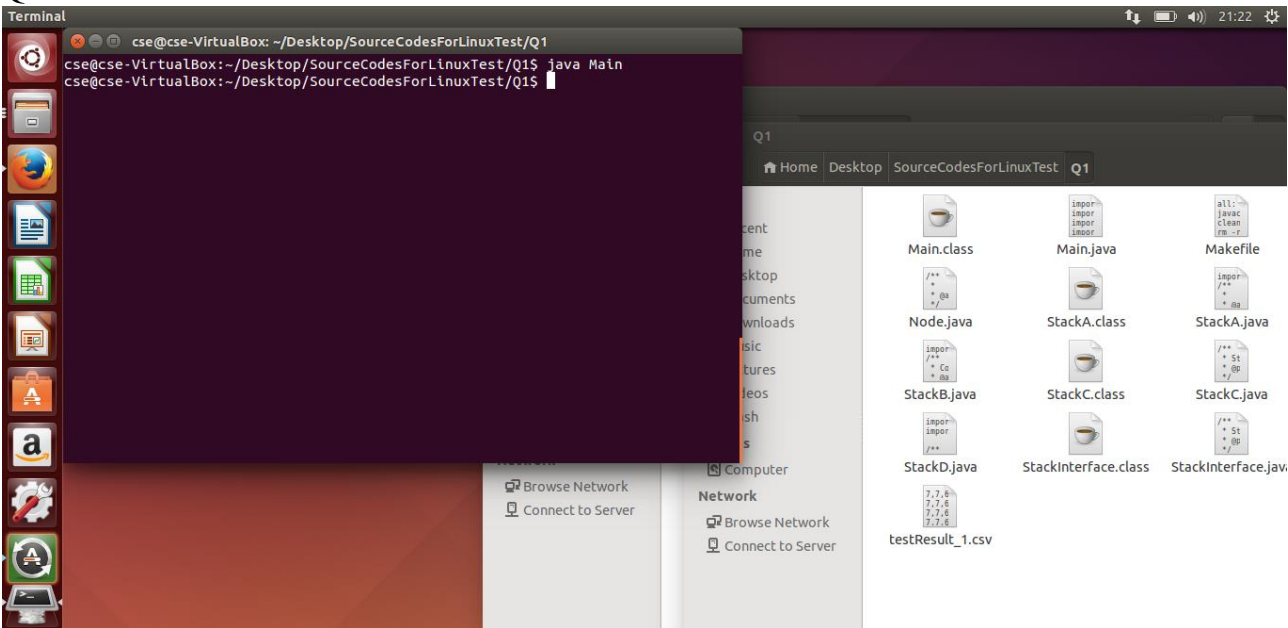
Q3:

Okuduğumuz tüm elemanları priorityQueue içine aktardıktan sonra sırayla çıkardığımız elemanları Göüntüledik kullandığımız test Case özetimiz

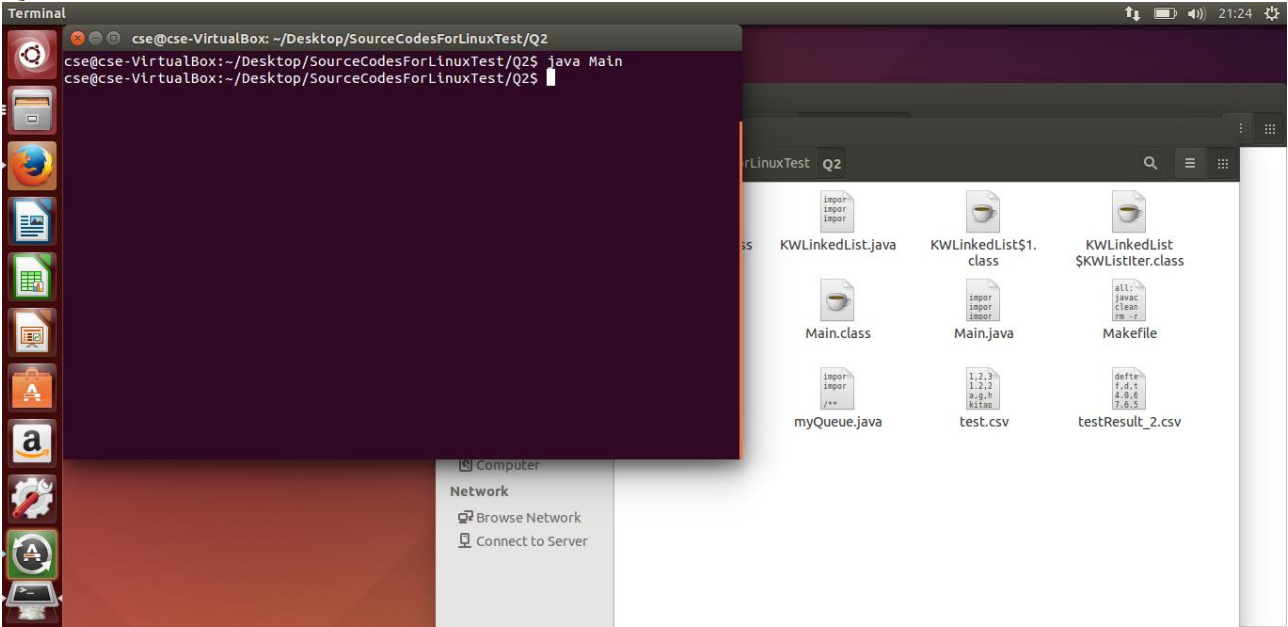
```
for(int j = 0; j < priorityQueueTest.size(); j += 2) {  
    int size = ((priorityQueueInterface)priorityQueueTest.get(j)).size();  
  
    int i;  
    for(i = 0; i < size; ++i) {  
        if(i != 0) {  
            output = output + ",";  
        }  
  
        output = output +  
((priorityQueueInterface)priorityQueueTest.get(j)).deleteMin();  
    }  
  
    output = output + "\n";  
  
    for(i = 0; i < size; ++i) {  
        if(i != 0) {  
            output = output + ",";  
        }  
  
        output = output + ((priorityQueueInterface)priorityQueueTest.get(j +  
1)).deleteMin();  
    }  
  
    output = output + "\n";  
}
```

## 4. Running and Results

Q1:



Q2:



Q3

