

Gebze Technical University
Computer Engineering
CSE 222

2017 Spring

HOMEWORK 4 REPORT

BURAK DEMİRCİ

141044091

Course Assistant: Nur Banu Albayrak

1. System Requirements

Bu ödev için gereksinimlerimizi şu şekilde sıralayabiliriz;

1.Soru için,

- StackInterface yazılması gerekiyor. Bunun için belirlenmiş metodlar incelenip oluşturuluyor
- StackA classı için StackInterfacesi implement ediliyor ve Arraylist classını ise Extend ediyor
- StackB classı için StackInterfacesi implement ediliyor ve Arraylist objesi kullanılıyor
- StackC classı için StackInterfacesi implement ediliyor ve Node yapısı inner class olarak kullanılıyor
- StackD classı için StackInterfacesi implement ediliyor ve Queue interfacesinden imlement edilmiş olan LinkedList objesi kullanılıyor.
- Bütün bu classlar için toString metodu yazıldı

2. Soru için,

- KWLinkedList classı oluşturuldu bu class AbstractSequentialList extend edip Queue yide imlement ediyor . Bu yöntemlerle alakalı birçok metod oluşturuldu
- myQueue classı için KWLinkedList clası extend edildi ve Queue classı imlement edildi . Bu classta reseerve metodları oluşturuldu

2. Problem Solutions Approach

1.Soru için,

StackA için,

- Arraylistin belli fonksiyonları kullanılacak amaç doğrultusunda override edildi
- StackInterfacesinde bulunan yöntemler pdfte açıklandığı doğrultuda oluşturuldu ve test edildi

StackB için,

- Arraylist objesi kullanılarak verilen datalar saklandı ve StackInterfasinde bulunan yöntemler bu doğrultuda yazıldı
- Bütün yöntemler oluşturulup test edildi

StackC için,

- Node inner classı verilenler doğrultusunda imlement edildi ve bu class StackC de implement edilen StackInterfacesinin yöntemlerinin içerisinde kullanıldı
- Bütün yöntemler oluşturulup test edildi

StackD için,

- StackInterface sindeki yöntemler implement edildi
- Bu yöntemleri imlement ederken Queue den türemiş olan LinkedList objesi kullanıldı
- Bütün yöntemler oluşturulup test edildi

2.Soru için,

KWLinkedList classı ders kitabından yararlanılarak oluşturuldu class AbstractSequentialList extend edip Queue yide imlement edildi . Bu yöntemler test edildi . KWLinkedList objesi kullanılarak myQueue classında işlemler yapıldı . Bu metodlar imlement edilirken bazı metodlar kullanım dogrultusunda @Override edildi bunlardan bazıları poll(),add()

4. Test Cases

1.Soru için,

StackTest Classı imlement edildi bu classta veriebu yapılar içerisinde tutuldu

```
StackA<String> stackAt = new StackA<>();
```

```
StackB<String> stackBt = new StackB<>();
```

```
StackC<String> stackCt = new StackC<>();
```

```
StackD<String> stackDt = new StackD<>();
```

İlk önce satır satır okunup sonra veriler virgüllere göre tek tek eklendi eklenen veriler toString() metodları kullanılarak result_1.csv dosyasına yazıldı. Performansları ölçüldü.

2.Soru için,

myQueueTest Classı imlement edildi ve test için şu objeler kullanıldı

```
myQueue<String> myqueue = new myQueue<>();
```

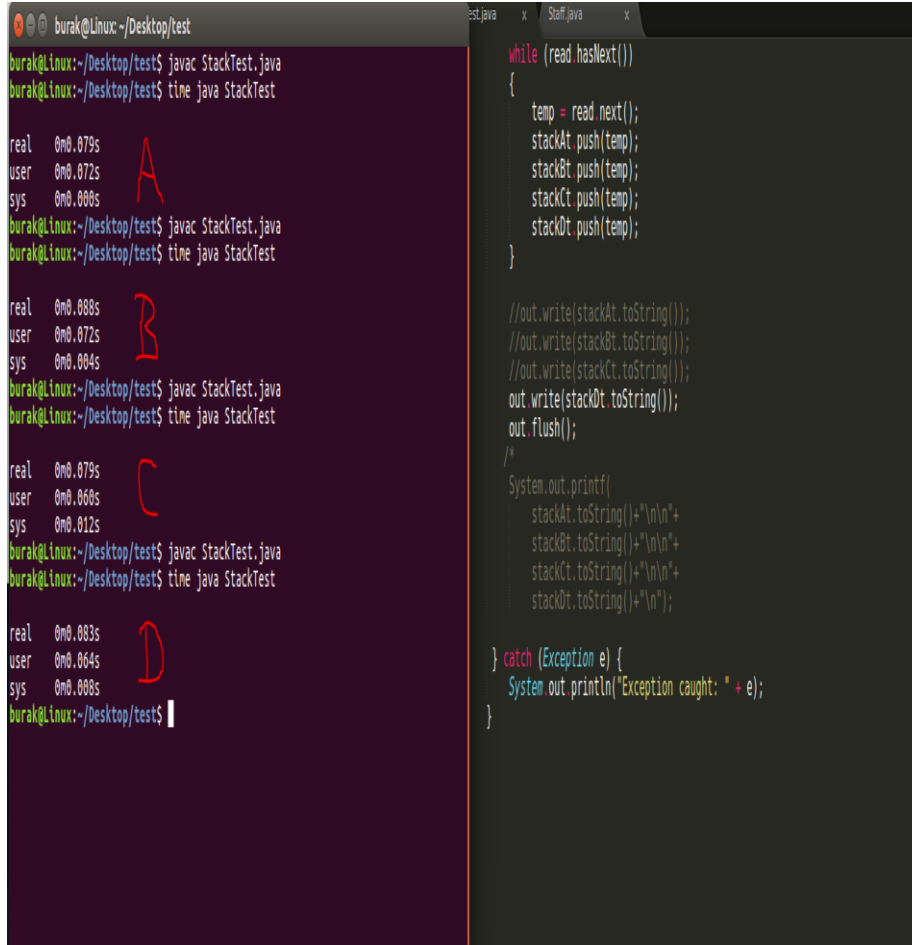
```
Queue<String> recQueue = new myQueue<>();
```

İlk obje reserve() metodunu kullanmak için oluşturulurken 2. Obje reserve(Queue Val) şeklinde recursive olarak tanımlanan metodu kullanmak için oluşturuldu .

İlk önce satır satır okunup sonra veriler virgüllere göre tek tek eklendi eklenen veriler toString() metodları kullanılarak result_2.csv dosyasına yazıldı.

5. Running and Results

1.Soru İçin:



```
burak@Linux: ~/Desktop/test
burak@Linux:~/Desktop/test$ javac StackTest.java
burak@Linux:~/Desktop/test$ time java StackTest

real    0m0.079s
user    0m0.072s
sys     0m0.008s
A
burak@Linux:~/Desktop/test$ javac StackTest.java
burak@Linux:~/Desktop/test$ time java StackTest

real    0m0.088s
user    0m0.072s
sys     0m0.004s
B
burak@Linux:~/Desktop/test$ javac StackTest.java
burak@Linux:~/Desktop/test$ time java StackTest

real    0m0.079s
user    0m0.060s
sys     0m0.012s
C
burak@Linux:~/Desktop/test$ javac StackTest.java
burak@Linux:~/Desktop/test$ time java StackTest

real    0m0.083s
user    0m0.064s
sys     0m0.008s
D
burak@Linux:~/Desktop/test$
```

```
while (read.hasNext())
{
    temp = read.next();
    stackAt.push(temp);
    stackBt.push(temp);
    stackCt.push(temp);
    stackDt.push(temp);
}

//out.write(stackAt.toString());
//out.write(stackBt.toString());
//out.write(stackCt.toString());
out.write(stackDt.toString());
out.flush();
/*
System.out.printf(
    stackAt.toString()+"\n\n"+
    stackBt.toString()+"\n\n"+
    stackCt.toString()+"\n\n"+
    stackDt.toString()+"\n\n");
} catch (Exception e) {
    System.out.println("Exception caught: " + e);
}
```

Stack A = 0,79
Stack B = 0,84
Stack C = 0,79
Stack D = 0,83

Bütün metotlar ortalama $O(n)$ zamanda çalışır
bu yüzden dolayı çalışma süreleri birbirine yakındır

2.Soru İçin:

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HW_4 src myQueueTest
StackInterface.java x StackTest.java x StackB.java x StackC.java x StackD.java x StackA.java x KWLinkedList.java x myQueue.java x myQueueTest.java x
myQueueTest main()
43 }
44 else
45 {
46     myQueue.add(Temp2);
47     recQueue.add(Temp2);
48     Temp2="";
49 }
50 }
51 myQueue.add(Temp2);
52 recQueue.add(Temp2);
53 Temp2="";
54 }
55
56 System.out.printf("\nFIRST READING VALUE\n");
57 System.out.println(myQueue.toString());
58
59 System.out.printf("\nREVERSE VALUE USING LOOP\n");
60 myQueue.reverse();
61 System.out.println(myQueue.toString()+"\n");
62
63 System.out.printf("\nREVERSE VALUE USING RECURSIVE CALL \n");
64 myQueue.reverse(recQueue);
65 System.out.println(recQueue.toString());
66
67 }catch (Exception e) {
68     System.out.println("Exception caught: " + e);
69 }
70 }
71
72 }
73
74 }
75
76 }
77
78 }
```

Run myQueueTest

```
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
```

FIRST READING VALUE
1, 2, 3, 4, 5, 6, 7, 1.2, 2.3, 4.5, 6.7, 4.0, a, g, h, j, t, d, f, kitap, kalem, defter

REVERSE VALUE USING LOOP
defter, kalem, kitap, f, d, t, j, h, g, a, 4.0, 6.7, 4.5, 2.3, 1.2, 7, 6, 5, 4, 3, 2, 1

REVERSE VALUE USING RECURSIVE CALL
defter, kalem, kitap, f, d, t, j, h, g, a, 4.0, 6.7, 4.5, 2.3, 1.2, 7, 6, 5, 4, 3, 2, 1

Process finished with exit code 0