

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 04 REPORT

MUSTAFA BİNGÜL 141044077

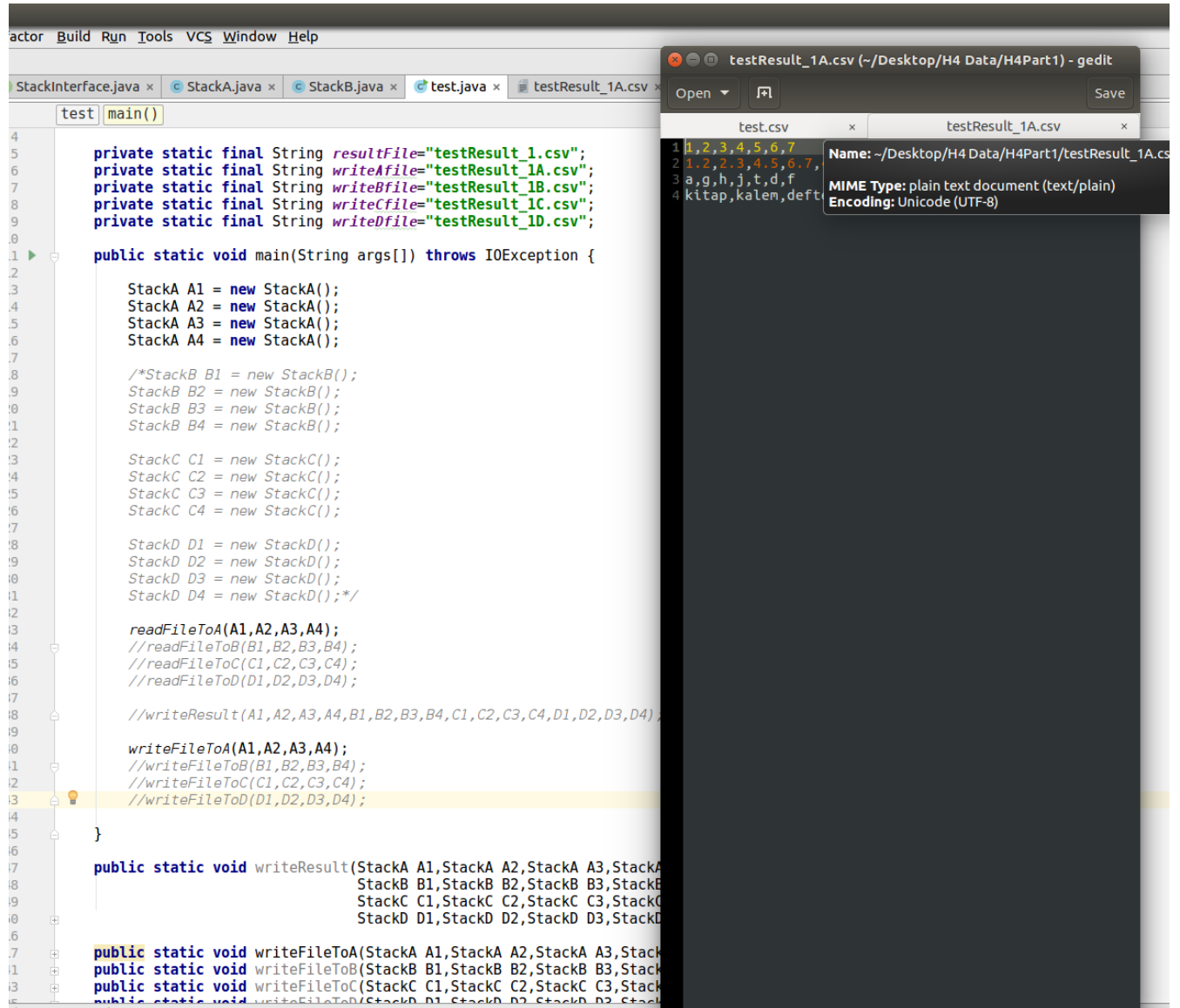
Course Assistant:

Nur Banu ALBAYRAK

Q1-/

1.Test Cases

StackA

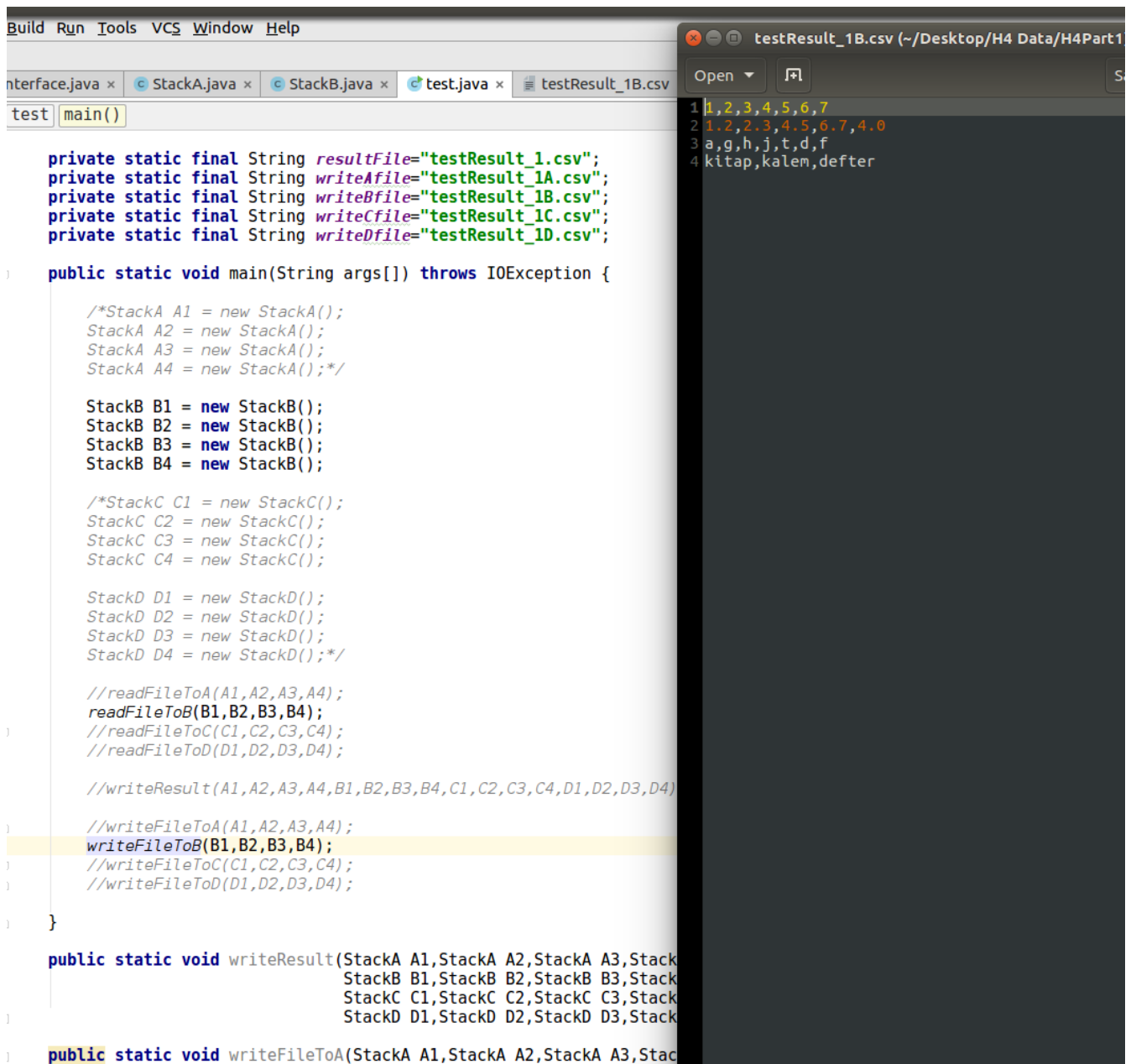


The screenshot shows an IDE with a Java file named `test.java` and a CSV file named `testResult_1A.csv`. The Java code defines a `main` method that creates four `StackA` objects (A1, A2, A3, A4) and four `StackB` objects (B1, B2, B3, B4). It then reads data from `test.csv` and writes it to `testResult_1A.csv`. The CSV file contains the following data:

1	2	3	4	5	6	7
1	2	3	4	5	6	7
1	2	2	3	4	5	6
a	g	h	j	t	d	f
4	kitap	kalem	deft			

4 Adet StackA tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackB



The screenshot shows an IDE with two windows. The left window displays the `test.java` file, which contains a `main` method and several utility methods. The right window displays the `testResult_1B.csv` file, which contains four lines of data.

```
Build Run Tools VCS Window Help
nterface.java x StackA.java x StackB.java x test.java x testResult_1B.csv
test main()

private static final String resultFile="testResult_1.csv";
private static final String writeAfile="testResult_1A.csv";
private static final String writeBfile="testResult_1B.csv";
private static final String writeCfile="testResult_1C.csv";
private static final String writeDfile="testResult_1D.csv";

public static void main(String args[]) throws IOException {

    /*StackA A1 = new StackA();
    StackA A2 = new StackA();
    StackA A3 = new StackA();
    StackA A4 = new StackA();*/

    StackB B1 = new StackB();
    StackB B2 = new StackB();
    StackB B3 = new StackB();
    StackB B4 = new StackB();

    /*StackC C1 = new StackC();
    StackC C2 = new StackC();
    StackC C3 = new StackC();
    StackC C4 = new StackC();

    StackD D1 = new StackD();
    StackD D2 = new StackD();
    StackD D3 = new StackD();
    StackD D4 = new StackD();*/

    //readFileToA(A1,A2,A3,A4);
    readFileToB(B1,B2,B3,B4);
    //readFileToC(C1,C2,C3,C4);
    //readFileToD(D1,D2,D3,D4);

    //writeResult(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4,D1,D2,D3,D4)

    //writeFileToA(A1,A2,A3,A4);
    writeFileToB(B1,B2,B3,B4);
    //writeFileToC(C1,C2,C3,C4);
    //writeFileToD(D1,D2,D3,D4);
}

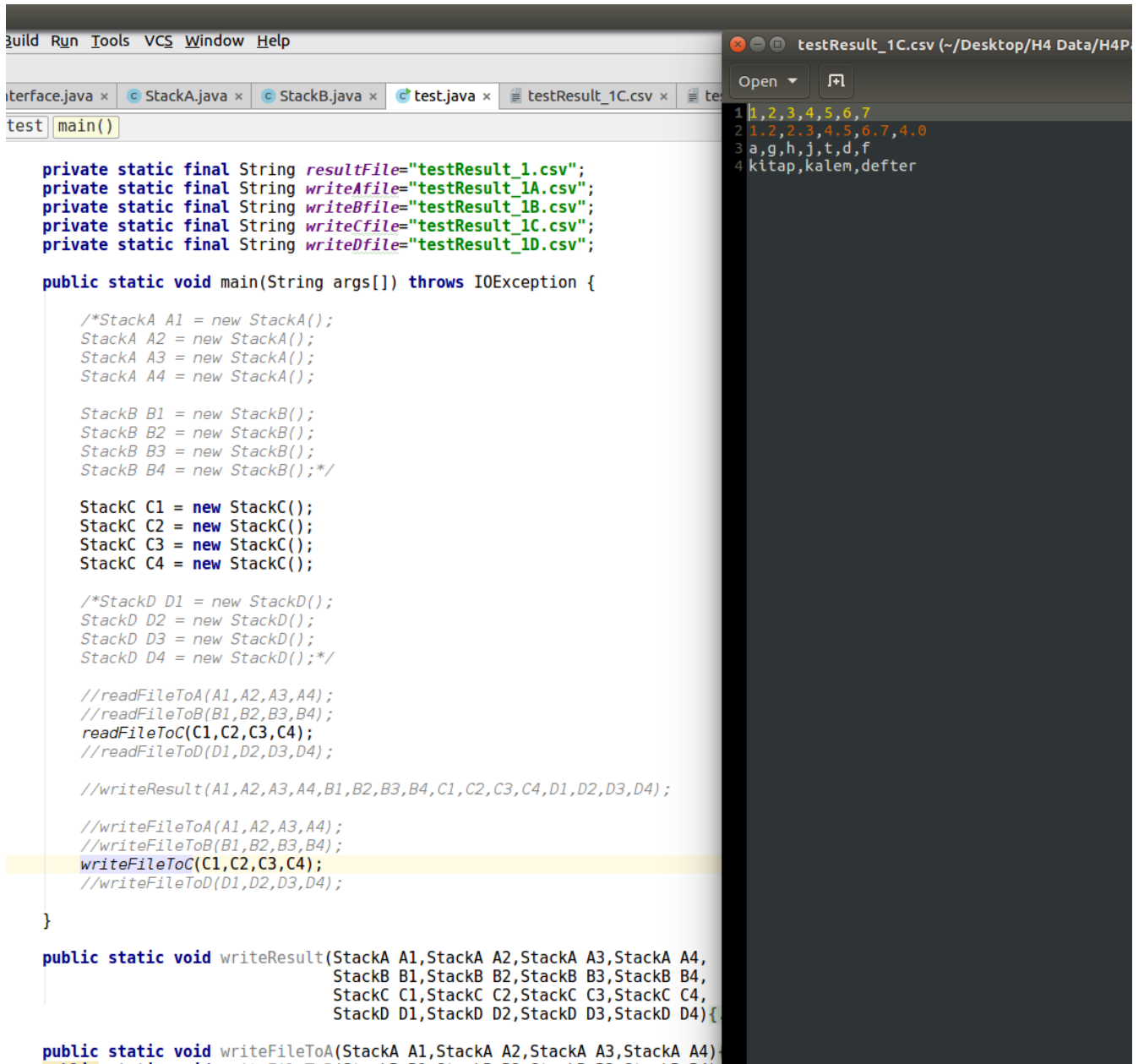
public static void writeResult(StackA A1,StackA A2,StackA A3,Stack
StackB B1,StackB B2,StackB B3,Stack
StackC C1,StackC C2,StackC C3,Stack
StackD D1,StackD D2,StackD D3,Stack

public static void writeFileToA(StackA A1,StackA A2,StackA A3,Stack
```

```
testResult_1B.csv (~/Desktop/H4 Data/H4Part1)
Open [icon] S
1 1,2,3,4,5,6,7
2 1.2,2.3,4.5,6.7,4.0
3 a,g,h,j,t,d,f
4 kitap,kalem,defter
```

4 Adet StackB tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackC



The screenshot shows an IDE with several open files. The active file is `test.java`, which contains the following Java code:

```
private static final String resultFile="testResult_1.csv";
private static final String writeAfile="testResult_1A.csv";
private static final String writeBfile="testResult_1B.csv";
private static final String writeCfile="testResult_1C.csv";
private static final String writeDfile="testResult_1D.csv";

public static void main(String args[]) throws IOException {

    /*StackA A1 = new StackA();
    StackA A2 = new StackA();
    StackA A3 = new StackA();
    StackA A4 = new StackA();

    StackB B1 = new StackB();
    StackB B2 = new StackB();
    StackB B3 = new StackB();
    StackB B4 = new StackB();*/

    StackC C1 = new StackC();
    StackC C2 = new StackC();
    StackC C3 = new StackC();
    StackC C4 = new StackC();

    /*StackD D1 = new StackD();
    StackD D2 = new StackD();
    StackD D3 = new StackD();
    StackD D4 = new StackD();*/

    //readFileToA(A1,A2,A3,A4);
    //readFileToB(B1,B2,B3,B4);
    readFileToC(C1,C2,C3,C4);
    //readFileToD(D1,D2,D3,D4);

    //writeResult(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4,D1,D2,D3,D4);

    //writeFileToA(A1,A2,A3,A4);
    //writeFileToB(B1,B2,B3,B4);
    writeFileToC(C1,C2,C3,C4);
    //writeFileToD(D1,D2,D3,D4);
}

public static void writeResult(StackA A1,StackA A2,StackA A3,StackA A4,
                               StackB B1,StackB B2,StackB B3,StackB B4,
                               StackC C1,StackC C2,StackC C3,StackC C4,
                               StackD D1,StackD D2,StackD D3,StackD D4){

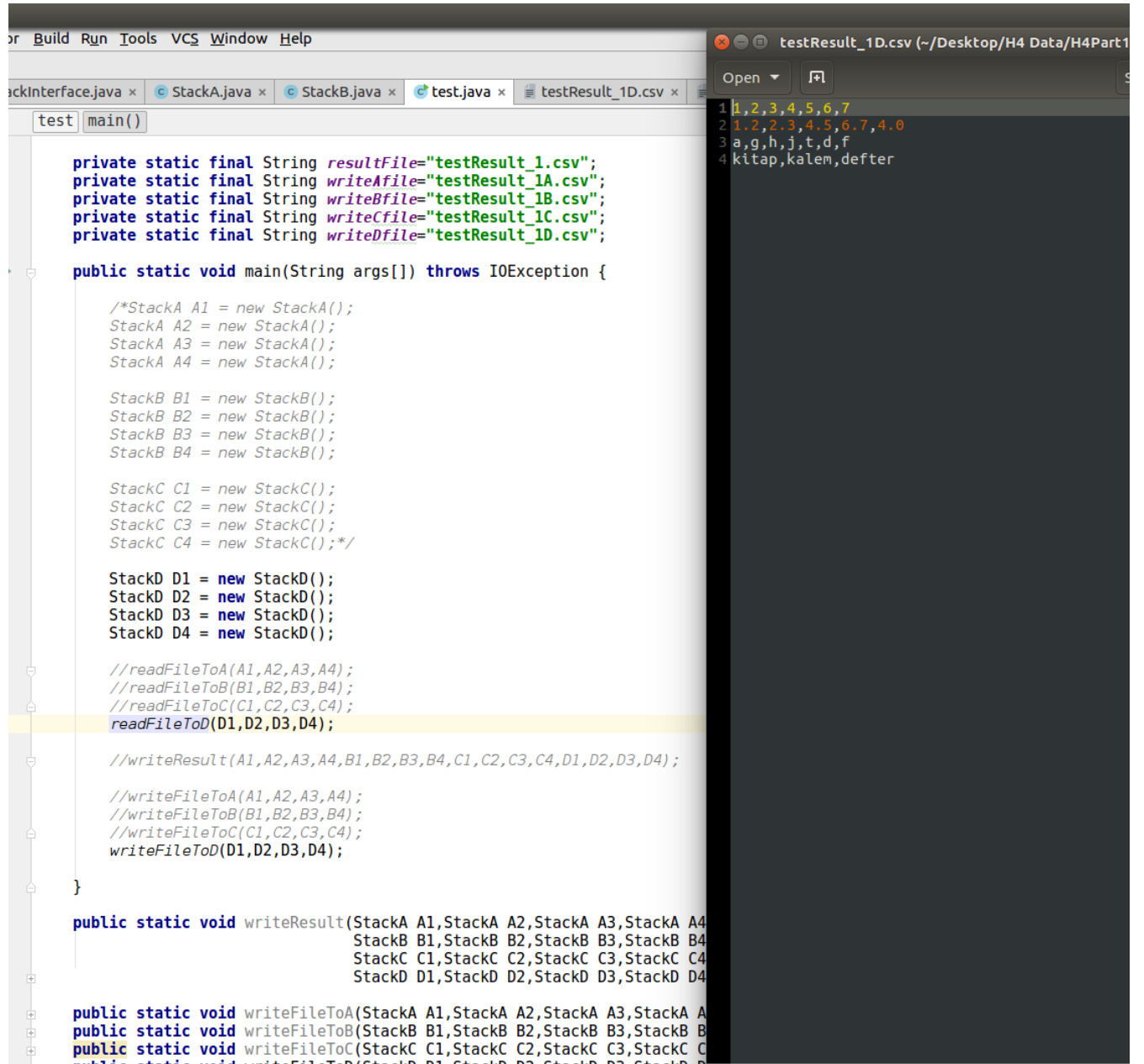
public static void writeFileToA(StackA A1,StackA A2,StackA A3,StackA A4){
```

The `testResult_1C.csv` file is also open, showing the following content:

```
1 1,2,3,4,5,6,7
2 1.2,2.3,4.5,6.7,4,0
3 a,g,h,j,t,d,f
4 kitap,kalem,defter
```

4 Adet StackC tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackD



The screenshot shows an IDE with two windows. The left window is 'test.java' with the following code:

```
private static final String resultFile="testResult_1.csv";
private static final String writeAfile="testResult_1A.csv";
private static final String writeBfile="testResult_1B.csv";
private static final String writeCfile="testResult_1C.csv";
private static final String writeDfile="testResult_1D.csv";

public static void main(String args[]) throws IOException {

    /*StackA A1 = new StackA();
    StackA A2 = new StackA();
    StackA A3 = new StackA();
    StackA A4 = new StackA();

    StackB B1 = new StackB();
    StackB B2 = new StackB();
    StackB B3 = new StackB();
    StackB B4 = new StackB();

    StackC C1 = new StackC();
    StackC C2 = new StackC();
    StackC C3 = new StackC();
    StackC C4 = new StackC();*/

    StackD D1 = new StackD();
    StackD D2 = new StackD();
    StackD D3 = new StackD();
    StackD D4 = new StackD();

    //readFileToA(A1,A2,A3,A4);
    //readFileToB(B1,B2,B3,B4);
    //readFileToC(C1,C2,C3,C4);
    readFileToD(D1,D2,D3,D4);

    //writeResult(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4,D1,D2,D3,D4);

    //writeFileToA(A1,A2,A3,A4);
    //writeFileToB(B1,B2,B3,B4);
    //writeFileToC(C1,C2,C3,C4);
    writeFileToD(D1,D2,D3,D4);
}

public static void writeResult(StackA A1,StackA A2,StackA A3,StackA A4,
                               StackB B1,StackB B2,StackB B3,StackB B4,
                               StackC C1,StackC C2,StackC C3,StackC C4,
                               StackD D1,StackD D2,StackD D3,StackD D4)

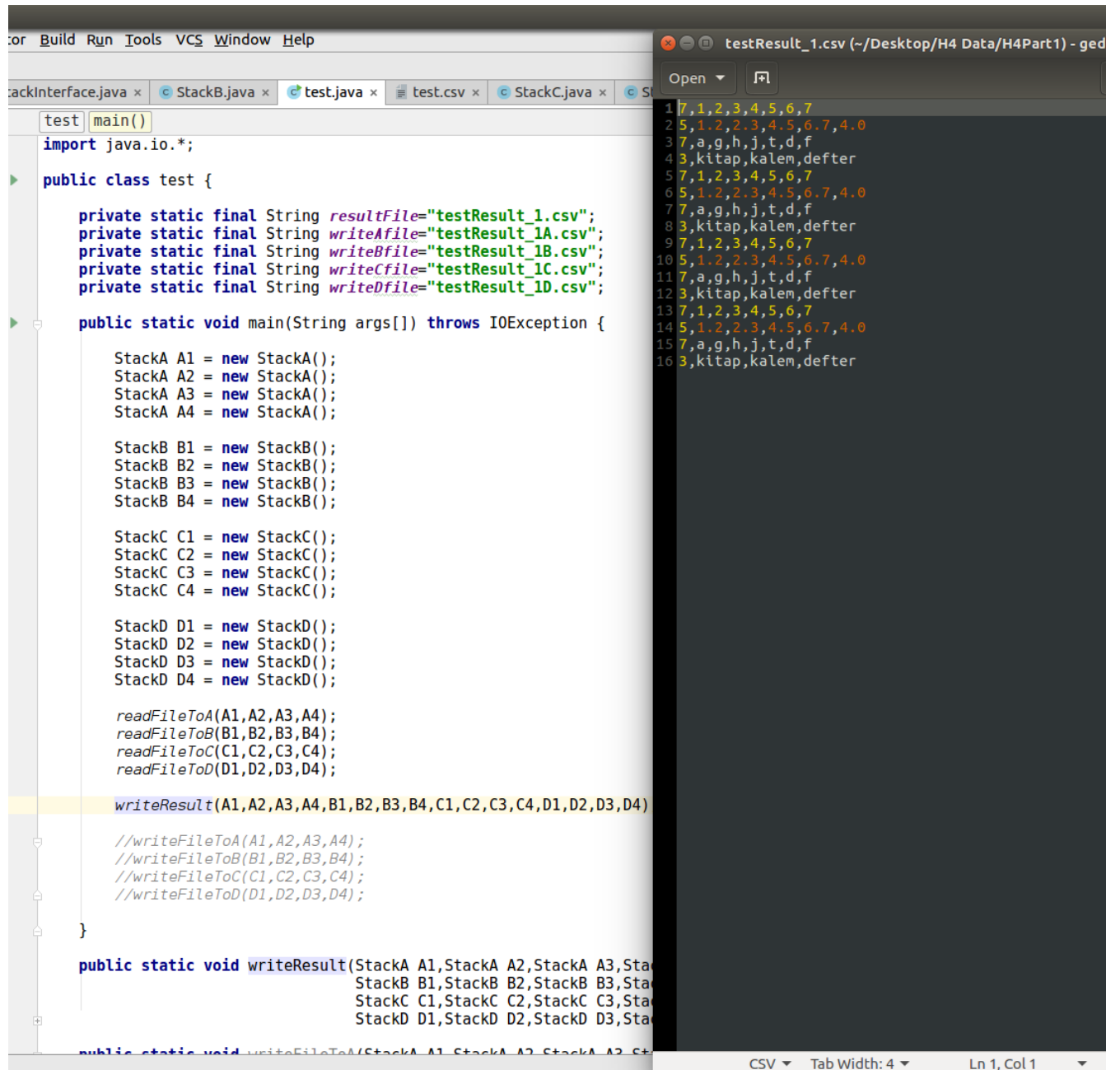
public static void writeFileToA(StackA A1,StackA A2,StackA A3,StackA A4)
public static void writeFileToB(StackB B1,StackB B2,StackB B3,StackB B4)
public static void writeFileToC(StackC C1,StackC C2,StackC C3,StackC C4)
public static void writeFileToD(StackD D1,StackD D2,StackD D3,StackD D4)
```

The right window is 'testResult_1D.csv' with the following content:

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

4 Adet StackD tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

Tüm stacklerin tek dosyaya başlarına da size lerini yazarak yazdırdığım durum.



The screenshot shows an IDE with a Java file named `test.java` and a CSV file named `testResult_1.csv`. The Java code defines a `test` class with a `main` method that creates four stacks (A, B, C, D) and writes their contents to a CSV file. The CSV file contains the output of the `main` method, showing the contents of each stack.

```
import java.io.*;

public class test {

    private static final String resultFile="testResult_1.csv";
    private static final String writeAfile="testResult_1A.csv";
    private static final String writeBfile="testResult_1B.csv";
    private static final String writeCfile="testResult_1C.csv";
    private static final String writeDfile="testResult_1D.csv";

    public static void main(String args[]) throws IOException {

        StackA A1 = new StackA();
        StackA A2 = new StackA();
        StackA A3 = new StackA();
        StackA A4 = new StackA();

        StackB B1 = new StackB();
        StackB B2 = new StackB();
        StackB B3 = new StackB();
        StackB B4 = new StackB();

        StackC C1 = new StackC();
        StackC C2 = new StackC();
        StackC C3 = new StackC();
        StackC C4 = new StackC();

        StackD D1 = new StackD();
        StackD D2 = new StackD();
        StackD D3 = new StackD();
        StackD D4 = new StackD();

        readFileToA(A1,A2,A3,A4);
        readFileToB(B1,B2,B3,B4);
        readFileToC(C1,C2,C3,C4);
        readFileToD(D1,D2,D3,D4);

        writeResult(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4,D1,D2,D3,D4)

        //writeFileToA(A1,A2,A3,A4);
        //writeFileToB(B1,B2,B3,B4);
        //writeFileToC(C1,C2,C3,C4);
        //writeFileToD(D1,D2,D3,D4);

    }

    public static void writeResult(StackA A1,StackA A2,StackA A3,StackA A4,StackB B1,StackB B2,StackB B3,StackB B4,StackC C1,StackC C2,StackC C3,StackC C4,StackD D1,StackD D2,StackD D3,StackD D4) {

        public static void writeFileToA(StackA A1,StackA A2,StackA A3,StackA A4) {
```

```
1 7,1,2,3,4,5,6,7
2 5,1.2,2.3,4.5,6.7,4.0
3 7,a,g,h,j,t,d,f
4 3,kitap,kalem,defter
5 7,1,2,3,4,5,6,7
6 5,1.2,2.3,4.5,6.7,4.0
7 7,a,g,h,j,t,d,f
8 3,kitap,kalem,defter
9 7,1,2,3,4,5,6,7
10 5,1.2,2.3,4.5,6.7,4.0
11 7,a,g,h,j,t,d,f
12 3,kitap,kalem,defter
13 7,1,2,3,4,5,6,7
14 5,1.2,2.3,4.5,6.7,4.0
15 7,a,g,h,j,t,d,f
16 3,kitap,kalem,defter
```

CSV Tab Width: 4 Ln 1, Col 1

2. Running and Results

StackA class ını ArrayList sınıfından extend edip StackInterface interfacesinden implement edip yazdım. Gerekli olarak gördüğüm toString methodunu ve zorunlu olan push,pop,size ve isEmpty methodlarını implement ettim. StackA objesini yapıp push methodu ile veri ekleyip, pop methodu ile de veri yapısının sonundan veri silebiliyoruz aynı zamanda pop methodunda eğer stackimiz boş ise exception fırlatıyoruz. Size methodu da elimizdeki objenin boyutunu vermekte, isEmpty methodu ise stack imizin boş olup olmadığını return etmekte.

StackB class ında ArrayList i composition olarak kullanıp aynı zamanda StackInterface interfacesini implement ediyorum. Gerekli olarak gördüğüm toString , get methodlarını ve zorunlu olan push,pop,size ve isEmpty methodlarını implement ettim. StackB objesini yapıp push methodu ile veri ekleyip, pop methodu ile de veri yapısının sonundan veri silebiliyoruz aynı zamanda pop methodunda eğer stackimiz boş ise exception fırlatıyoruz. Size methodu da elimizdeki objenin boyutunu vermekte, isEmpty methodu ise stack imizin boş olup olmadığını return etmekte.

StackC class ında Inner class olarak Node tanımlayıp aynı zamanda StackInterface interfacesini implement ettim. Node inner class ı için kullandığım methodları da implement ettim. Gerekli olarak gördüğüm toString methodunu ve zorunlu olan push,pop,size ve isEmpty methodlarını da implement ettim. StackC objesini yapıp push methodu ile veri ekleyip, pop methodu ile de veri yapısının sonundan veri silebiliyoruz aynı zamanda pop methodunda eğer stackimiz boş ise exception fırlatıyorum. Size methodu da elimizdeki objenin boyutunu vermekte, isEmpty methodu ise stack imizin boş olup olmadığını return etmekte.

StackD class ında data field olarak verilerimi tuttuğum Queue yu composition olarak kullandım ve aynı zamanda StackInterface interfacesini implement ettim. Gerekli olarak gördüğüm toString methodunu ve zorunlu olan push,pop,size ve isEmpty methodlarını da implement ettim bunların yanında verilere ulaşabilmek için get methodunu da implement ettim.. StackC objesini yapıp push methodu ile veri ekleyip, pop methodu ile de veri yapısının sonundan veri silebiliyoruz aynı zamanda pop methodunda eğer stackimiz boş ise exception fırlatıyorum. Size methodu da elimizdeki objenin boyutunu vermekte, isEmpty methodu ise stack imizin boş olup olmadığını return etmekte.

Genel olarak,

test class ında implement ettiğim methodlardan, writeResult methodu 4 ayrı stack classından 4 er tane farklı stack objelerini alıp içlerinde ki verileri (*pdf) de istenilen formatta (testResult_1.csv) dosyasına yazmaktadır.

readFileToA(...),readFileToB(...),readFileToC(...) ve readFileToD(...) methodları yazılmış olan 4 farklı Stack class ından her bir method tek bir sınıfa ait olan 4 er tane parametre alıp stack objelerine (test.csv) dosyasından verileri okur ve Stack yapılarımızı doldurur. Ancak yazdığım bu method okunacak dosyada 4 adet satır var ise çalışmakta verilerin parametrelere göre objelere sırası ile dosyadan 1. satır, 2. Satır, 3.satır ve 4. satır okunur. Her bir stack objesine bir satır okunur.

writeFileToA(...), writeFileToB(...),writeFileToC(...) ve writeFileToD(...) methodları Stack larimize okunmuş olan verileri ayrı ayrı sırası ile (testResult_1A.csv), (testResult_1B.csv), (testResult_1C.csv) ve (testResult_1D.csv) dosyasına yazar.

3.Analysis

Programda analizimi System.`nanoTime()` methodu ile gerçekleřtirdim. Ve stackları doldurmak için yazmış olduđum methodların baslangıcına ve bitişlerine koydum. 4 adet StackA objesine verileri okumam çok hızlıydı çok kısa sürdü. Bunu az veri olduğundan tam süresini hesaplayamadım ancak bu sınıfı yazarken ArrayList sınıfını extend etmişim ve arařtırmalarımaya göre methodlarına direk ulařtıđımızdan en hızlı sınıfın bu olduğuna söyleyebilirim. 4 adet StackD objesine veri okumamda Queue yapısını kullandım ve veri az olduğundan zaman hesaplama methodu bu veri yapısını hesaplamamdadada pek sağlıklı olduğunu söyleyemem. Ancak queue verilerini arrayda tuttuđundan hızlı olduğunu söyleyebilir. Bunun hız olarak 2. sırada gelebileceđini söyleyebilirim. 4 adet StackB objesine verileri okumamdadada ArrayList veri yapısını tanımladım ve verilerimi onda tuttum. Aynı řekilde veriler arrayda tutulduğundan hızlıdır diyebilirim. Hız konusunda 3. sırada gelir. Ve son olarak 4 adet StackC objelerine verileri okumamda Node yapısı kullandım. Her bir veri eklememde sürekli olarak tüm verinin üzerinde baştan sona gezdiğimden veri eklemede hız konusunda 4 farklı stack yapısından en son sırada gelir.

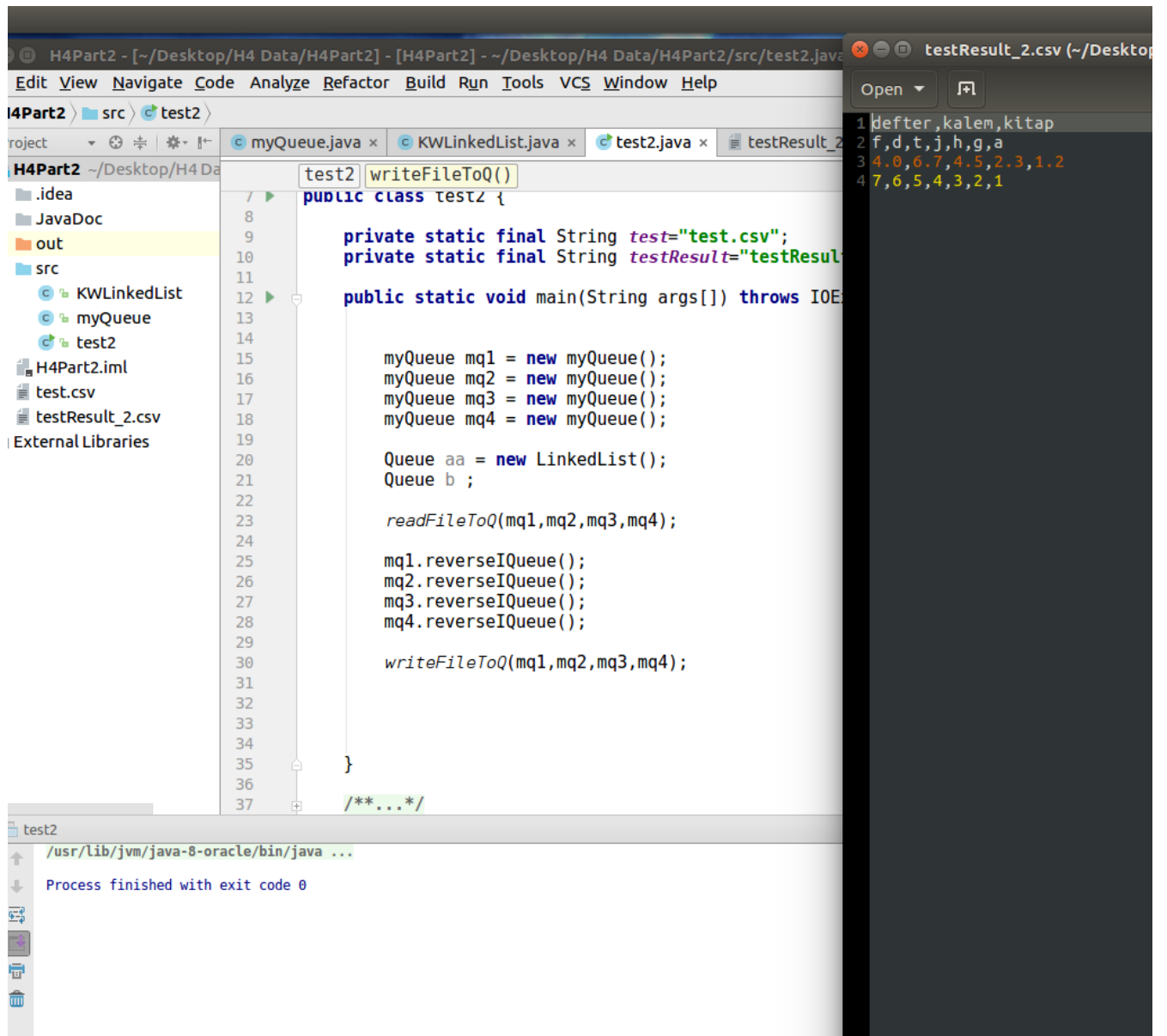
Her ne kadar stacklere verileri okuma sürelerini hesaplasamda bu method sürekli olarak deđiřkendir. Çünkü bu method sistemden sisteme deđiřebilir.

Bu hız hesaplarını hep veri ekleme yani push methodu için yaptım.


```
Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
H4Part1 > src > test
project StackInterface.java x StackB.java x test.java x StackA.java x test.csv x StackC.java
H4Part1 ~/Desktop/H4 test main()
.idea
JavaDoc
out
src
  StackA
  StackB
  StackC
  StackD
  StackInterface
  test
Tests
H4Part1.iml
test.csv
testResult_1.csv
testResult_1A.csv
testResult_1B.csv
testResult_1C.csv
testResult_1D.csv
External Libraries
31 StackD D2 = new StackD();
32 StackD D3 = new StackD();
33 StackD D4 = new StackD();
34
35 long s=0;
36 long e=0;
37 s=System.nanoTime();
38 readFileToA(A1,A2,A3,A4);
39 e=System.nanoTime();
40 System.out.println("A stack veri okuma nanosaniye= "+(e-s));
41
42 s=System.nanoTime();
43 readFileToB(B1,B2,B3,B4);
44 e=System.nanoTime();
45 System.out.println("B stack veri okuma nanosaniye= "+(e-s));
46
47 s=System.nanoTime();
48 readFileToC(C1,C2,C3,C4);
49 e=System.nanoTime();
50 System.out.println("C stack veri okuma nanosaniye= "+(e-s));
51
52 s=System.nanoTime();
53 readFileToD(D1,D2,D3,D4);
54 e=System.nanoTime();
55 System.out.println("D stack veri okuma nanosaniye= "+(e-s));
56
57 writeResult(A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4,D1,D2,D3,D4);
58
59 //writeFileToA(A1,A2,A3,A4);
60 //writeFileToB(B1,B2,B3,B4);
61 //writeFileToC(C1,C2,C3,C4);
62 //writeFileToD(D1,D2,D3,D4);
test
/usr/lib/jvm/java-8-oracle/bin/java ...
A stack veri okuma nanosaniye= 152007
B stack veri okuma nanosaniye= 110899
C stack veri okuma nanosaniye= 401541
D stack veri okuma nanosaniye= 100380
Process finished with exit code 0
```

Q2-/

1. Test Cases



Iterative olarak yazdığım `reverseIQueue` methodu yeni bir node oluşturmada verisini ters çeviriyor.

2. Running and Results

4 farklı myQueue objesi tanımlayıp readFileToQ() methodu ile 4 farklı objeye dosyadan veriler okunur.(her bir objeye bir satır.) Ondan sonra her bir obje üzerine reverseIQueue() methodu çağrılarak obje üzerinde ki veri reverse edilir. WriteFileToQ() methodu ile de (testResult_2.csv) sonuçlarıbu dosyaya yazdırılır.

NOT: Hocam recursive olan method içinde column ları ters cevrilmesini sağladım ancak satırlarda ters ceviremedim vaktim yetmedi.

NOT: recursiveQueue methodu myQueue objesi üzerinden çağrılmaktadır.!!

https://github.com/mstfbngl/HW04_141044077