Gebze Technical University Computer Engineering

CSE 222

2017 Spring

HOMEWORK 04 REPORT

MUSTAFA BİNGÜL 141044077

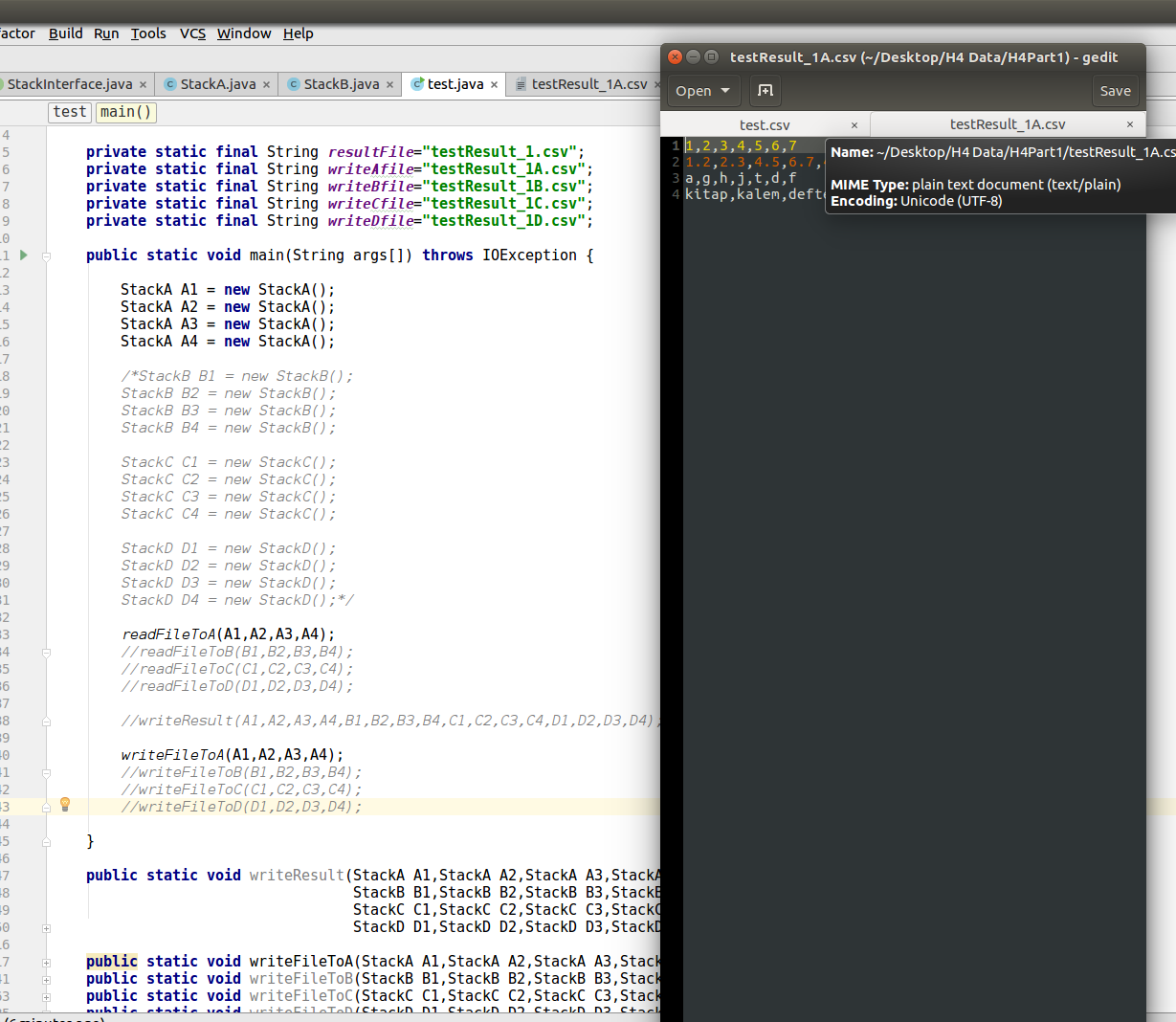
Course Assistant:

Nur Banu ALBAYRAK

Q1-/

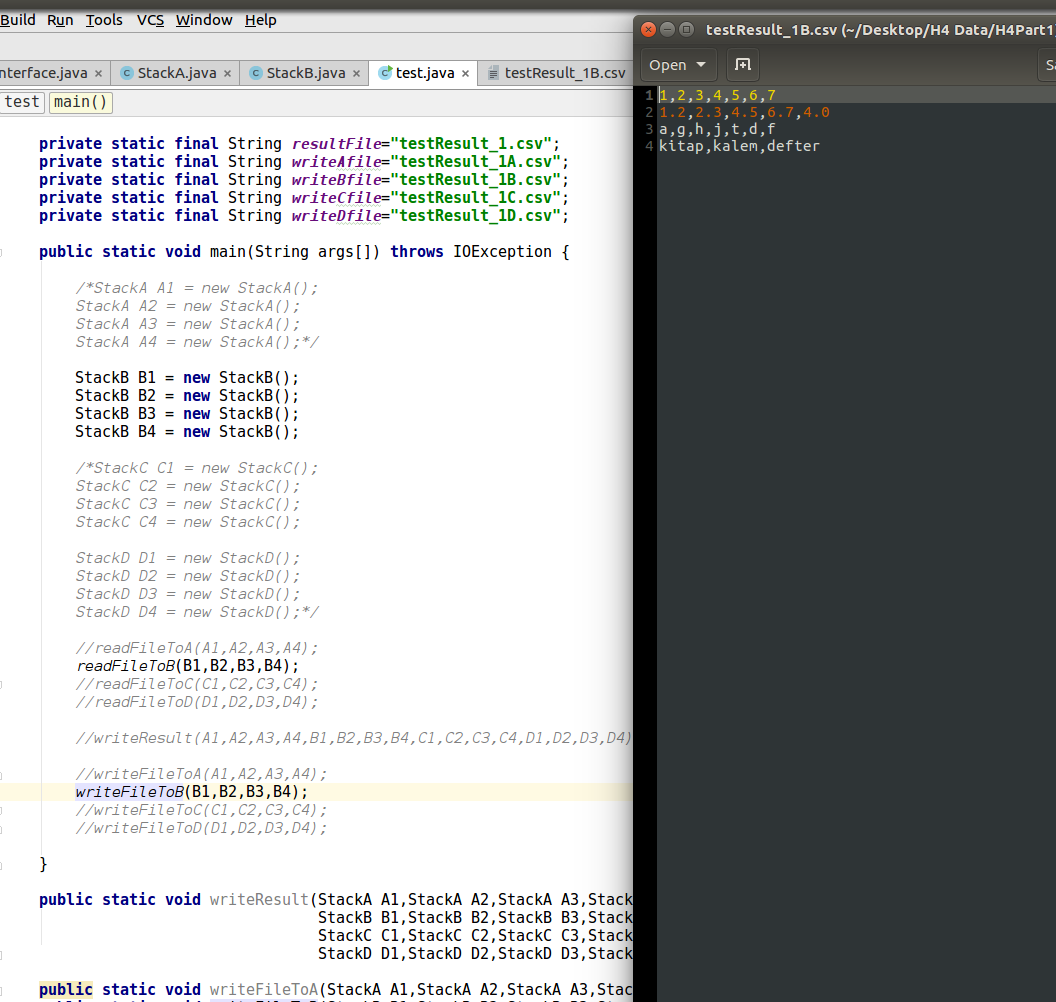
1.Test Cases

StackA



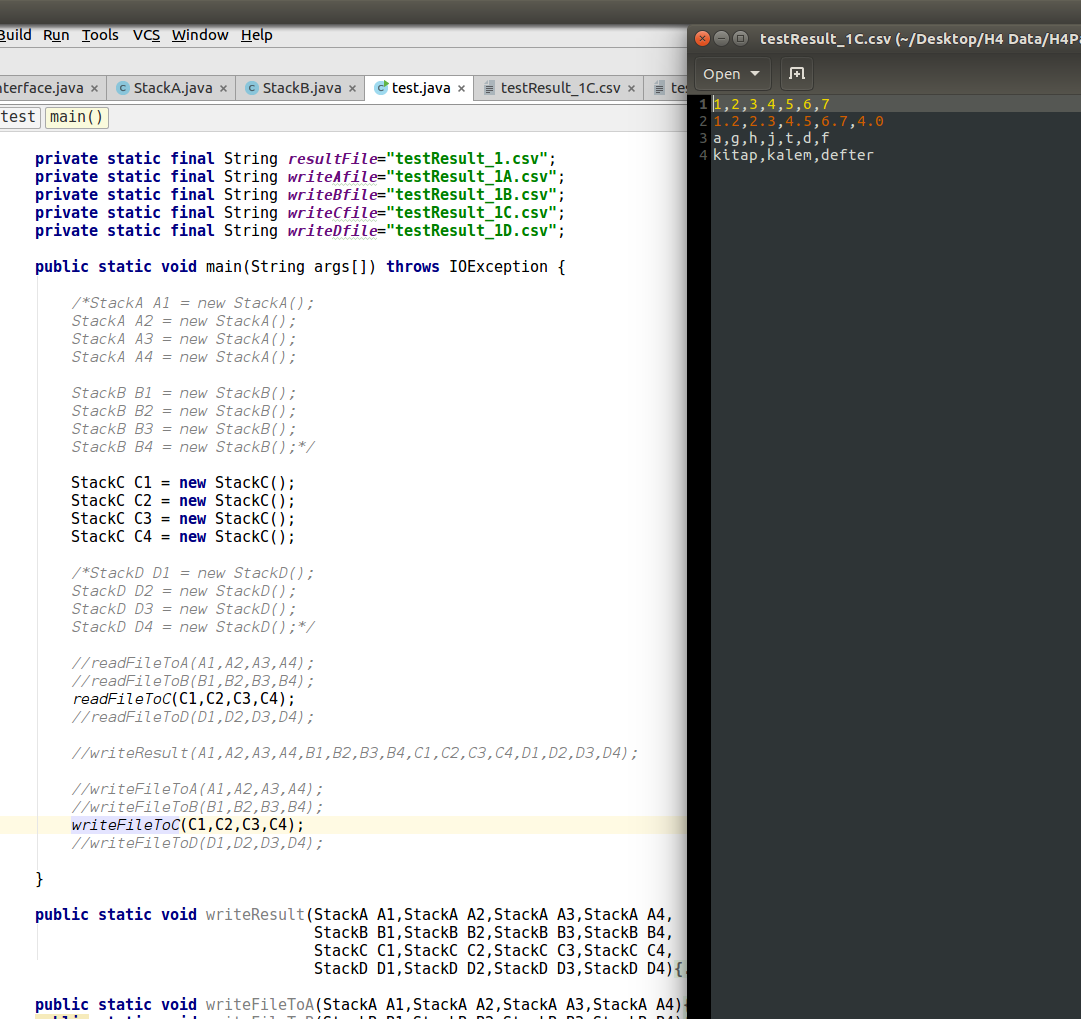
4 Adet StackA tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackB



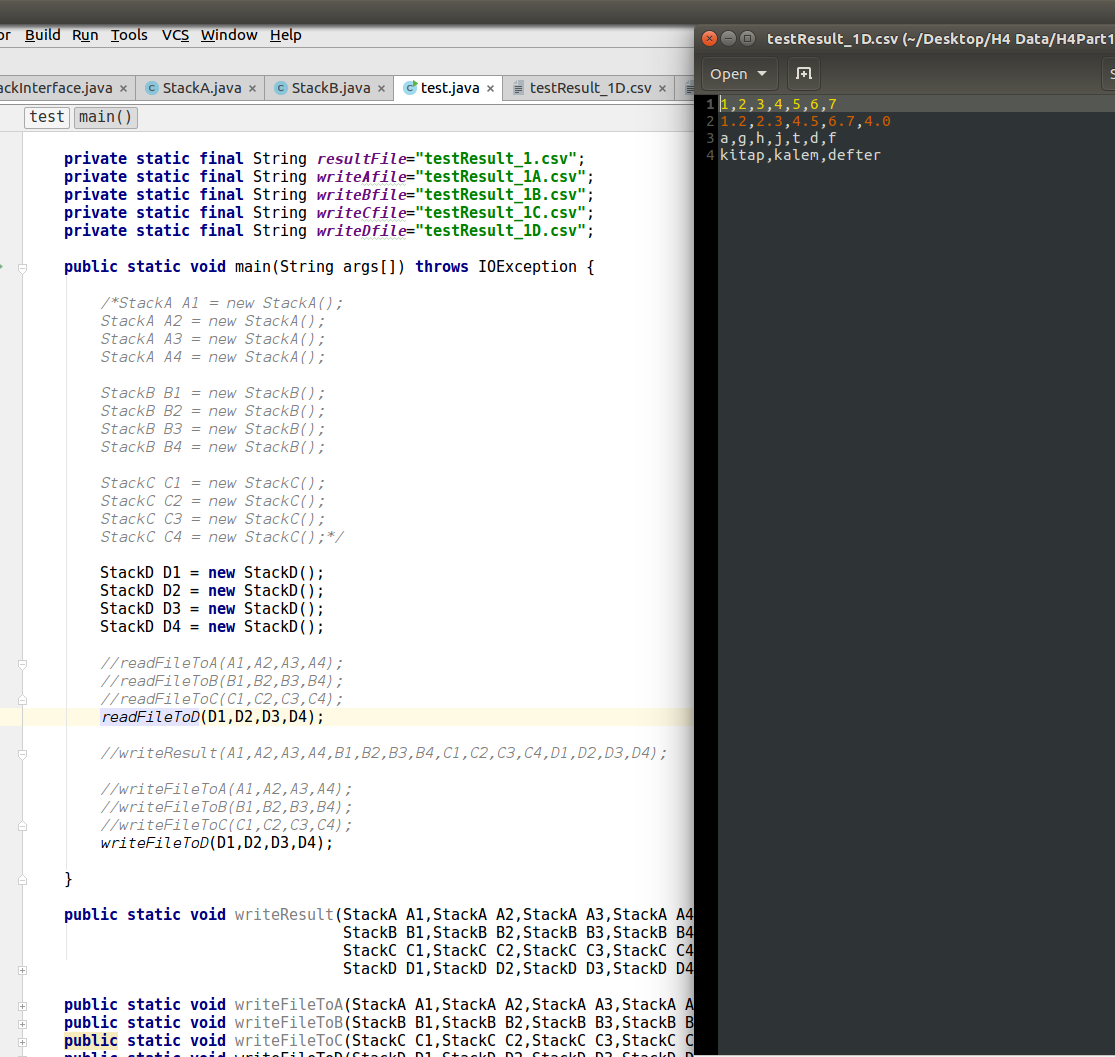
4 Adet StackB tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackC



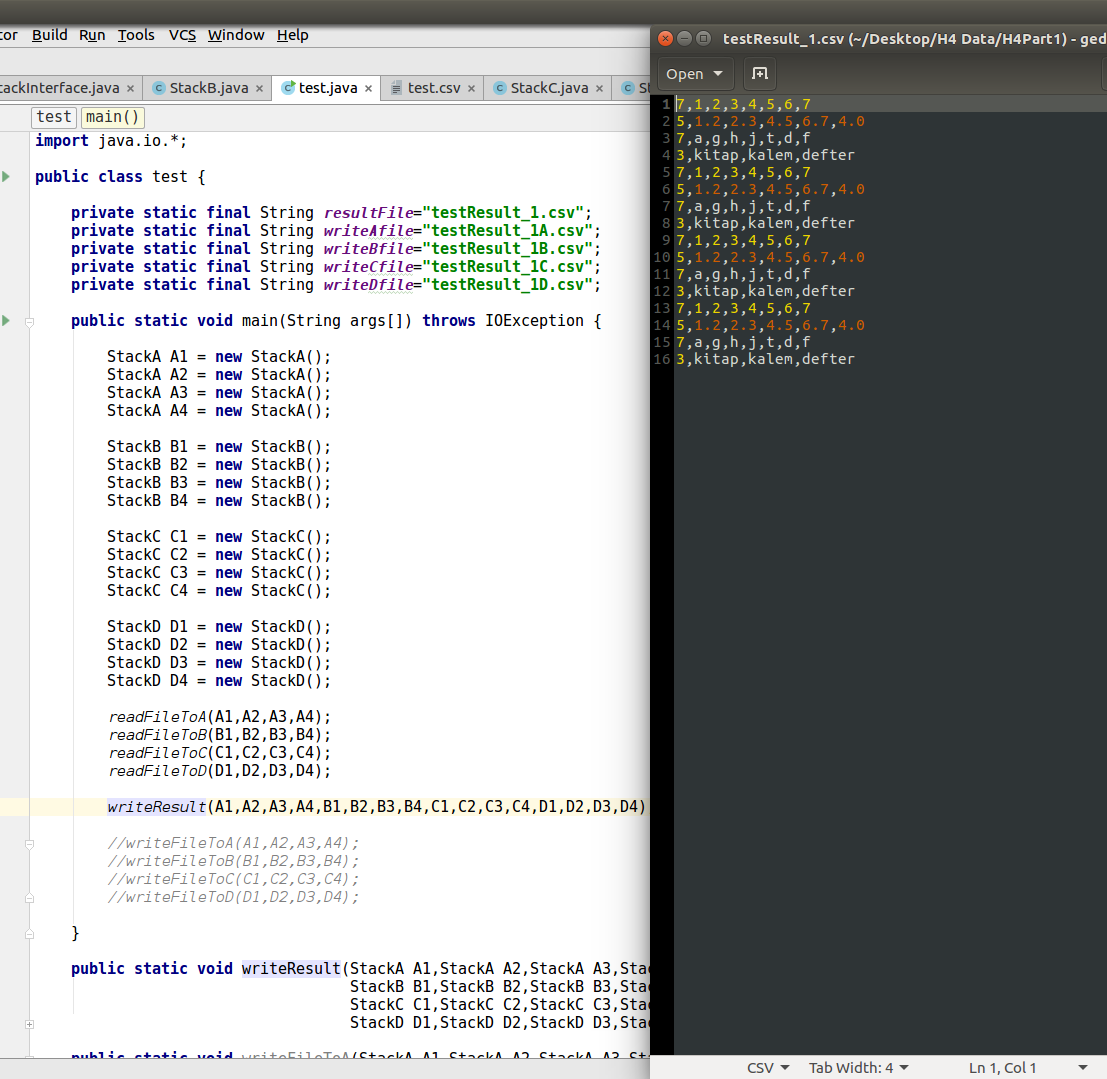
4 Adet StackC tanımlayıp her bir satırı ayrı ayrı Stacklere okuyup. Kendi dosyasına yazdırıyorum. Sonra onu kendi (.csv) dosyasına yazdırıyorum. Çalışıyor.

StackD



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.



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 ımda 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 çalışmakda 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 gerçekleştirdim.

Ve stackları doldurmak için yazmış olduğum 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 etmiştim ve araştırmalarıma göre methodlarına direk ulaştığımızdan en hızlı sının bu olduğunu söyleyebilirim.

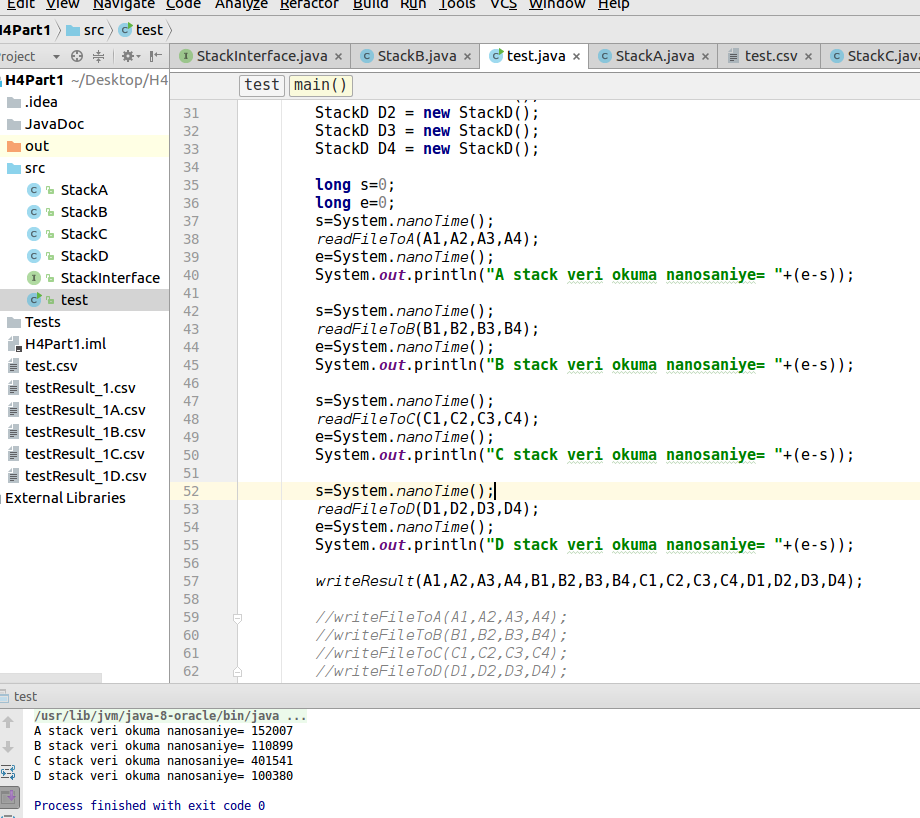
4 adet StackD objesine veri okumamda Queue yapısını kullandım ve veri az olduğundan zaman hesaplama methodu bu veri yapısını hesaplamamdada pek sağlıklı olduğunu söyleyemem. Ancak queue verilerini arrayda tuttuğundan hızlı olduğunu söyleyebilir. Bunun hız olarak 2. sırada gelebileceğini söyleyebilirim.

4 adet StackB objesine verileri okumamdada ArrayList veri yapısını tanımladım ve verilerimi onda tuttum. Aynı şekilde veriler arrayda tutulduğundan hızlıdır diyebilirim. Hız konusunda 3. sırada gelir.

Ve son olarak 4 adet ScackC 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.

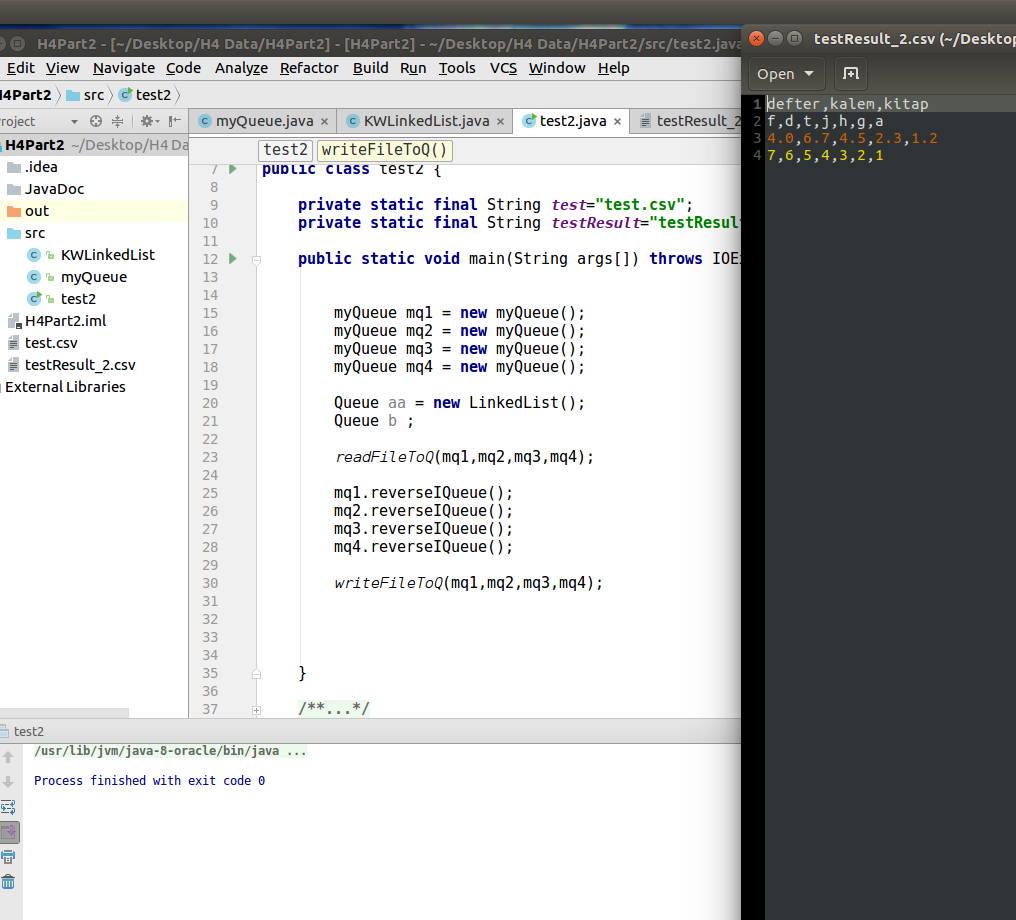
Her ne kadar stacklere verileri okuma sürelerini hesaplasamda bu method sürekli olarak değişkendir. Çünkü bu method sistemden sisteme değişebilir.

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



Q2-/

1. Test Cases



Iterative olarak yazdığım reverseIQueue methodu yeni bir node oluşturmadan 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