Gebze Technical University Computer Engineering

CSE 222

2017 Spring

HOMEWORK 09 REPORT

MUSTAFA BİNGÜL 141044077

Course Assistant:

Ahmet SOYYIĞIT

1-Problem Solution Approach

Verilen sınıfları ve interfaceleri kullandım. Ödevde istenen abstract bir sınıf implement etmemiz. Ve istenilen methodları implement etmemiz sonrada test etmemis isteniyor ödevde. İstenilen AbstractGraphExtended sınıfı AbstractGraph sınıfını extend etmekte.

Ödevde implement etmemiz gereken methodlar.

AddRandomEdgesToGraph(int edgeLimit) methodu verilen edgeLimitine kadar random bir sayı belirleyip o random sayı adedinde edge eklemekte ve eklediği edge lerin vertexleride 0 ile o random sayı arasında belirlenmektedir. Öncelikle bu methodun içinde eklenen edgeleri daha öncedende objeye eklenip eklenmediğini kontrol etmek(isEdge methodunu kullandım.) için yani bir edge yi bir kez eklemek için bir HashSet objesi kullandım obje üzerinde olan tüm edgeleri oraya ekledim ve yeni oluşturduklarımıda bu HashSet objesine ekleyerek obje üzerinde bir edgenin tek kez olmasını sağladım. Ve return değeri olarakda sadece yeni eklenen edge sayısını döndürdüm.

BreadthFirstSearch(int start) methodu çağırılan graph objesi üzerinde verilen start vertexiyle ona bağlı olan ve en yakınından itibaren tüm vertex leri gezmektedir. Bu method için kitaptaki implementasyonu kullandım. Vertexleri gezme sırası için bir Queue objesi kullanılmıştır. Vertexlerin parentlerinin tutulduğu bir dizi ve gezilip gezilmediğini belirlemek içinde bir boolean array i kullanılmıştır. Bir vertexden başlanıp yakınındaki vertexler gezilmektedir. Ve sonunda vertexlerin parenlerini array olarak return etmektedir.

writeGraphToFile (String fileName) methodu okuduğumuz formatta aynı zamanda ödev pdf inde belirtilen formatta dosya ya yazan method.

Edge leri tuttuğum bir set objesi oluşturup obje üzerindeki hep bir vertex I gezerek bu set üzerine ekledim ve eklerken tersinin oluğ olmadığını kontrol edip ekledim böylece yönsüz graphlardaki birden fazla kez yazma sorununu engellemiş oldum.

GetConnectedComponentUndirectedGraph() methodu çağırılan graph objesi üzerindeki birbiri ile bağlı yönsüz graphları Graph dizisi olarak return etmektedir. Methodu yazarken öncelikle bir helper yardımcı method kullandım(helperConnected(int start, Boolean[] visited)). Helper method verilen vertex den başlayıp bağlı olduğu tüm vertexleri gezip ve onları visited olarak işaretlemektedir. Asıl işi yapan methodumuzda bir tane HashSet objesi ve gezilmiş olanlar için Boolean dizisi oluşturdum. Önce bir kez gezerek obje üzerinde kaç adet connected olmayan graph sayısını buldum.

Ondan sonra ise objenin ListGraph yada MatrixGraph olmasına göre iki ayrı durum için değerlendirme yaptım. Method içinde oluşturduğum her iki graph objesi içinde duruma göre değerlendirip graph sayısı kadar yer aldım. Sonra da obje üzerindeki bağlı olmayan graphları bu graphlara ekledim.

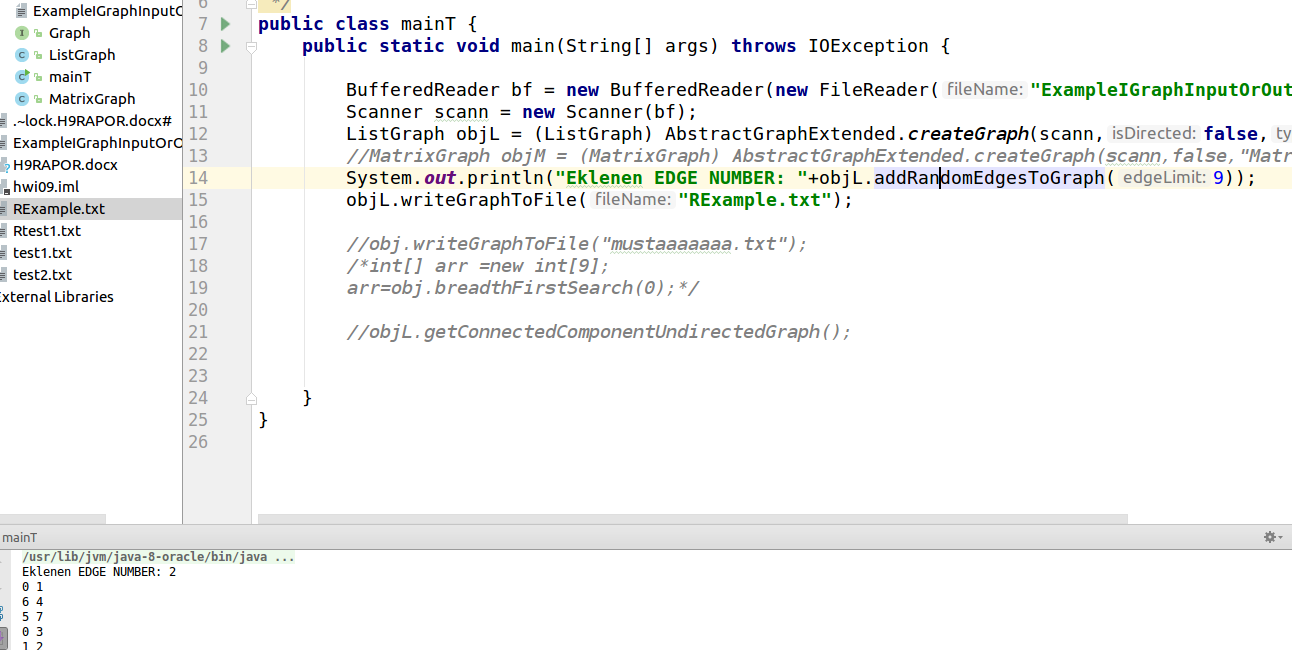
HelperConnected(int start, Boolean[] visited) methodu yardımcı method verilen vertex te ve o vertexe bağlı olan vertexleri gezip işaretleyen method. GetConnectedComponentUndirectedGraph() methoduna yardım için yazılmıştır.

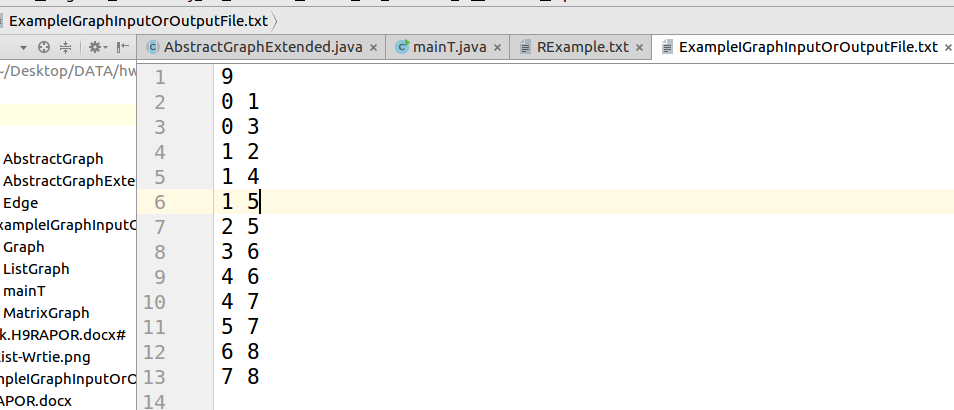
IsBipartiteUndirectedGraph() bu methodu implement etmedim.

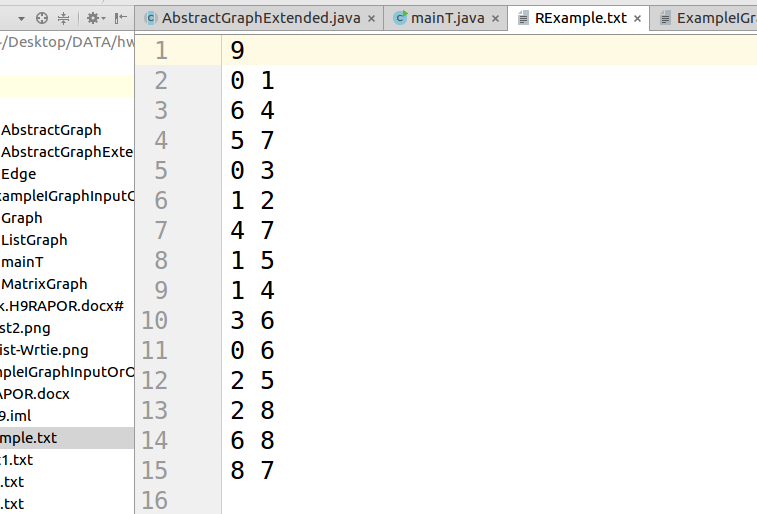
2- Test Cases

CASE-1:

ListGraph objesine ödevin yanında verilen example.txt den edge leri okuyup sonra tekrardan addRandomEdgesToGraph ile verilen aralıkta yeni edgeler ekledim. Ekliyor.



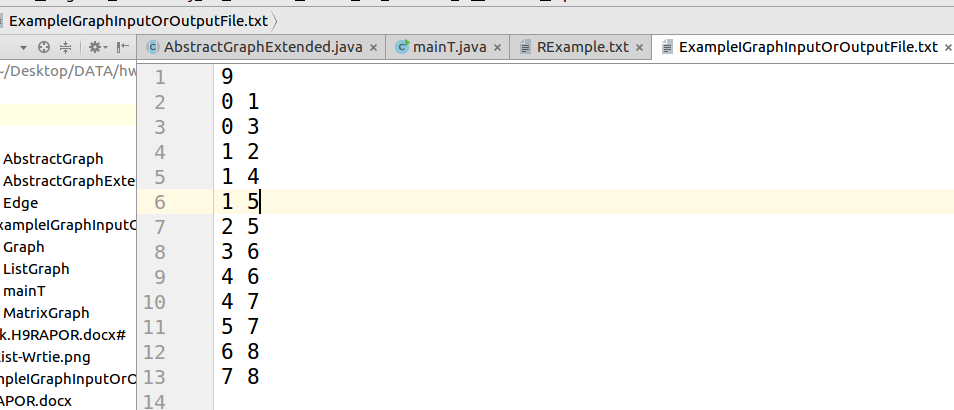
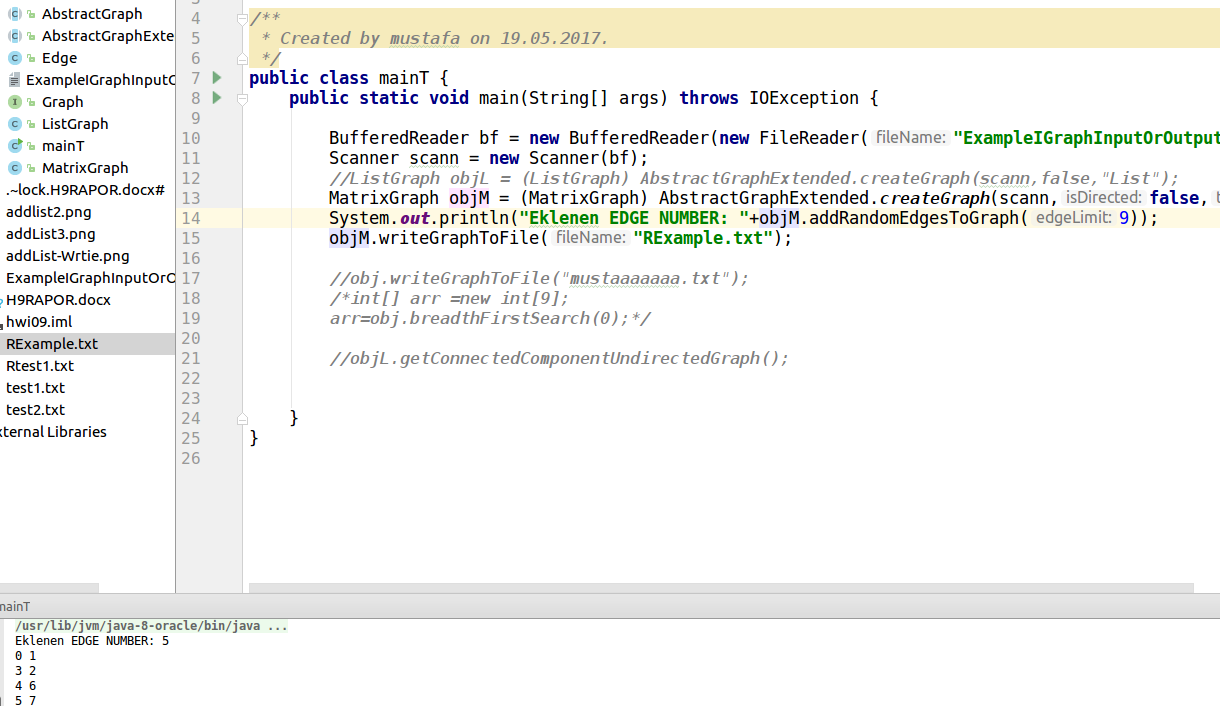




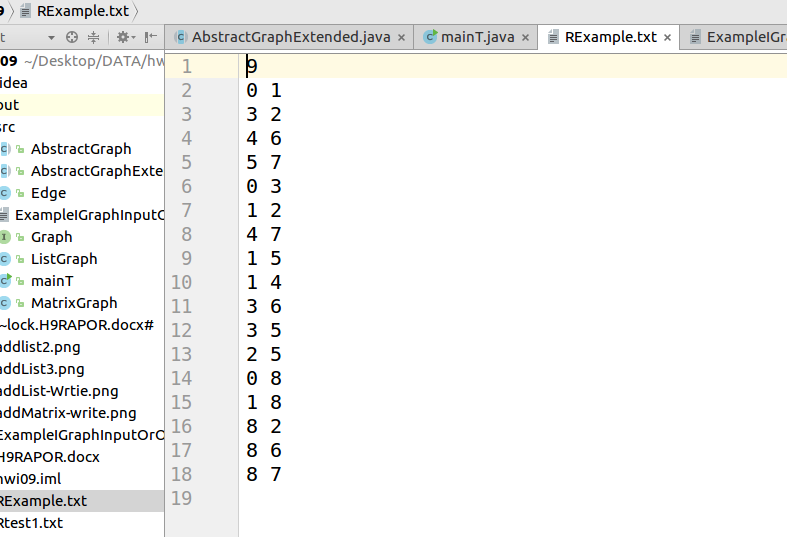
ListGraph a limit olarak 9 verdim vertex sayısı kadar çalıştırdıktan sonra 2 adet yeni edge ekledi ve return değeri olarak console da gösterdim. Result dosyasındada Rexample.txt dede 2 tane fazla edge görülmekte.

CASE-2:

MatrixGraph objesi için addRandomEdgesToMatrix methodunu çağırdım ve eklediği edge sayısını console ve edge lerin vertexlerini de Result dosyasına yazdırdım.

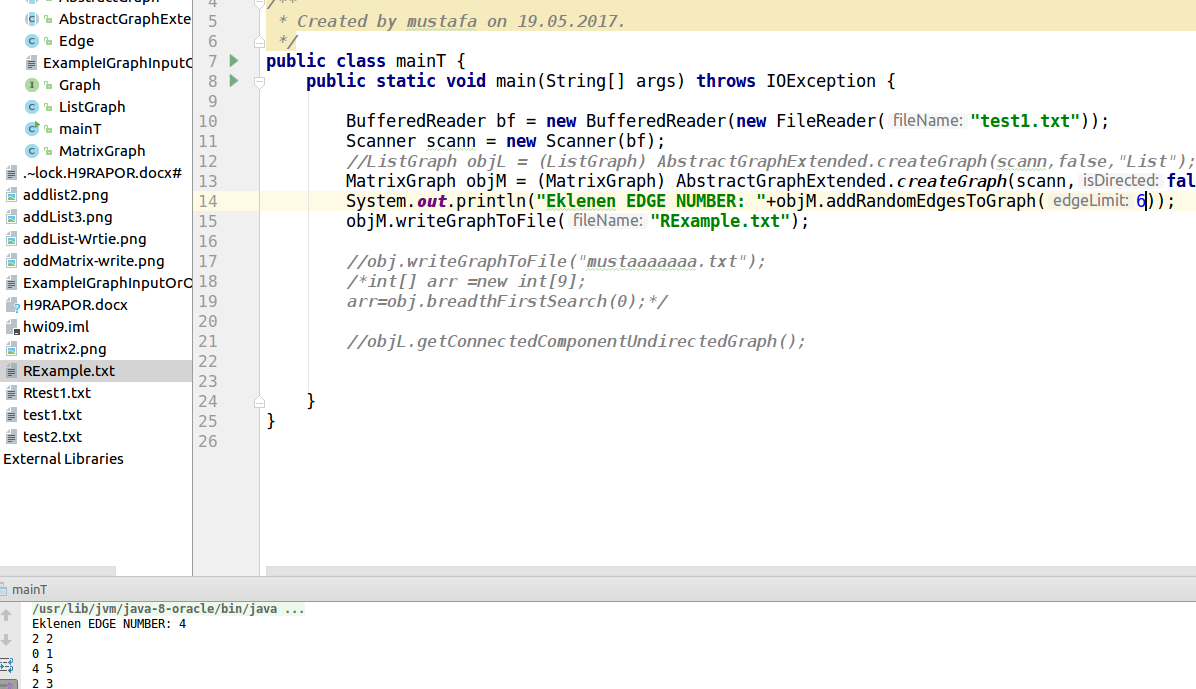
 Example.txt

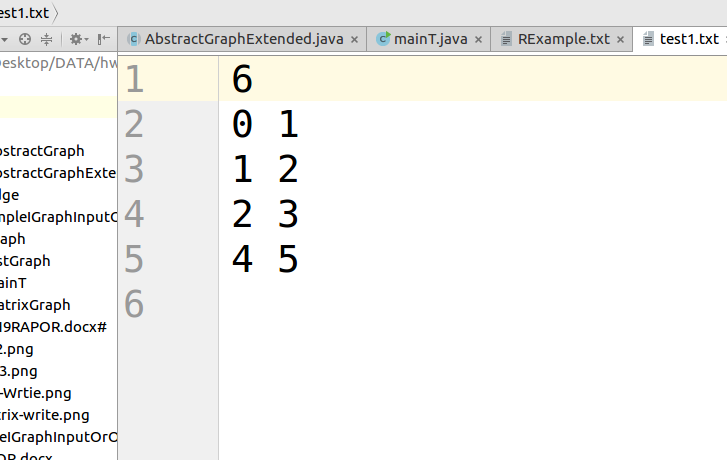
ResultExample.txt

5 adet yeni edge eklenmiş durumda.

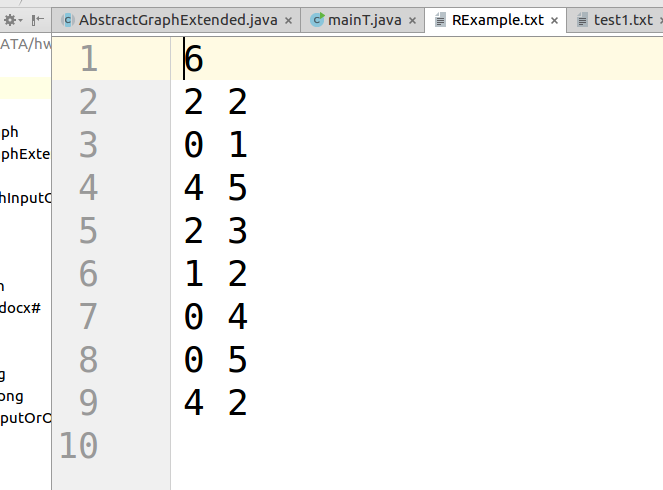
Text1.txt dosyasından kendi testim kendi vertexlerimi yazdım.

MatrixGraph objesi için.





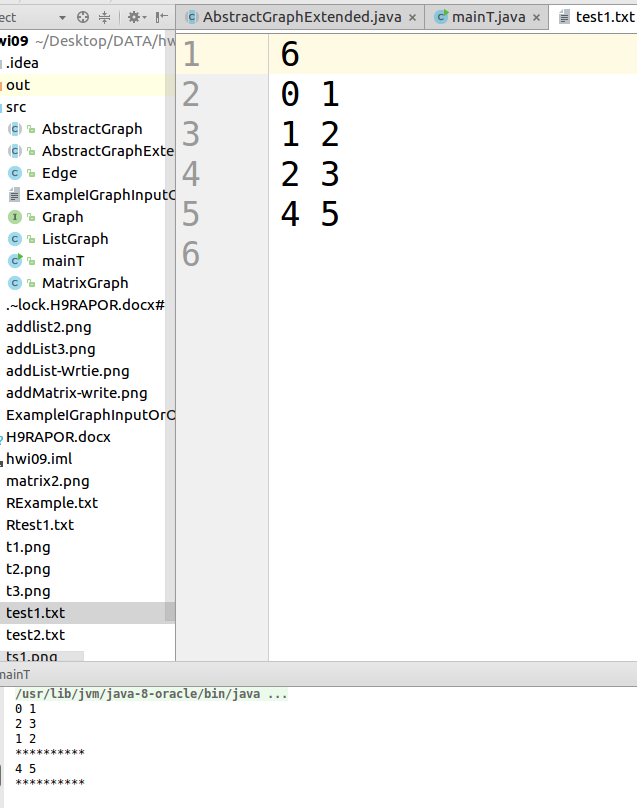
Test1.txt

belirlenen random sayı sonrasındaki ekleme 4 adet yeni edge daha eklendi.

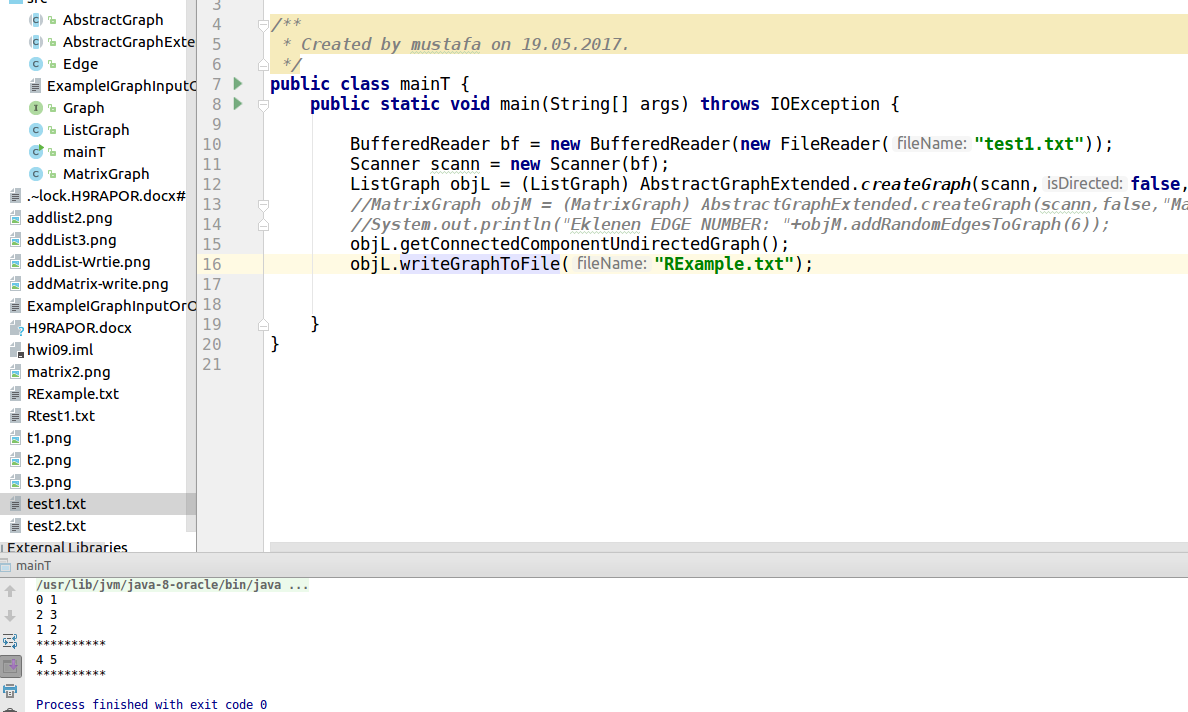
Aynı edge ler hiçbir zaman eklenmemektedir.

CASE-3

Çağırılan ListGraph objesi üzerindeki bağlı olmayan yönsüz graphları console da yazdırdım. Ve bağlı olmayanları ayrı ayrı yazmakta.

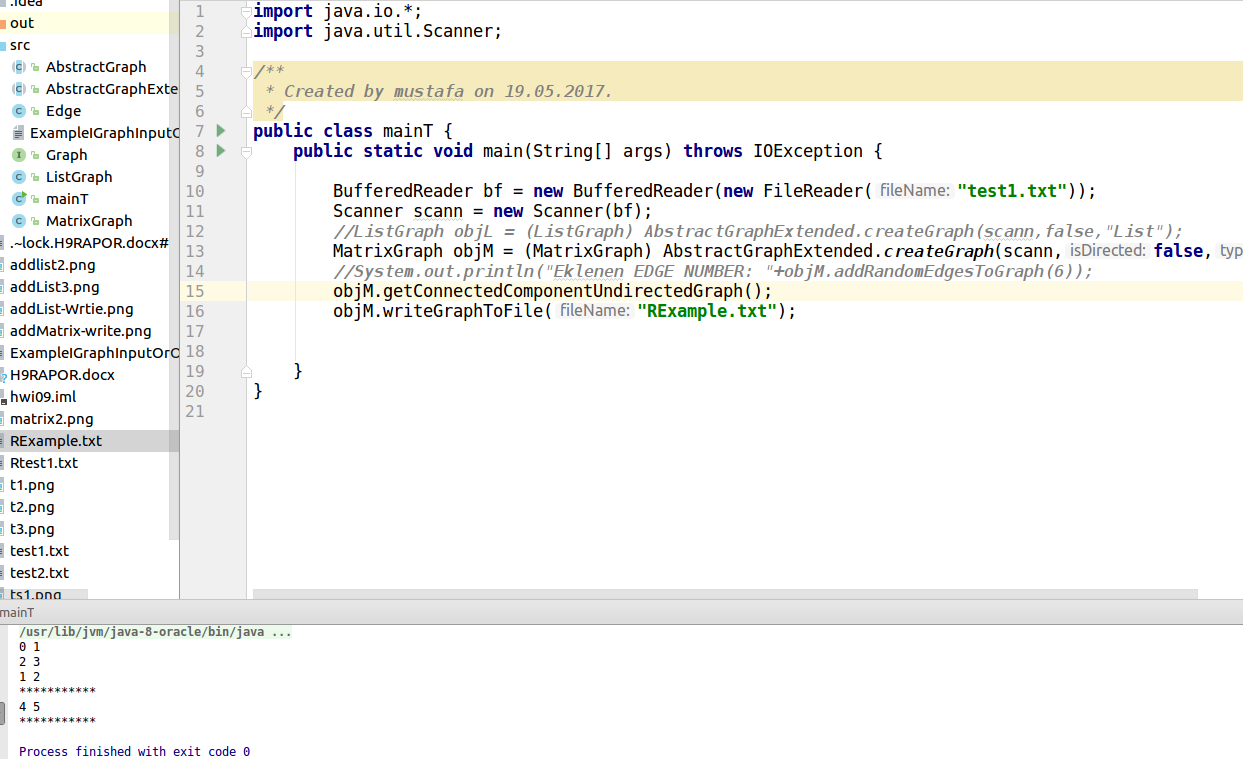


Test1.txt dosyamızda bulunan graph(edge ler).



Consoleda da görüldüğü gibi ayrı olarak yazmakta.

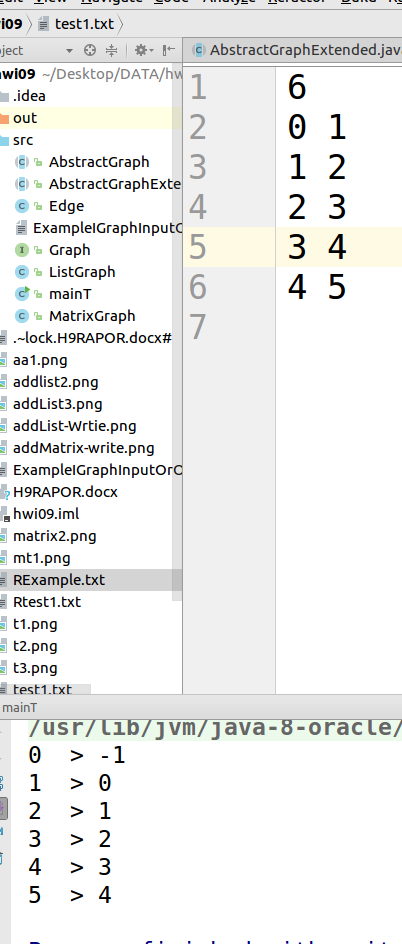
Aynı durumu MatrixGraph objesi için gerçekleştirdim.



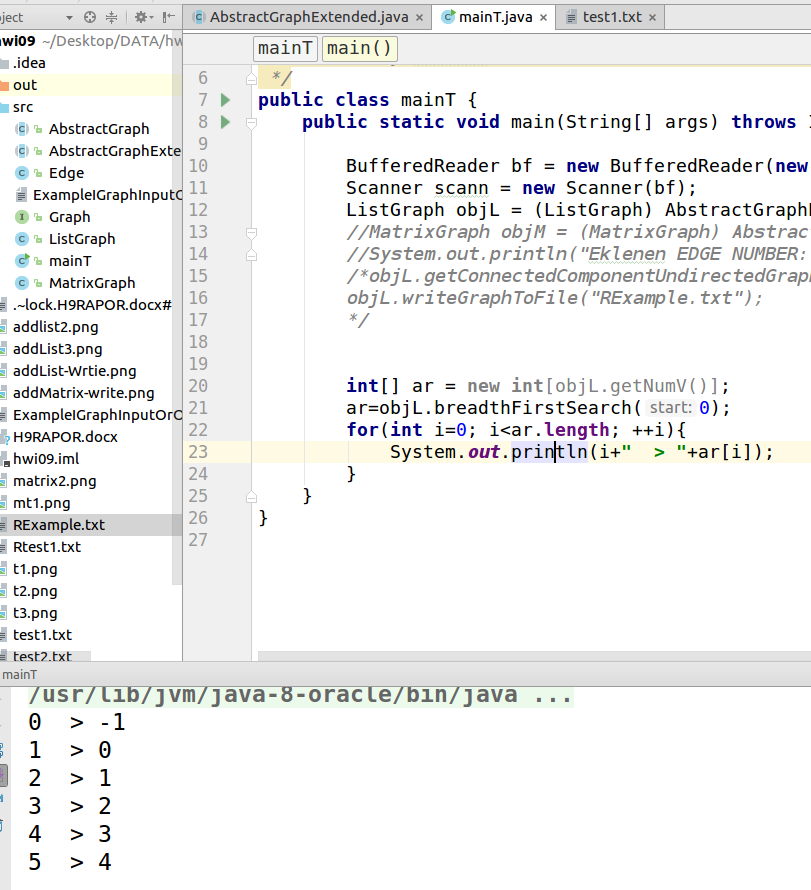
Aynı şekilde çalışmakta.

CASE-4

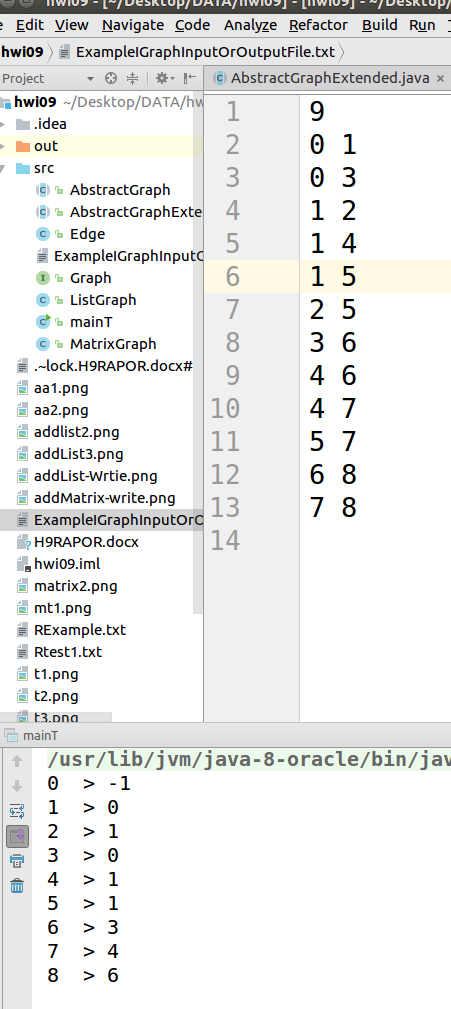
ListGraph Objesi için breadthFirstSearch methodu parentlerini return etmekte.

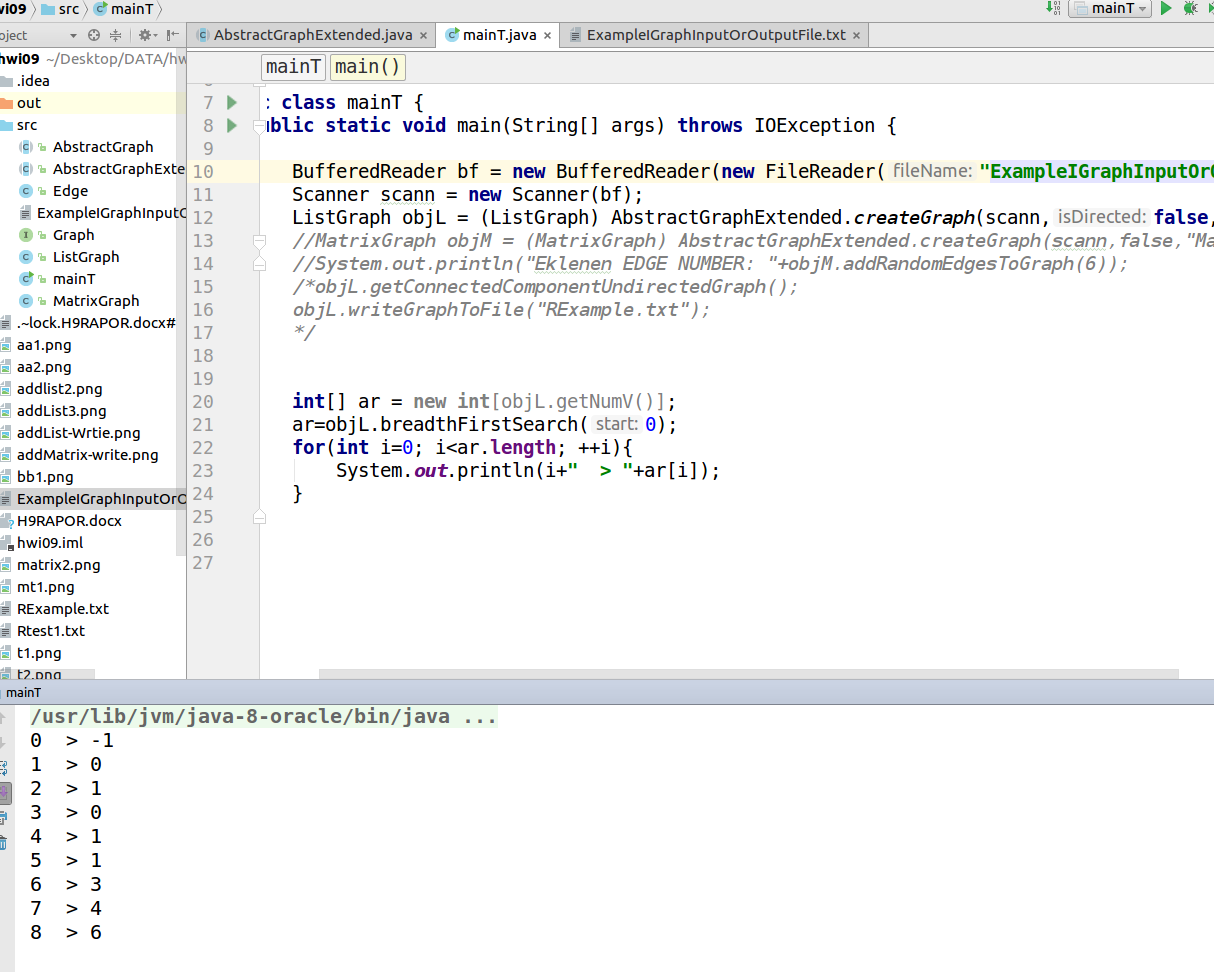


Parentleri console da görülmekte.



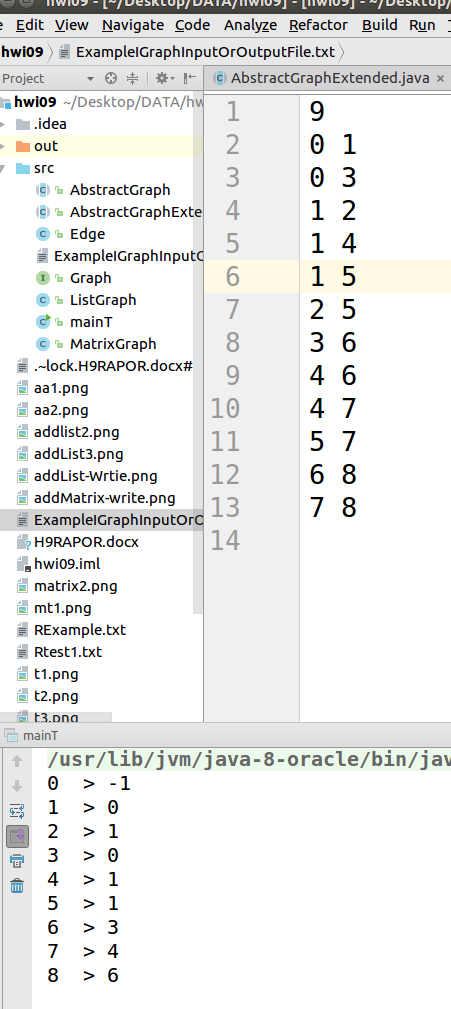
Ödevde verilen Example.txt dosyası ile de denedim.

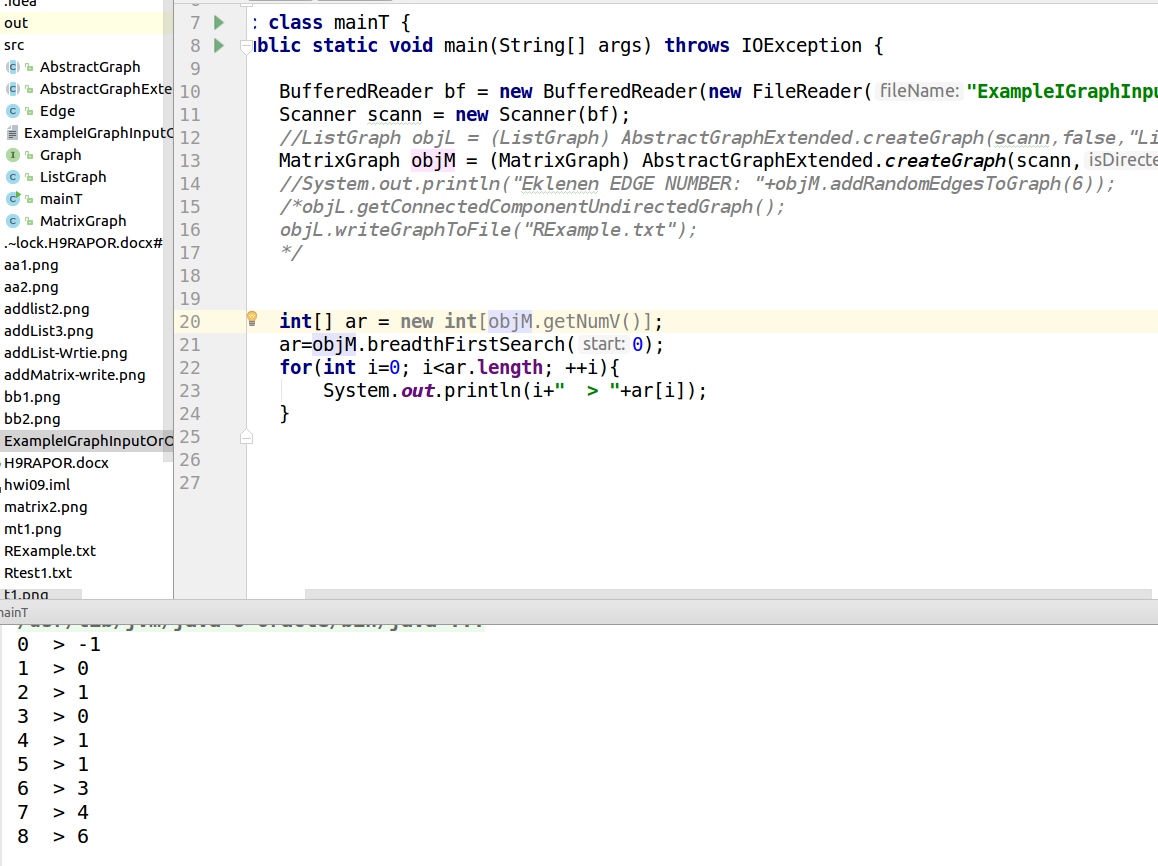




Console da parentleri gözükmekte.

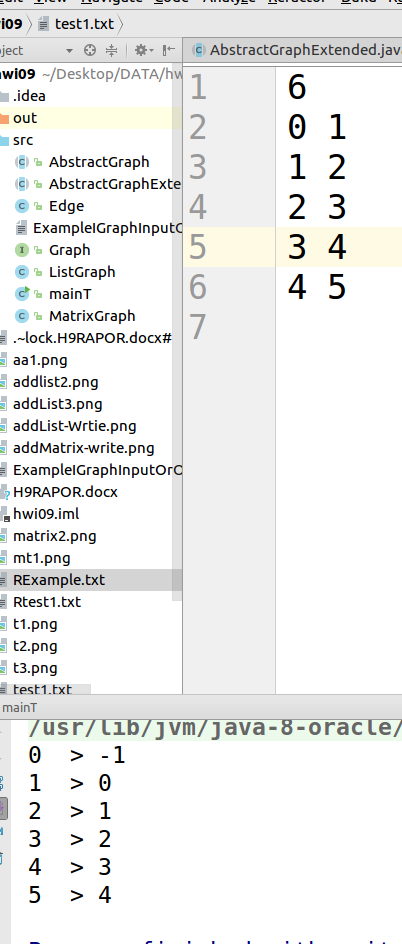
Şimdi bir de aynısını MatrixGraph objesi için deneyelim.

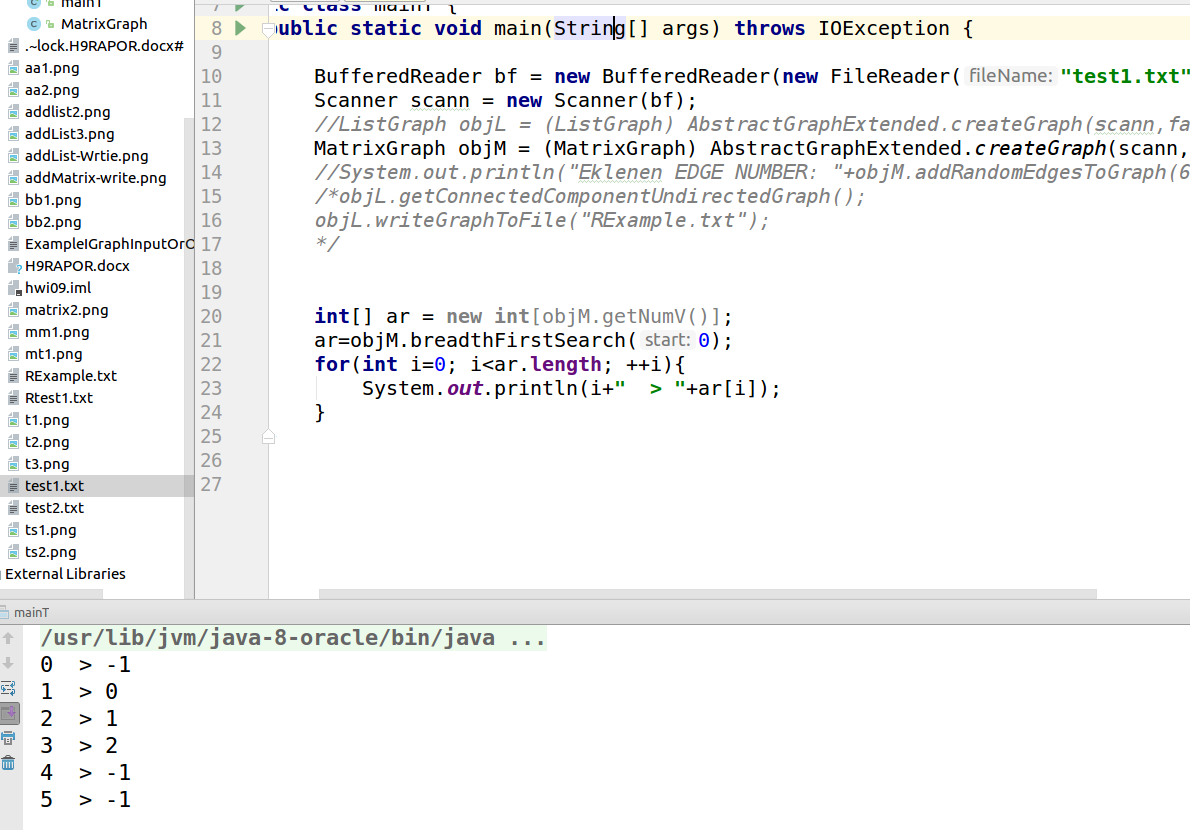




Parentleri consoleda gözükmekte.

Birde MatrixGraph objesini kendi yazdığım test ile denedim.





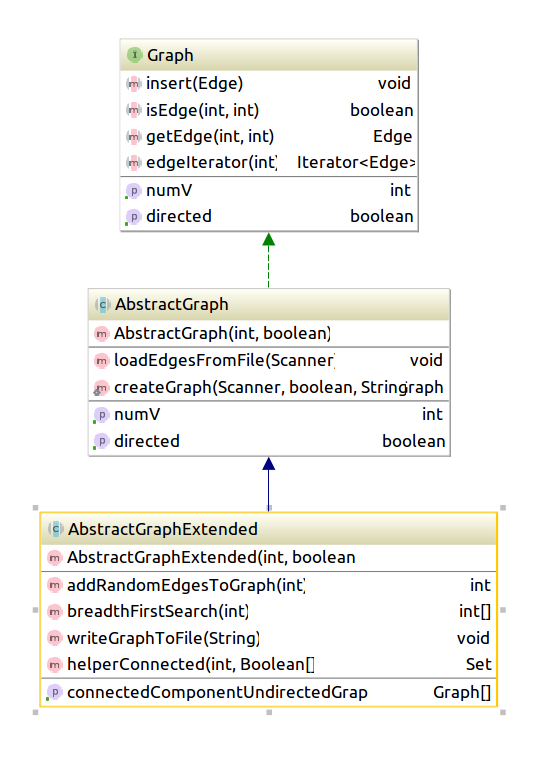
BreadthFirstSearch methodu bu obje ile de çalışıyor.

3-Running Command and Results

Case durumlarında tüm çalışma durumlarınıda adım adım gösterdim.

Buraya da eklemedim tekrardan.

4-Class Diagram



https://github.com/mstfbngl/141044077\_hw09