

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 2 REPORT

**SEFA NADİR YILDIZ
131044031**

Course Assistant:
Fatma Nur Esirci
Mehmet Burak Koca
Tuğbagül Altan Akın

1 INTRODUCTION

1.1 Problem Definition

- **Part 1**

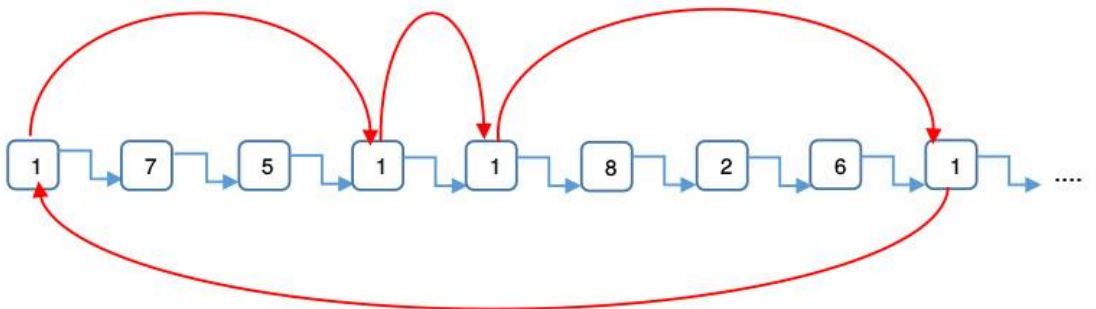
GTÜ Bilgisayar Mühendiliği kredili ders yapısını diyagramı csv dosyasına “ ; ” ayracı kullanılarak semester, course code, course title, ECTS credits, GTU credits, H+T+L başlıkları altında bilgiler kaydedilmiştir. Bu verileri **linkedlist** yapısında tutup bu yapı üzerinde getByCode, listSemesterCourses, getByRange methodlarının implement edilmesi istenmektedir.

- **Part 2**

Bu partta yazacağımız sınıfı LinkedList sınıfında extend edip generic olacak şekilde yapmamız istenmektedir. **disable**, **enable** ve **showDisabled** methodlarının extend ettiğimiz linkedlist üzerinde implement edilip çalışması beklenmektedir. Disable edilen linkedlist elemanlarının **get** , **set**, **size**, **remove** ve **listIterator** methodlarının bu elemanların üzerinde çalışmaması istenilmektedir.

- **Part 3**

Bu partta LinkedList sınıfı ihtiyacımız olan sınıf ve methodlar ile yeniden kendimizin implement edilip bize ait bir LinkedList sınıfını kullanmamız istenilmektedir. Listeye dersler ile birlikte hangi yarı yıla ait olduğu bilgisi eklenmektedir. Bu ekleme sırası single LinkedList yapısında olup aynı yarı yıla ait derslerin ise birbirleri ile circle LinkedList yapısında olması beklenmektedir. add, remove, next, nextInSemester, size methodlarının kesinlikle olup kendi LinkedListimiz üzerinde implement edilip çalışması beklenmektedir. Yapısı aşağıda gösterilen şekildeki gibi olacaktır.



1.2 System Requirements

PART 1

Bu kısım için semester, course code, course title, ECTS credits, GTU credits, H+T+L bilgilerini içeren bir dosya gerekmektedir. Dosya uzantısı .csv olmalıdır. Bu dosyadaki bilgileri parse edip tutabilmek için bir yapı lazımdır. Bunun için GTUCourseStructure adında dosyadaki bilgileri tutan bir sınıfım mevcuttur.

Bu sınıf sadece bir derse ait bilgileri tutmaktadır. Dosyadaki bilgilerin tamamının parse edilip tutabilmesi için bu sınıfın arrayi gerekmektedir. Bu arrayi oluştururken dosyanın satır sayısını hesaplayan yardımcı bir methoda ihtiyaç olacaktır.

GetByCode methodu ders koduna karşılık gelen tüm ders adlarını belilemesi gerekmektedir. ListSemesterCourses methodu belirten semester yılındaki tüm dersleri belirlemesi gerekmektedir. GetByRange methodu LinkedList'te belirtilen indeksler arasındaki tüm dersleri belirlemesi gerekmektedir.

PART 2

Bu kısımda kendimize ait methodları içeren bir LinkedList sınıfı olması için yazacağımız sınıfın LinkedList sınıfından extend edilmesi gerekmektedir. Yeni LinkedList sınıfımızın disable, enable ve showDisable methodları bulunmalıdır.

Disable methodunu kullandığımız zaman işlevsiz duruma gelecek olan elemanın datasını ve konumu tutmak için bu bilgileri saklayacak yapılara ihtiyacı vardır. Bu yeni methodların disable veya enable methodları yanlış eleman veya indeks gönderilmesi durumunda hata vermesi beklenmektedir.

PART 3

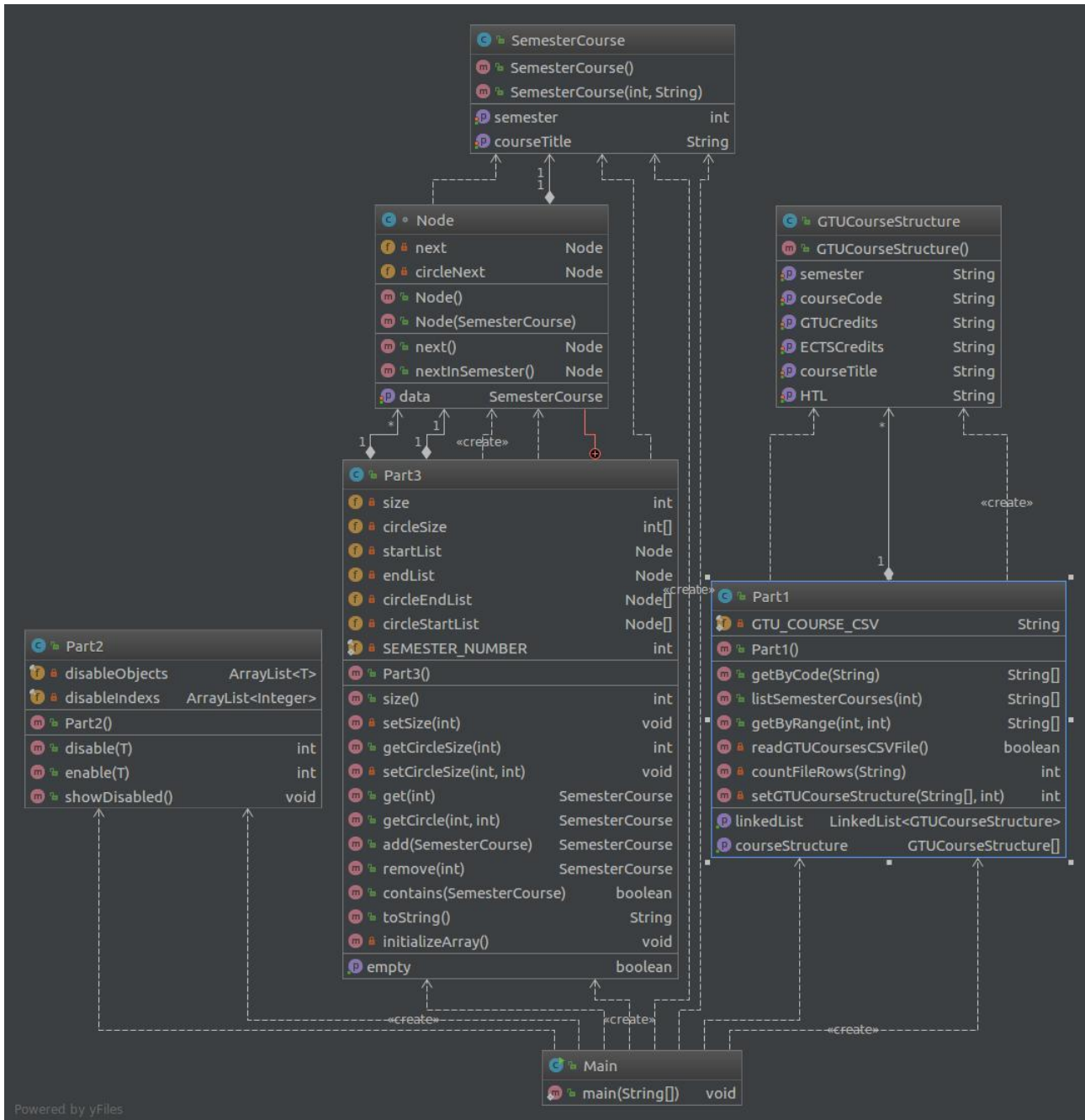
Bu kısımda LinkedList yapısını yeniden yazmamız gerekmektedir. Yazacağımız bu LinkedList derslerin semester ve ders adı bilgisini içermesi gerekmektedir. Yapısı bir dönem için ders seçimi yapan öğrencinin bilgi sistemine benzetilebilir.

Dersler birbirleri ile single yapıda, aynı semester döneminde ait dersler ise birbirleri arasında circle yapıda olmalıdır. Bağlantıları sağlamak için Node yapısı gerekmektedir. Bu yapıda next ve circle next adımlarını gösteren üyeleri ile birlikte içinde bilgi tutacak bir üyesi daha gerekmektedir.

Oluşturduğumuz LinkedList yapısının eleman eklemek için add methodu, eleman silmek için remove methodu, bir sonraki dersi göstermek için next, aynı semester döneminde bir sonraki nodu göstermek için nextInSemester methodu ve kaç tane ders eklendiğini bulabilmek için size methodu mutlaka bulunmalıdır. Yapı arasındaki ilişkini kurulabilmesi için baş ve kuyruk kısımları iyi kontrol edilip bilgileri tutulması gerekmektedir.

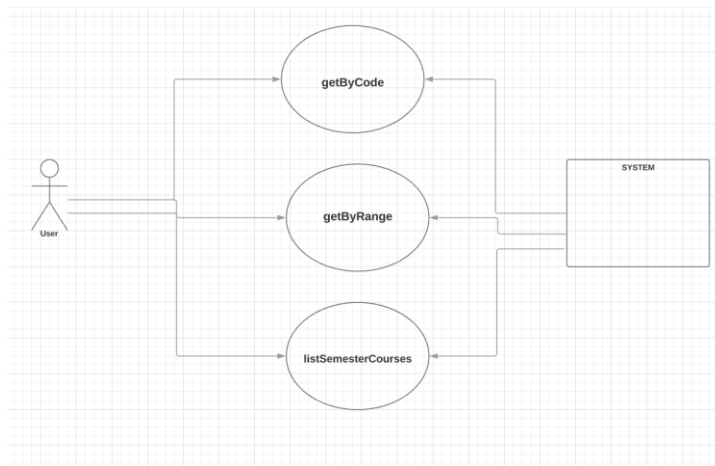
2 METHOD

2.1 Class Diagrams

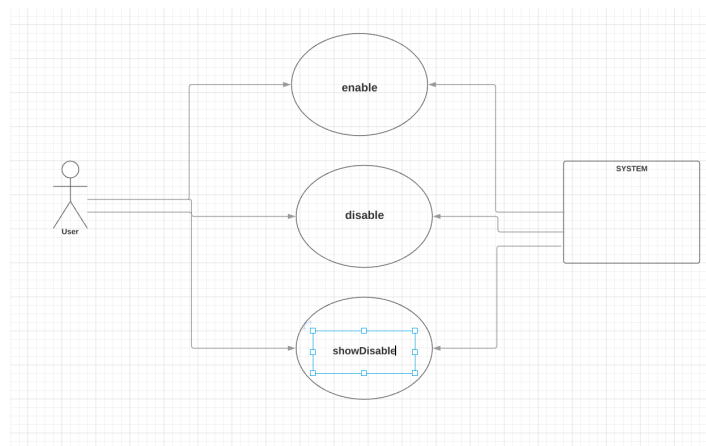


2.2 Use Case Diagrams

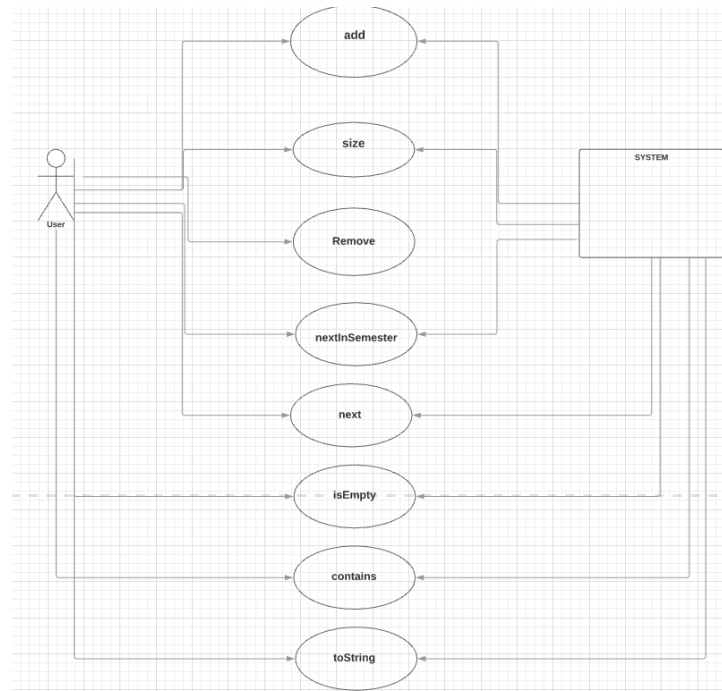
PART1



PART 2



PART 3



2.3 Problem Solution Approach

PART 1

Ders bilgilerini tutmak için GTUCourseStructure sınıfında şu değişkenleri tuttum.

```
private String semester, courseCode;  
private String courseTitle, ECTSCredits;  
private String GTUCredits, hoursTheoryLaboratory;
```

Bu değişkenlerin public olarak getter ve setter methodlarını tanımladım. Part1 için yazdığım sınıfta GTUCourseStructure tipinde gtuCourses LinkedList yapısını tanımladım. Bu partta karşılaştığım problemleri çözmek için neler yaptığımı methodlar üzerinden anlatacağım.

private final LinkedList<GTUCourseStructure> gtuCourses;

Ders bilgileri bulunan .csv dosyasını parse edip gtuCourses yapısına yerleştirmek için **readGTUCoursesCSVFile()** methodu kullandım.

- **private boolean readGTUCoursesCSVFile()** : Bu method dosyadan bilgileri satır satır okuyarak **split methodu** ile parse eder ve **gtuCourses yapısına** yerleştirir. Dosyanın başarılı şekilde parse edilmesi sonucu true değerini döndürür.
- **private int countFileRows(String fileName)** : Bu method parametreden dosyanın adını alır. Dosyayı satır satır okuyarak kaç satırdan oluştuğunu bulur ve return eder.
- **private int setGTUCourseStructure(String[] courseStructure, int index)** : Bu method dosyadan okunan satırın parse edildikten sonra parça parça **GTUCourseStructure[] gtuCourseStructure** dizine yerleştirilmesini sağlar. İşlem başarılı ise 1 return eder.
- **public String[] getByCode(String courseCode) throws Exception** : Bu method parametreden String olarak aldığı ders koduna göre bu koda karşılık gelen ders isimleri belirler. Belirlediği ders isimlerini String arrayi olarak return eder. Eğer ders kodu sistemde bulunmuyorsa **exception** fırlatır.
- **public String[] listSemesterCourses(int semester) throws Exception:** Bu method parametreden aldığı semester yılına göre o dönem olan tüm derslerin isimlerini belirler ve String arrayi olarak return eder. Parametreden gelen semester değeri sistemde bulunmuyorsa exception fırlatılır.
- **public String[] getByRange(int start_index, int last_index) throws Exception** : Bu method belirtilen başlangıç ve bitiş noktaları arasındaki dersleri belirler. Arama yapılırken bitiş noktası hesaba ([3, 9) etc.) katılmaz. Belirlenen dersler String arrayi olarak return edilir. Belirtilen indeksler hatalı ise exception fırlatılır.

PART 2

Kendi ekleyeceğim özel methodların da çalışabileceği bir LinkedList yapısı oluşturmak için öncelikle sınıfımı LinkedList sınıfından extend ettim. Disable methodunda listeden çıkarılacak olan elemanı daha sonra enable edebilmek için datasını ve bulunduğu konumu tutabileceğim iki tane ArrayList tanımladım.

```
private final ArrayList<T> disableObjects;  
private final ArrayList<Integer> disableIndexs;
```

Disable methodunda ilk önce elemanın olup olmadığı kontrol ettim. Eğer yoksa exception fırlattım. **disableObjects** ArrayList'ine disable edilecek elemanı, **disableIndexs** ArrayList'ine disable edilecek elemanın bulunduğu konum indeksini koyup LinkedList'ten bu elemanı remove işlemiyle uzaklatırdım.

Enable methodunda ise ilk önce **disableObjects** ArrayList'inde enable edilecek elemanın olup olmadığını kontrol ettim. Eğer yoksa exception fırlattım. **disableIndexs** ArrayList'inden elemanın konum bilgisini de alarak LinkedList'ime geri ekledim.

ShowDisabled methodunda ilk önce disable edilmiş olan elemanın olup olmadığını kontrol ettim. Eğer disable edilmiş eleman yoksa hata mesajı bastırdım. Disable edilen elemanları bu method içerisinde **disableObjects** ArrayList'inden okuyarak ekrana bastırdım.

Disable sırasında elemanı LinkedList'den uzaklaştırdığım için size otomatik olarak değişmektedir. Geriye kalan get, set, size, remove and listIterator methodları da bu eleman üzerinde işlem yapamamaktadır.

Disable edilen elemanı enable ederken tekrardan LinkedList' e eklediğim için size yine otomatik olarak değişmektedir. get, set, size, remove and listIterator methodları da bu eleman üzerinde tekrardan işlem yapabilir duruma gelmektedir.

PART 3

Kendi LinkedList'imi oluşturmak için önce Semester Course classı yazıp içinde semester ve ders adı bilgileri tuttum daha sonra inner Node classımı yazarak içinde şu memberları tuttum. CircleNext meberını aynı dönem olan dersler için kullandım.

```
private SemesterCourse data;  
private Node next;  
private Node circleNext;
```

Add methodunda ekleyeceğim eleman için yeni node oluşturarak listeme ekledim. Her add işlemi içinde circle olma durumunu da kontrol etmekteyim.

Remove işlemi yaparken çıkaracağım elemanı aradan çekerek elemandan bir önceki ve bir sonraki elemanları nextleri ile birbirine bağladım. Remove işleminden sonra circle listesinde ekli olup olmadığını da kontrol ettim ama **circle yapısından remove işlemi yaparken her zaman doğru sonuçlar alamıyorum**. Onun dışında doğru çalışıyor.

Method işlemlerim sırasında elemanın listede ekli olup olmadığını bulmak için **contains** methodu yazdım. Eleman varsa true return ettirdim.

İstediğim indeksteki bilgiyi alabilmek için **get** methodu yazdım. Circle listeler için ise **getCircle** methodunu yazarak konumdaki bilgilere ulaştım.

Listenin boş olup olmadığını kontrol etmek için **isEmpty** methodu yazarak size sıfıra eşit ise true return ettirdim.

Bir sonraki elemana ilerlemek için **next**, circle yapıda bir sonraki elemana ilerlemek için **nextInSemester** methodlarını yazdım.

Listeyi ekrana bastırabilmek için **toString** methodunu override ettim.

Özetle bu partta olması istenen aşağıdaki methodları remove methodunun circle yapılar üzerinde bazen yanlış sonuç vermesi dışında diğerlerini tam olarak yaptım.

add(): Add new item to list
remove(): Delete specified item from list
next(): Move next node
nextInSemester(): Move next node in same semester
size(): Get size of the list

LinkedList'e ek olarak eklediğim methodlar şunlardır.

contains():
isEmpty():
get():
getCircle():

Methodlarımı tamamlayarak kendi LinkedList sınıfımı yazmış oldum.