

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 03 REPORT

**EMIRE KORKMAZ
141044043**

MEHMET BURAK KOCA

1 INTRODUCTION

1.1 Problem Definition

LinkedList veri yapısını kullanarak GTU'deki derslerin ve derslere ait bilgilerin tutulması ve üzerinde istenen işlemlerin yapılması.

1.2 System Requirements

NOT: Türkçe karakterler hata verdiği için csv dosyasındaki Türkçe karakterleri değiştirdim.

Part 1 için:

```
public String getByCode (String code) throws IndexOutOfBoundsException
```

Bu fonksiyon verilen ders kodunu alarak class'ın içindeki LinkedList objesinde tuttuğu listedeki ders kodlarıyla eşleşip eşleşmediğine bakıyor. Eğer eşleşme varsa kodu verilen dersin adını return ediyor. Eğer verilen kod geçersizse `IndexOutOfBoundsException` fırlatıyor.

```
public LinkedList<Data> listSemesterCourses (int semester) throws  
IndexOutOfBoundsException
```

Bu fonksiyon verilen dönem bilgisini alarak ders bilgilerini tuttuğu listedeki verilerle karşılaştırıyor. Eğer verilen dönem geçersizse `IndexOutOfBoundsException` fırlatıyor. Eğer geçerliyse verilen dönemdeki tüm dersleri bir LinkedList'e atıyor ve onu return ediyor.

```
public LinkedList<Data> getByRange(int start_index, int last_index) throws  
IndexOutOfBoundsException
```

Bu fonksiyon verilen indexler arasındaki class'ta tuttuğu ders bilgilerini yeni bir LinkedList'e ekliyor. Daha sonra bu listeyi return ediyor. Index'lerin geçersiz olması durumunda `IndexOutOfBoundsException` fırlatıyor.

Part 2 için:

```
public void disable(E course);
```

Bu fonksiyon disable edilecek dersin bulunduğu **Data**(Data tipinde bir değişken oluşturulursa sadece adının verilmesi yeterli olur) tipinde bir değişken alıp o dersin diğer fonksiyonları

kullanmasını engelliyor.

```
public void enable(E course);
```

Bu fonksiyon enable edilecek dersin bulunduğu **Data**(Data tipinde bir değişken oluşturulursa sadece adının verilmesi yeterli olur) tipinde bir değişken alıp o dersin diğer fonksiyonları kullanmasını sağlıyor.

```
public void showDisabled()
```

Bu fonksiyon disable edilmiş dersleri ekrana yazıyor.

```
public int size(E givenData)
```

Bu fonksiyon bir ders adı alıp size bilgisini döndürüyor. Eğer verilen ders disable edilmişse hata mesajı veriyor.

Part 3 için:

```
public void add(E item, int sem);
```

Bu fonksiyon arguman olarak listeye eklenecek dersi ve o dersin dönemini alıyor. Önceki sorularda olduğu gibi ekleme işlemini yapıyor ayrıca dönem dersleri arasında bir circular list oluşturuyor. Bu circular list'i de eklenen her bir dersin bulunduğu Node'daki **Node** tipindeki **prev** ve **circularNext** değişkenlerine gerekli atamaları yaparak oluşturuyor.

```
public void addIndex(E item, int index, int sem)
```

Bu fonksiyon aynı addItem fonksiyonu gibi hem ekleme yapıp hem de eklenen derslerin dönemlerine göre circular bir list oluşturuyor.

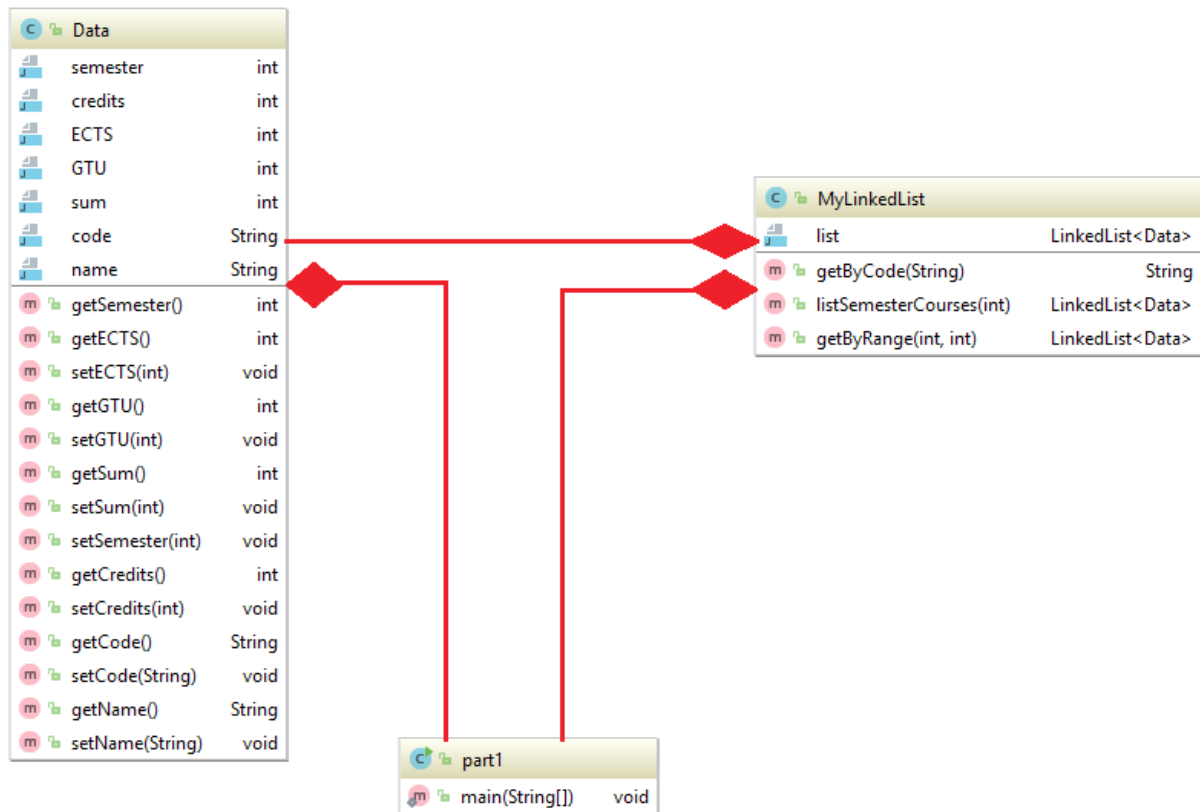
```
public void circularPrint(int s);
```

Bu fonksiyon aynı dönemdeki circular list şeklindeki dersleri ekrana basıyor.

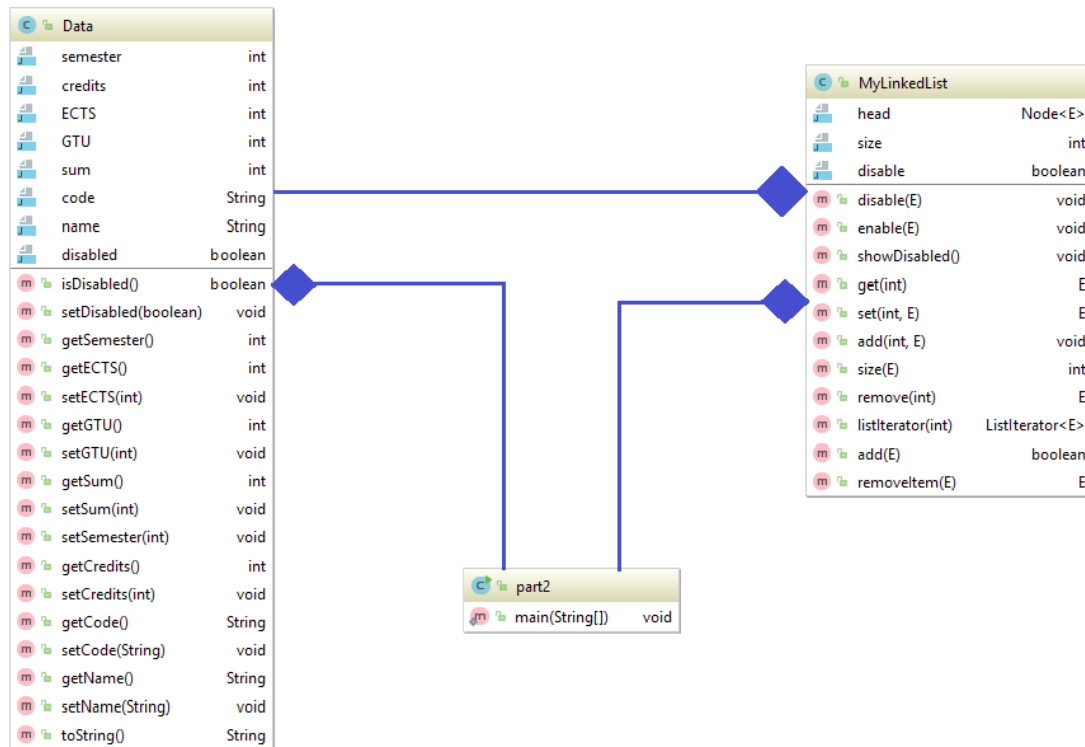
2 METHOD

2.1 Class Diagrams

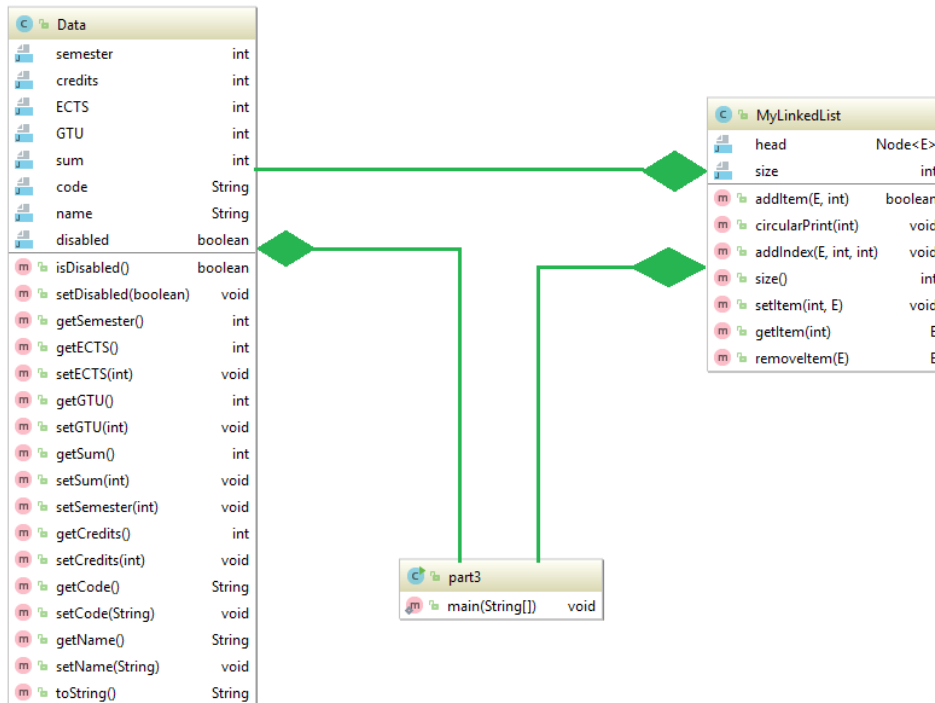
Part 1:



Part 2:



Part 3:



2.2 Use Case Diagrams

Add use case diagrams if required.

2.3 Other Diagrams (optional)

Add other diagrams if required.

2.4 Problem Solution Approach

Part 1:

Derslerin verilerini tutmak için **Data** adında bir class tanımladım. Derslerle alakalı tüm bilgileri (adı, kodu, kredisi vs.) bu class'ı kullanarak tuttum. **MyLinkedList** isimli bir class oluşturup **Data** tipinde bir LinkedList objesi oluşturup. Okuğum ders verilerini bu listede tutup işlem yaptım. **getByCode** fonksiyonu dersin kodunu alıp listedeki derslerle karşılaştırıp dersi bulup adını return ediyor. **ListSemesterCourses** arguman olarak int semester değeri alıp o dönemdeki bütün dersleri **Data** tipindeki bir LinkedList'e kaydedip return ediyor. Son olarak **getByRange** fonksiyonu başlangıç ve bitiş indexleri olmak üzere iki index alıp listedeki verilen indexler arasında bulunan dersleri Data tipindeki bir LinkedList'e atıp return ediyor.

Part 2:

Derslerin verilerini tutmak için **Data** adında bir class tanımladım. Derslerle alakalı tüm bilgileri (adı, kodu, kredisi vs.) bu class'ı kullanarak tuttum. **MyLinkedList** adında bir class oluşturup LinkedList'i extend ettim. LinkedList'teki bazı methodları override ettim. Override edilen fonksiyonların yanı sıra **Node** adında bir inner class oluşturdum. Her bir node'un data ve next bilgilerini ve onların getter ve setter'ları tuttum. Ayrıca **disable** adında bir dersin disable edilip edilmediğini anlamak için boolean bir değişken tuttum. Her bir ders bir node olduğu için eğer o ders disable edilmişse o bilginin de node'daki diğer bilgilerle beraber olması gerektiğini düşündüm. Diğer fonksiyonlar çağırılırken öncelikle disable edilip edilmediği kontrol ediliyor. Eğer disable edilmişse bir hata mesajı verilip fonksiyondan çıkılıyor. Eğer disable edilmemişse istenen işlemleri yapıyor. **Enable** fonksiyonu çağırıldığında parametrede verilen disable edilmiş ders enable ediliyor. **ShowDisabled** fonksiyonu disabled edilmiş dersleri ekrana yazdırıyor.

Part 3:

Bu bölümde diğer sorularda da olduğu gibi ders bilgilerini tutmak için **Data** isimli bir class oluşturdum. Dosyadan okuduğum ders bilgilerini bu yapı aracılığıyla yine kendim oluşturup LinkedList'ten extend ettiğim **MyLinkedList** isimli class türünden bir değişkenin içine yazdım. Diğer sorulardan farklı olarak bu class'ta inner class'ımız olan **Node** class'ında ayrıca **prev** ve **circularNext** isimli **Node<E>** tipinde iki değişken tuttum. Bu değişkenleri aynı dönemdeki derslerin bir arada bir circular list oluşturabilmesi için kullandım. Ekleme yaparken yani add methodunu kullanırken ayrıca bu değişkenleri de set ettim. **add** methoduna bir parametre daha

ekleyip semester bilgisini orada aldım. Test etmek için de **circularPrint** isimli bir fonksiyon yazdım. Bu fonksiyon arguman olarak dönem bilgisini alıp o dönemdeki dersleri ekrana basıyor.

3 RESULT

3.1 Test Cases

Part 1:

Unit Test 'te ilk olarak **getByCode** fonksiyonunu test ettim test için ders kodu olarak yorum olarak belirttiğim satırda bulunan **code** değişkenine “**CSE 241**”i atayarak test ettim. Sonra **listSemester** methodunu yine yorum yazarak belirttiğim satırdaki **semester** değişkenini **3** yaparak test ettim. En sonda da **getByRange** fonksiyonunu yine yorum yazdığım satırda bulunan **start_index** ve **last_index** değişkenlerine 3 ve 5 atayarak test ettim. Yorum yazarak belirttiğim değişkenler değiştirilerek farklı değerlerle de test edilebilir.

Part 2:

Fonksiyonlarımın çoğu **void** olduğu için testlerimi **main**'de yaptım. Fonksiyonları test etmek için ders bilgilerini tuttuğum **Data** tipinde 3 tane ders tanımladım. Bu üç dersten iki tanesini **myLinkedList** üzerinden **disable** fonksiyonunu çağırarak disable ettim. **showDisabled** methodunu çağırarak bu dersleri ekrana yazdırarak disable edildiklerinden emin olduktan sonra disable olunca erişemeyecekleri fonksiyonları çağırdım. Kendi örneğimde **remove** fonksiyonunu ve **get** fonksiyonunu gerekli parametreleri vererek çağırdım. Disable edildikleri için ekrana hata mesajı bastılar. Daha sonra disable ettiğim iki dersten bir tanesini enable ettim ve enable ettiğim dersin artık disable olmadığını göstermek için **showDisabled** methodunu bir kere daha çağırdım ardından enable olunca erişmesinde sorun olmayacak olan **remove** fonksiyonunu çağırdım. Daha fazla dersle ve diğer methodlarla da denenebilir.

Part 3:

Bu bölümde aynı dönem içindeki derslerden bir circular list oluşturmamız istendi. Dersleri okurken yani **add** methodunu çağırırken aynı zamanda circular olarak bir list oluşturmuştum. Test etmek için **circularPrint** adında **semester** değerini arguman olarak alan bir fonksiyon yazmıştım. Main'de bu fonksiyonu çağırarak test ettim. Benim testimde ben semester değerine 3 verdim ve istenen şekilde o dönemdeki dersleri ekrana bastı. Farklı semester değerleriyle de test edilebilir.

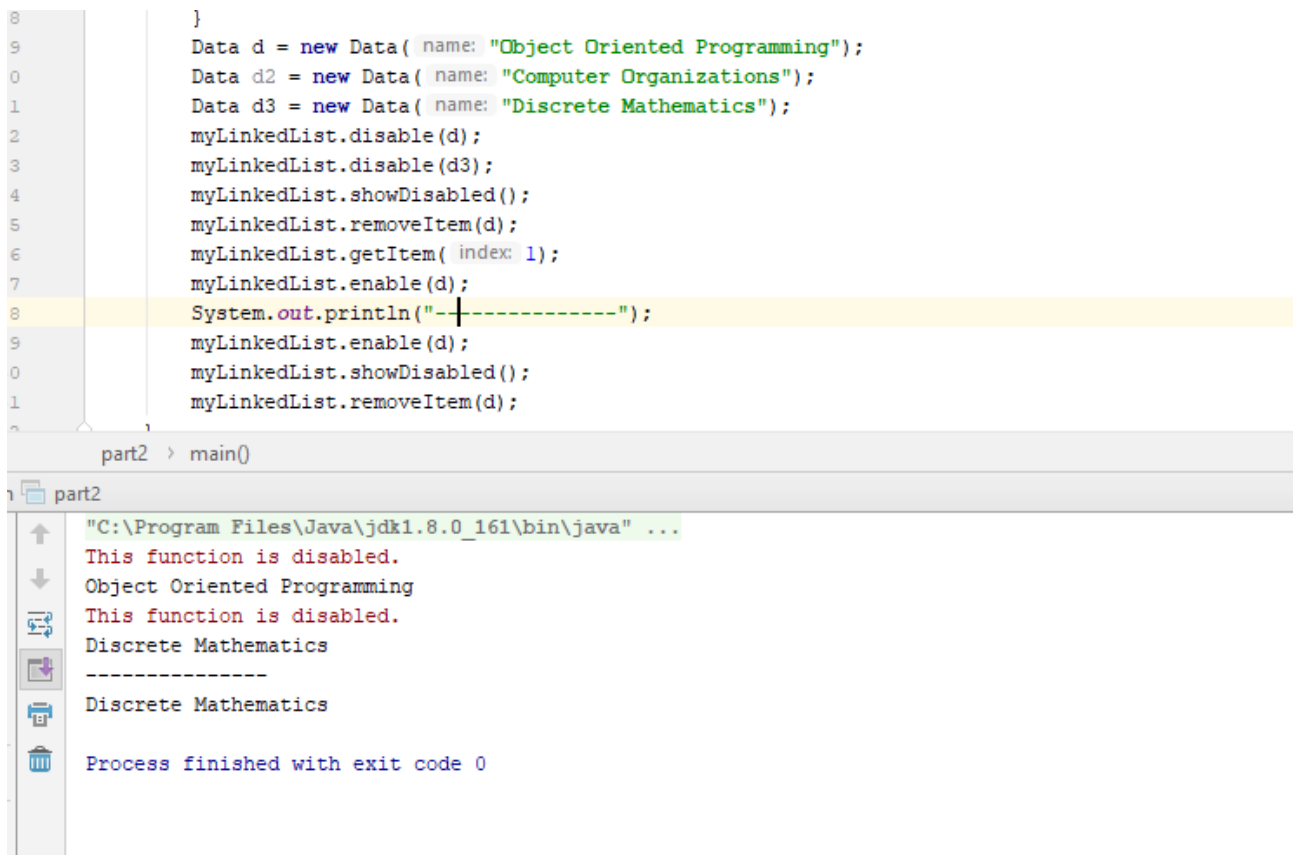
3.2 Running Results

Give running information and results of your solution. It should contain screenshots or any output.

Part 1:



Part 2:



* Hata mesajı verdiği için önce basmış.

Part 3:

```
29         plusSign = plusSign.replaceAll( regex: "\\+", replacement: "+");
30
31         temp = cols[5].split( regex: "\\+");
32         credits =Integer.parseInt(temp[0])+Integer.parseInt(temp[1])+Integer.parseInt(temp[2]);
33
34         data.setCredits(credits);
35         myLinkedList.addItem(data, data.getSemester());
36     }
37     i++;
38 }
39 myLinkedList.circularPrint( s: 3);
40 }
41
```

part3 > main()

part3

↑ "C:\Program Files\Java\jdk1.8.0_161\bin\java" ...

↓ Object Oriented Programming

↓ Discrete Mathematics

↓ Circuits And Electronics

↓ Circuits And Electronics Laboratory

↓ Teknik Olmayan Secmeli (SSB)

↓ Process finished with exit code 0