

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 03 REPORT

Celal TEMİZ

101044070

Course Assistant : Mehmet Burak Koca

1 INTRODUCTION

1.1 Problem Definition

Part 1 – Problem Java LinkedList sınıfı kullanılarak GTU Bilgisayar Mühendisliği ders yapısının tasarlanması ve implement edilmesidir. Bu yapınının oluşturulduktan sonra bu yapı üzerinde çalışacak 3 farklı methodun da implement edilmesi istenmektedir. Fakat bu methodlar implement edilirken Java LinkedList sınıfı extend edilmeyecektir. Bu methodlar :

getByCode(String code) : Verilen ders koduna karşılık gelen dersi döndürmektedir.

listSemesterCourses(int semester) : Verilen semester değerine karşılık gelen o dönemki dersleri döndürmektedir.

getByRange(int start_index, int last_index) : Verilen başlangıç ve bitiş index değerleri aralığındaki dersleri döndürmektedir.

Part 2 – Problem Java LinkedList sınıfının extend edilerek Generic Type' lar üzerinde de çalışabilen bir sınıf yapısının tasarlanması ve implement edilmesidir. Tasarlanan bu yapı üzerinde 3 farklı methodum implement edilmesi istenmektedir. Bu methodlar :

disable() : Verilen değere karşılık liste üzerindeki elemanın erişilemez hale getirilmesidir. Bu durumda erişilemeyen liste elemanı için get(), set(), size(), remove() ve listIterator() methodları engellenmesi istenmektedir.

enable() : Engellenen liste elemanının yeniden erişilebilir ve üzerinde liste methodlarının çalışır hale getirilmesi istenmektedir.

ShowDisabled() : Erişimi engellenen tüm liste elemanlarının listelenmesidir.

Problem genel olarak erişimi engellenen bir liste elemanını, tekrar erişilebilir hale getirmek ve önceki konumuna yerleştirilmesinin sağlanmasıdır.

1.2 System Requirements

Geliştirme Ortamı :

intelliJ version = 2017.3.4, jdk = 8 – Virtual Machine - Debian OS

Ders Verileri :

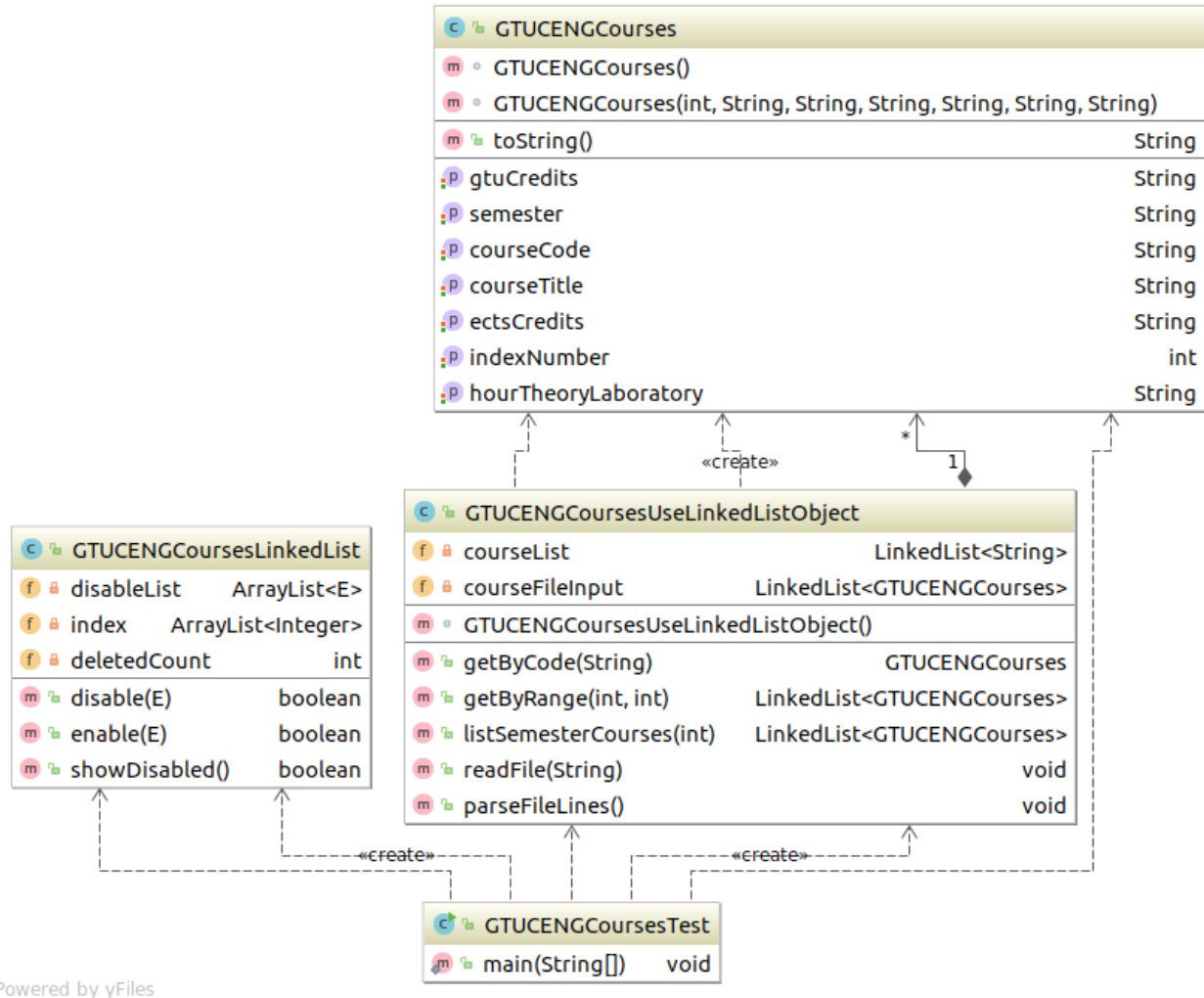
Paylaşılan dosya yapısında bazı Türkçe karakterler görülmeyip, ekrana farklı karakterler basmaktaydı. Bu Türkçe karakterler İngilizce karakterler olarak düzeltilmiştir.

GTU Bilgisayar Mühendisliği derslerine ait tüm bilgileri içeren **Courses(CSV).csv** dosyası. Dosya içeriği aşağıdaki ekran görüntüsünde verilmektedir.

Program kullanıcıdan herhangi bir input değeri almamaktadır. İstendiği gibi kullanıcıdan elle tek tek herhangi bir parametre almadan, test içerisinde verilen parametrelerle oluşturulmuş bir yapıdan oluşmaktadır.

2 METHOD

2.1 Class Diagrams



GTUCENGCourses : Bu sınıf ders bilgilerine ait verileri tutan bir sınıftır.

GTUCENGCoursesUseLinkedListObject : Bu sınıf part - 1' de bahsedilen problem için implement edilmiştir. Java LinkedList sınıfı extend edilmeyip, objesi kullanılmaktadır. Verileri dosyadan okuma, getByCode(), listSemesterCourses() ve getByRange() methodları burada implement edilmiştir.

GTUCENGCoursesLinkedList : Bu sınıf part – 2' de bahsedilen problem için implement edilmiştir. Java LinkedList sınıfı extend edilmiş ve generic type'lar için enable(), disable() ve showDisabled() methodlarını implement etmektedir.

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

Problemin çözümü için öncelikle genel bir **GTUCENGCourses** sınıfı oluşturulmuştur. Bu sınıfın değişkenleri ders bilgilerine ait index bilgisi, semester, course code, course title, ects credits, gtu credits, hours+theory+laboratory' dir.

Part – 1 problemine ait methodları implement etmek için **GTUCENGCoursesUseLinkedObject** sınıfını implement edilmiştir. Bu sınıf içerisinde readFile() methoduyla dosyadan okunan satırlar courseList adlı bir LinkedList'e kaydedilmektedir. Daha sonra bu satırlar parseFileLines() methodu ile GTUCENGCourses objelerini oluşturularak courseList adında LinkedList'e kaydedilmektedir. Bu liste üzerinde istenen methodlar çalışmaktadır. Methodlar için ilgili dersin kayıtlarda bulunup bulunmam durumu kontrol edilip, exception'lar fırlatılmıştır. LinkedList sınıfının objeleri bu kayıtların tutulması için kullanılmıştır.

Part – 2 problemine ait yapı için **GTUCENGCoursesLinkedList** sınıfı LinkedList sınıfını extend ederek generic bir yapı oluşturulmuştur. Disable() methodu verilen liste elemanına erişimi engellemektedir. Bu yapıyı sağlamak için verilen liste elemanı, listeden silinmektedir ve silinen bu eleman başka bir listeye (disableList) eklenmektedir. Bunun hemen ardından silinen elemanın konum bilgisiyle silinen elemanın sayısına bakarak liste üzerinde hareket edilip başka bir listeye (index) eklenmektedir. ShowDisabled() methodu implement edilirken, silinen liste elemanlarını tutan bu listenin elemanları bastırılmaktadır.

Enable() işlemi yapılırken silinen elemanın yukarıdaki bilgilerinden faydalananarak eski listede önceki konumlarına yerleştirme yapılmaktadır. Fakat bu çözüm karmaşık, tekrar eden birçok elemanı içeren test durumlarında başarılı olmamaktadır. Bu yöntemin daha

başarılı sonuçlar verebilmesi için silinen eleman konumu için , ona ait konum aralığından faydalanmak daha verimli olabilir.

3 RESULT

3.1 Test Cases

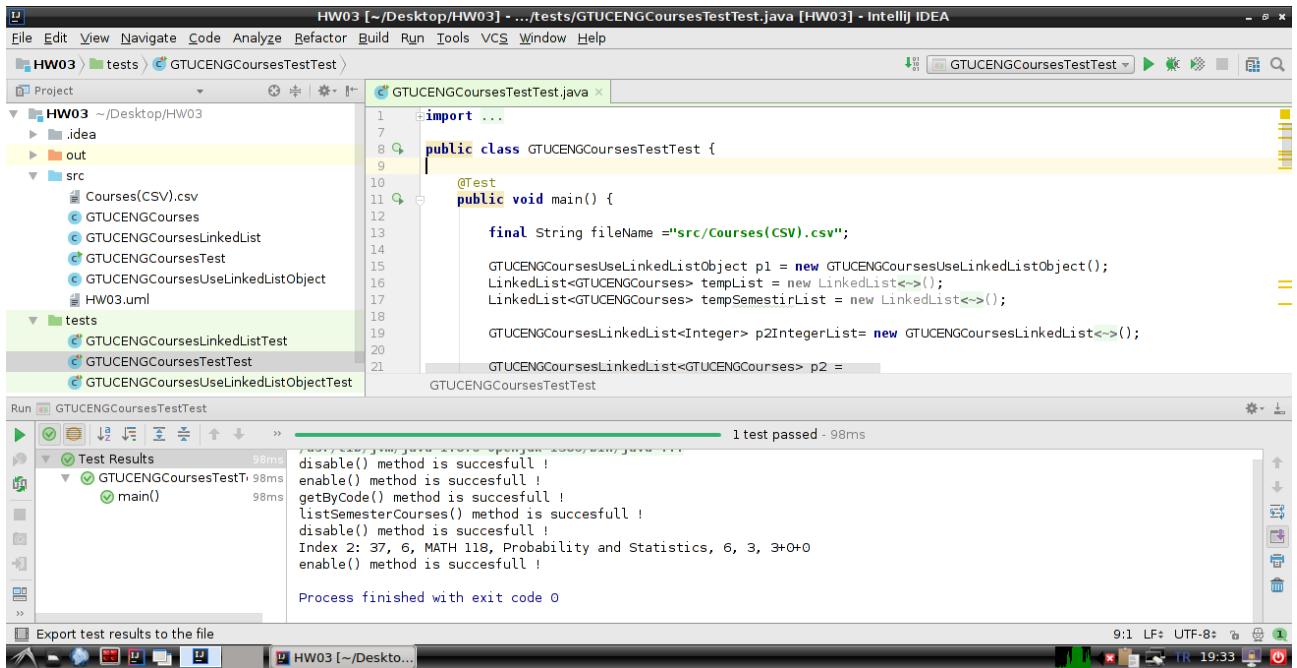
Part 1 ve Part 2 için istenen yapılar ve methodlar main() fonksiyonunda bir senaryo ile test edilmektedir.

Sınıfların testi için yazılan unit testler, proje dosyasının **tests** klasörüne eklenmiştir. Testler çalıştırılabilir durumdadır.

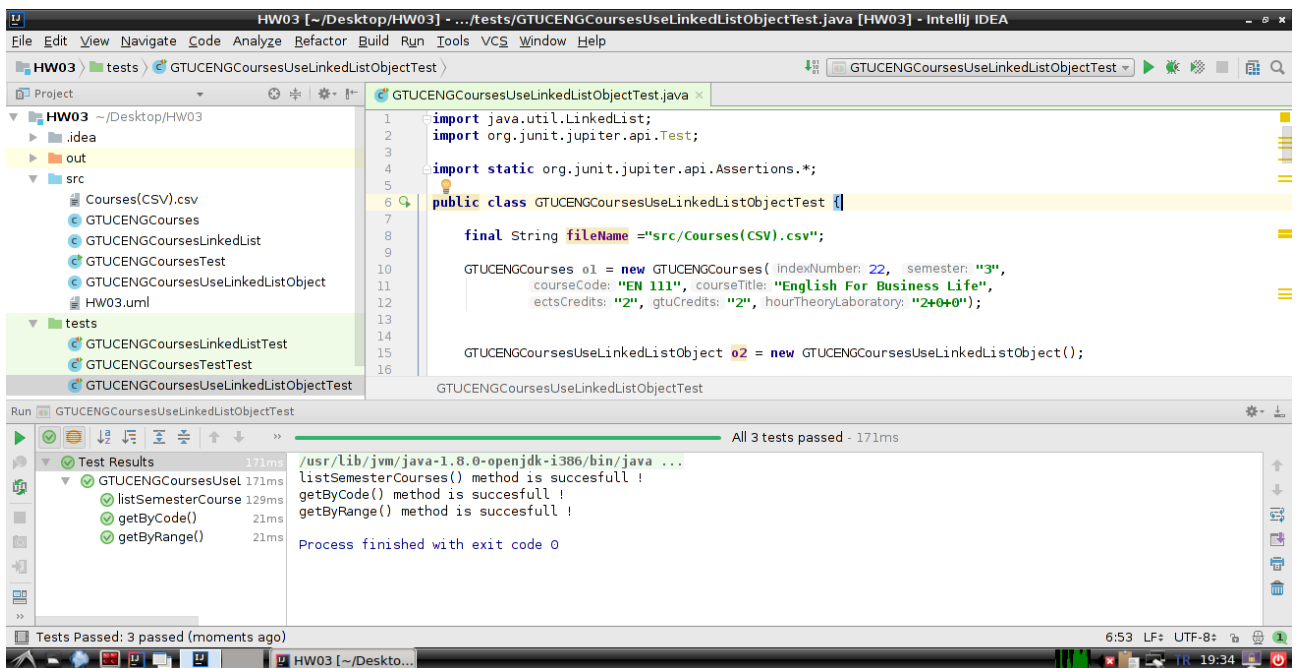
3.2 Running Results

Main Unit Test : Bu test içerisinde Part-1 ve Part – 2’ de istenen methodların testi gerçekleştirilmiştir. Part – 2 methodu generic type’lar için de çalışması istendiğinden öncelikle integer eleman tutan bir liste oluşturuldu ve bu listeye bazı değerler eklendi. Bu işlemden sonra verilen elemanların disable edilmesi, bu değerlerin listelenmesi ve tekrar enable edilmesi için kodlanan methodların doğruluğu test edilmiştir.

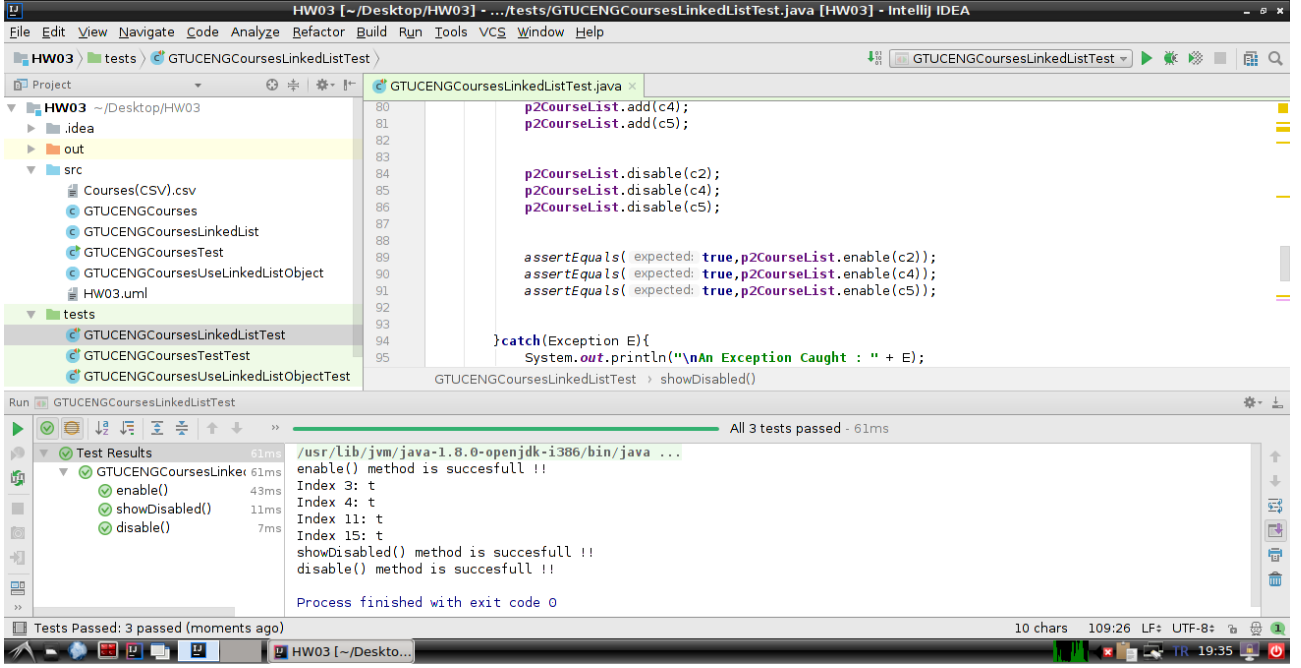
Daha sonra Part – 1 problemine ait getByCode(), listSemesterCourses(), getByRange() methodları test edilmiştir.



Part – 1 Unit Test : Bu test içerisinde örnek obje ve listeler oluşturularak Part – 1’ de yer alan getByCode(), listSemester() ve getByRange() methodlarının testleri gerçekleştirilmiştir.



Part – 2 Unit Test : Bu test içerisinde örnek obje ve listeler oluşturularak Part – 2’ de yer alan enable(), disable() ve showDisabled() methodlarının testleri gerçekleştirilmiştir. disable() methodu test edilirken erişimi engellenmek istenen eleman parametre olarak verilmiş, işlem başarılı olursa, true döndürmesi sağlanmıştır. Aynı şekilde enable() methodu da başarılı olduğun method, true döndürmektedir.



4 Time Complexity

Part – 1 problemi için methodlar `LinkedList` yapısı kullanılmış ve bu yapı üzerinde eleman ekleme yapılmıştır. `LinkedList` yapısına eleman ekleme $O(1)$ zaman almaktadır. Buradaki methodlarda listeye eleman eklenmesi ya da listedeki elemanların gösterimi için liste sonuna kadar bakıldığı için zaman karmaşıklığı $O(n)$ olmaktadır.

Part – 2 problemi için, `disable()` methodu liste boyutu kadar elemana bakıp `add()` ve `remove()` constant time' da işlemler yaptığı için $O(n)$ 'dir.

Diğer methodlarda içerideki döngülerde yapılan işlemler constant time'dır.

Bu nedenle içdöngü elemanları kadar bu işlemler gerçekleştiğinde $O(n)$ olacaktır.

Dıştaki döngü de listedeki eleman sayısı kadar döneceğinden bu methodların zaman karmaşıklığı $O(n^2)$ ' dir.