Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 03 REPORT

DENİZ BABAT
131044013

İçindekiler

Şekil Tablosu

1 PART1

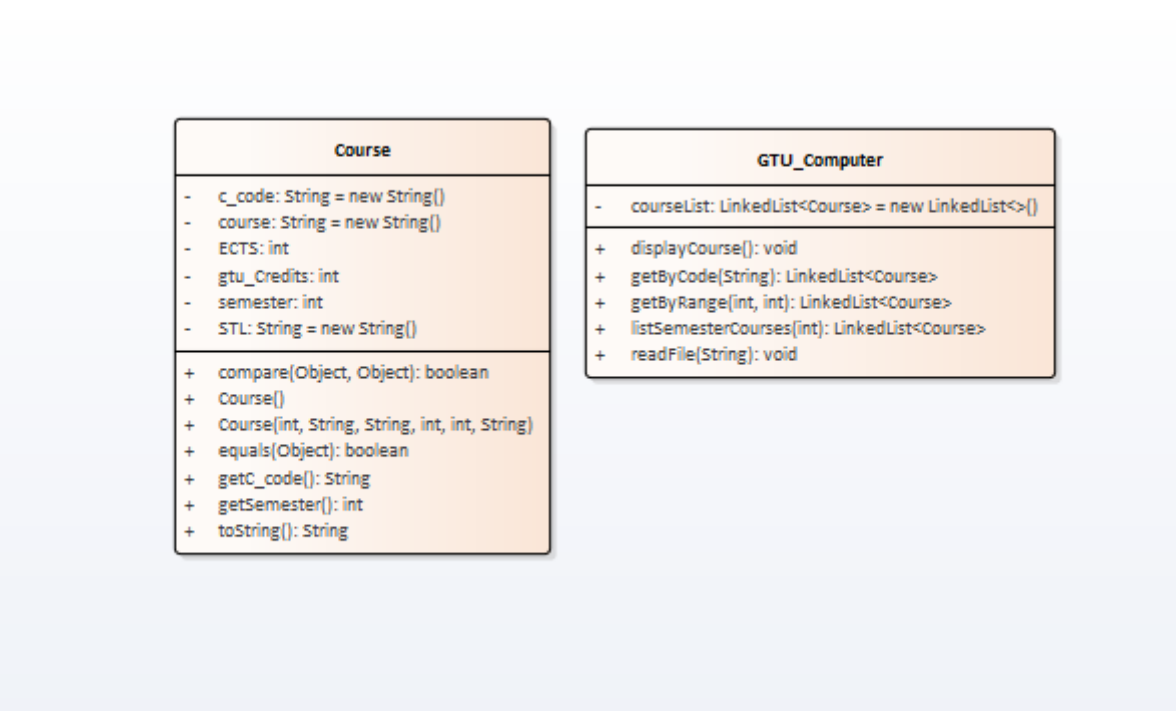
1.1 GİRİŞ

1.1.1 PROBLEM TANIMI:

İlk olarak, bir "Course class" oluşturdum. Bu class sayesinde Extend ettiğim java linkedlist veri yapısında yerleştirdim. Daha sonra tüm datatı csv file'dan okudum ve link list'e doldurdum. Şimdi Funksiyonların hep açıklamasını hemde gereksinimlerini aşağıda bullet biçiminde yazacağım.

- `LinkedList<Course> getByCode(String code): link liste doldurulan, fonksiyona verilen ders kodu parametresi ile listede olan aynı ders konduna sahip dersleri listesini verir. T(n) = Big-O(n)'dir.`
- `LinkedList<Course> listSemesterCourses (int semester): Aynı dönemdeki derslerin listesini verir. T(n) = Big-O(n)'dir.`
- `LinkedList<Course>getByRange(int start_index, int last_index) parametre olarak verilen dönem aralığındaki derslerin listesini verir. T(n) = Big-O(n)'dir.`
- `public void readFile(String filename) throws IOException : dosyadan verileri okur ve linkliste doldurur.`
- `public void displayCourse()listedeki courseları ekrana bastırır.`

1.1.2 CLASS DİAGRAMI:



Şekil 1: Part1 Class Diagramı

Part 1 class diagramı olarak şekil 1 de gözükmektedir. Yukarıda açıklaması yapılmıştır.

1.1.3 TEST CASE

public void getByCode() **throws** Exception : Test case i yapılmış ve **MATH 101** code verilerek istenilen liste alınmıştır. Ekrana print edilmiştir.

public void listSemesterCourses() **throws** Exception: Test case i yapılmış ve "1" semester code verilerek istenilen liste alınmıştır. Ekrana print edilmiştir.

public void getByRange() **throws** Exception Test case i yapılmış ve "1. ve 3." semester code verilerek istenilen liste alınmıştır. Ekrana print edilmiştir.

2 PART 2

2.1 GİRİŞ

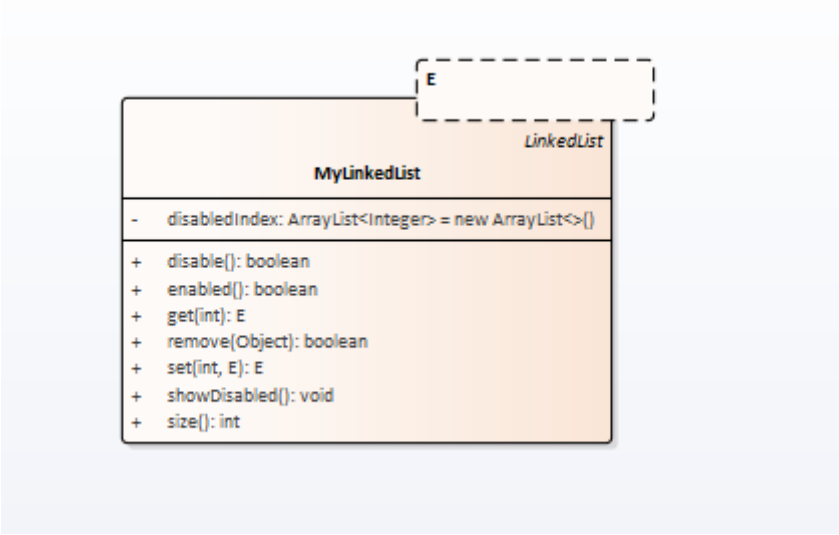
2.1.1 PROBLEMİN TANIMI

Extend edilen link liste eş olarak bir arraylist tutulmaktadır. Bu array list disable olan node'ların indexlerini turmaktadır. Böplece işlemler basit bie şekilde yapılmıştır.

- **public boolean** disable(): Random olarak şeçilen index eğer disable olan arraylist te yoksa ve linkedlist e bir index 'e denk geliyorsa o node disabled olur. Disable olan node varsa true return eder. T(n) = Big-O(n)
- **public boolean** enabled(): disable olanlardan random bir tane seçerek enable yapar. T(n) = Big-O(n)
- **public void** showDisabled(): Disable olan node ları ekrana bastırır.

NOT: Disabled olan nodelar get, set, size, remove and listIterator methodları içinde gösterilmez.

2.1.2 CLASS DİAGRAMI



Şekil 2: Part2 Class diagramı

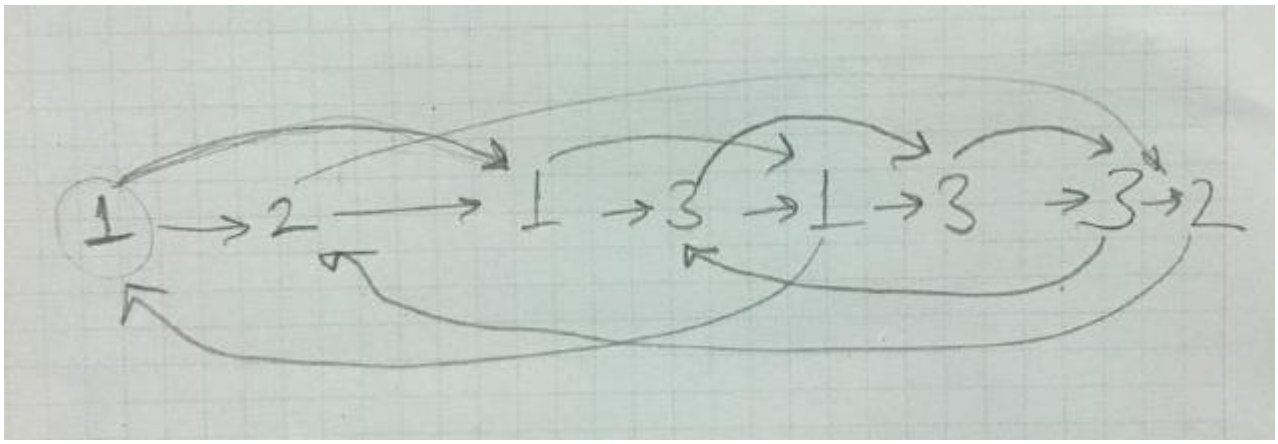
2.1.3 TEST CASE

Testleri yapılmıştır. Test packeage'ında çalıştırabilir ve değişiklik yapabilirsiniz.

3 PART 3

3.1 GİRİŞ

3.1.1 Problem tanımı



Şekil 3: Problem tanımı.

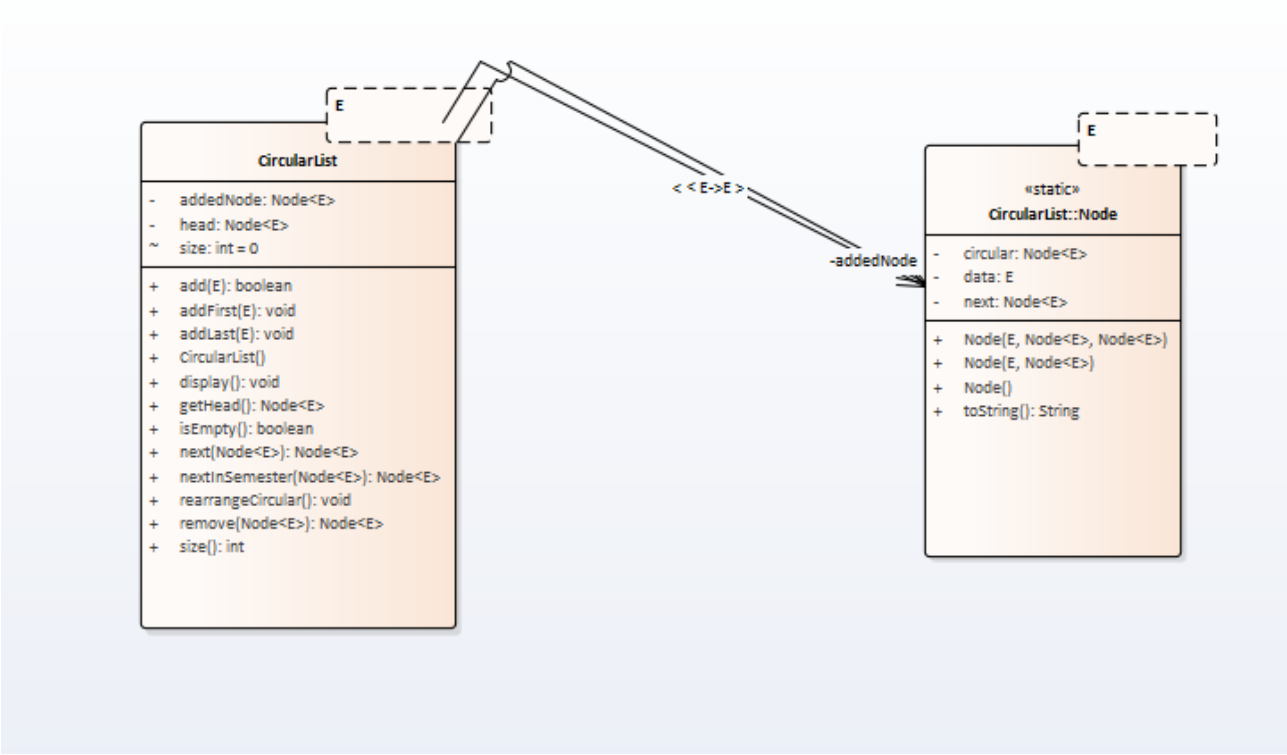
- Static node class'ı oluşturdum ve klasik node classına CircularNext pointer'nı ekledim. Böylece circular yapabilecektim.
- Link list'in başı için head ve en son eklenen ve silinen node için bir yedek tutum. Böylece bir şey eklendiğinde ve silindiğinde hangi circular yapı bozolduğunu bilip onu onarabileceğim.
- courseların listeye eklenmesi ve silinmesi için add ve remove fonksiyonlarını yazdım.
- Daha sonra bunları update edecek rearrangeCircular () fonksiyonunu yazığ tekrar circular yapıyı inşa edecebiliyorum.
- Circuları düzenleyen fonksiyonun T(n) = Big-O(n)'de çalışır.

Foknsiyonlardan sadece add, remove, nextInSemester, next ve rearrangeCircular analatacağım. Diğerleri normal linkedlist fonksiyonları gibi çalışmaktadır.

- Add fonksiyonu normal linklisteki add gibi çalışmaktadır. Sadece en son eklenen node saklanmaktadır ve ekleme sonunda circular yapı tekrar inşa edilmektedir. T(n) = Big-O(n^2) dir.
- Remove fonksiyonu normal linklisteki remove gibi çalışmaktadır. Sadece en son silinen node saklanmaktadır ve silme sonunda circular yapı tekrar inşa edilmektedir. T(n) = Big-O(n^2) dir
- rearrangeCircular eklenen ve silenen dataya bakarak, hangi circular'ın bozulduğuna karar verir ve yapıyı sadece bozulan için inşa eder T(n) = Big-O(n) dir.

- nextInSemester sonraki kendi semesterinde olan node'u getirir. Bunu yaparkende "return temp.circular;" kullanarak yollar. T(n) = Big-O(n) dir.
- Next fonksiyonu önce linklistte kendi nodeunu bulur ve sonrakini return eder. T(n) = Big-O(n) dir.
- Size fonk. Constatnt zamanda çalışır.

3.1.2 CLASS DİAGRAMI



Şekil 4: Part4 class diagramı

Part 4 için class diagramı yukarıda verilmiştir.

3.1.3 TEST CASE

Testleri yapılmıştır. Test packeage'ında çalıştırabilir ve değişiklik yapabilirsiniz.