

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 3 REPORT

**AKIN ÇAM
151044007**

Course Assistant:Mehmet Burak KOCA

1 INTRODUCTION

1.1 Problem Definition

Part1 → Dosyadan alınan ders bilgileri ile bir öğretim programının oluşturulması. Bu program için linkedlist kullanılması ve istenilen dönemlerin derslerinin gösterilmesi, verilen indexler arasındaki derslerin gösterilmesi ve kullanıcıdan alınan kod ile eşleşen derslerin gösterilmesi gibi metodların oluşturulması

Part2 → Bir listede bulunan elemanlara erişim izninin verilmesi ya da engellenmesini sağlamak. Erişim izni verilmediğinde listedeki ilgili elemanlar üzerinde LinkedList yapısının metodlarının kullanılmasını engellemek ve bu erişim izni verilmeyen elemanları listelemek. LinkedList ten extend edilmiş ve bu ek özellikler eklenerek oluşturulmuştur.

Part3 → Hiçbir liste yapısı kullanılmadan özgün bir liste yapısının oluşturulması. Bu liste yapısına LinkedList yapısının metodlarının implement edilmesi. Ayrıca part1 için oluşturulan öğretim planı içinde bulunan her dönemin kendi dersleri arasında bir döngü(bağlantı) oluşturulması.

1.2 System Requirements

Part1 → CourseStructureDiagram_part1 adında bir interface bulunur. Bu interface metodları Curriculum_part1 de implement edilmiştir. Curriculum_part1 de veriler linkedlist ile tutulmaktadır. Bu linked list Course_part1 tipindedir. Course_part1 class ı oluşturulmuştur. Bu class bir ders için ders dönemi, kodu , adı, akts, Gtu kredisi, ve ders-uygulama sayısı verilerini tutmaktadır. Parametresiz ve parametrelili constructor bulunmaktadır. Parametresiz olan instance variable ları ilklendirir ve dosya adını önceden bilinen dosya adının atamasını yapar. Parametrelili constructor diğeri farklı olarak String parametresi alır ve öğretim planının bulunduğu dosya adını alır ve instance variable ları ilklendirir.

Public int readCurriculum() → Bu method verilen dosyadaki ders bilgilerini okur ve linkedlist e ekler. Eğer dosya ismi yanlış ise Exception fırlatır. Dosya csv dosyasıdır. Dosyadan bir ders için ders dönemi, kodu , adı, akts, Gtu kredisi, ve ders-uygulama sayısı verilerini alır String arrayi oluşturup böler ve verileri bir Course_part1 objesine atar ve bunu linkedliste ekler. Toplam ders sayısını return eder.

Public Course_part1[] listSemester(int) → Bu metod bir integer parametresi alır. Alınan parametreye uygun bir dönem yoksa Exception fırlatılır. Eğer listelenmek istenen doğru ise LinkedList e bakılır o dönemdeki dersler Course_part1[] arrayine eklenir. Eklenenler kullanıcıya gösterilir. Ve Bu array return edilir. Eğer istenen döneme henüz hiç ders

eklenmemişse Exception fırlatılır.

Public Course_part1[] getByRange(int int)→ Bu metod verilen indexler arasındaki dersleri kullanıcıya gösterir ve bu dersi-dersleri return eder. Eğer verilen indexlerden herhangi biri uygun değilse exception fırlatılır. Aksi takdirde o aralıktaki dersler kullanıcıya gösterilir return edilir. Örneğin indexler 3 ile 6 ise 3,4,5 ve 6. Index return edilir.

Public Course_part1[] getByCode(String)→ Bu metod verilen ders koduna bakar eğer eşleşen dersler varsa bunları Course_part1 arrayi ile birlikte return eder ve o dersleri kullanıcıya gösterir. Eğer eşleşen kod yok ise Exception fırlatılır.

Public Boolean addCourse(...)→ Bir ders için gerekli tüm bilgileri alır. Legal olmayan bir durum var ise Exception fırlatılır. Eğer girilen dersin aynısı mevcut ise false return edilir. Ders eklenebiliyorsa eklenir ve true return edilir.

Public void printCourses()→ Öğretim programını ekrana yazdırır.

Part2→ Bu bölümde linkedlist extends edilip yeni özellikler eklenmiştir.

Bir eleman disable edilmek istendiğinde listede olup olmadığına bakılır yoksa false return edilir. Eğer listede ise liste kaydedilir. Disable edilen eleman ve indexi kaydedilir. Bu eleman tekrar enable edilmek istendiğinde disable edildiğinde kaydedilen liste ile şimdiki liste karşılaştırılır ve disable edilen eleman eski komşularının yanına onlar kaldırılmışsa bir yandaki komşularının yanına kısaca eski indexine ve komşularına en uygun yere kaydedilir.

Enable edilen eleman ve indexi disable listten silinir. Enable edildiğinde en uygun indexe yerleştirilmesi için kullanılan listeye normal listeye eleman eklendiğinde o listede eklenir daha geniş bir gözden bakabilmek daha uygun bir indexe koymak için bu yüzden LinkedList add metodları override edilmiştir.

Public boolean disabled(E)→ İlk önce elemanın disable edilip edilmediğine bakar edilmişse exception fırlatır. Edilmemişse edilebiliyor mu diye bakılır edilemiyorsa false return edilir. Disable edilebiliyorsa bu listenin tamamı kaydedilir. Disable elemanı ve indexi ayrı bi yere kaydedilir ve bu eleman linkedlist metodlarında kullanılmaması için listeden çıkarılır.

Public boolean enabled(E)→ elemanın disable edilip edilmediğine bakar edilmişse disable edilmemişse false return eder. Disable edilmişse **setEnabledIndex(int ,E)** methodu çağrılır. Bu metoda listenin kaydedilen en geniş haliyle şimdiki hali karşılaştırılır ve uygun index return edilir. Bu indexe enable edilmek istenen eleman eklenir. Disable index ve eleman verileri silinir. True return edilir.

Public boolean showDisabled()→ Eğer hiç disabled eleman yoksa false return eder ve

bunu mesaj olarak bildirir. Eğer disable verilerinin tutulduğu yerde eleman var ise bunlar ekrana yazılır. True return edilir.

Public int EnableorDisabled(e)→ Verilen eleman disable ise 2 enable ise 1 eleman hiçbir yerde yok ise 3 return edilir.

Public void printItems(String)→ Hangi elemanları ekrana yazdırmak istediğini String ile alır eğer enable ise enable elemanlar disable ise disable elemanlar yanlış girdi ise Inccorret input ekrana yazılır.

Part3→

NewCustomList_part3 classında bir Node inner static node class ı oluşturulmuştur. Static olmasının sebebi outer classı değiştirememesidir. Burada next -> bir sonraki eleman için circular-> ilgili elemanlar arasında döngü oluşturmak için ve verileri tutmak için data(Node) bulunur. NewCustomList_part3 classında head elemanlar için en başı tutar. Add metodlarında hatalı indexe ekleme çıkarma durumlarında exception fırlatılmıştır.

Protected Node<E> connectSemester(int int arr)→ verilen index ile bu index ile ilgili methodu circular ile birbirine bağlar.

CircularCustomList_part3 part1 için üsteki classtan extend edilmiştir.

Bu class her dönemde bulunan toplam ders sayısını dönemler ilerledikçe toplam ders sayısını arraylerde tutmaktadır.Consturucturda part 1 deki class in elemanları get ile alınır ve bu yeni listeye eklenir.

Private void addCourses(Curriculum)→ bu dönemler ilerledikçe artan dersleri toplar.

Public E nextSemester()→ Aynı dönem derslerinde circular oluşturulmak istendiğinde kullanılır. Tüm dersler eklenirken sıralı olarak eklenmiştir. Ve her dönemin son dersi o dönemin ilk dersi ile bağlanmıştır. Hangi dönem isteniyorsa ve o dönemin son dersi ise connectSemester ile bağlanmış olan node o dönemin ilk dersinin indexi ile bulunup çağrılır. Aksi takdirde normal next methodu kullanılır ve ders ayrımı yapmaksızın ilerler. chooseSemester hangi semester ile devam etmek istediğini sorar.

Public E next()→ eğer hiç eleman yoksa size==1 ise veya listenin sonunda ise null return eder aksi takdirde listede bir tane next yapıp üzerinden geçtiği elemanı return eder.

Time Complexity→

Part1→

Public int readCurriculum()→ Best case dosyanın boş ya da yanlış isimli dosya olmasıdır. $\Omega(1)$ olur. Worst case dosyanın sonuna kadar dönmesidir. $O(n)$ olur ve her seferinde işinde k tane işlem yapar. Worst case de k yaklaşık olarak n dir ve de $O(n^2)$ çalışmış olur.

Public Course_part1[] listSemester(int)→ Best case de girdinin yanlış olma durumudur $\Omega(1)$ olur. Worst case de iki fordada sonuna kadar dönmesidir ve $O(n^2)$ olur. Time

complexity $O(n^2)$ olur.

Public Course_part1[] getByRange(int int)→ Best case de girdinin yanlış olma durumudur $\Omega(1)$ olur. Worst case de yukarıdaki ile aynı şekilde iki fordada sonuna kadar dönmesidir ve $O(n^2)$ olur. Time complexity $O(n^2)$ olur.

Public Course_part1[] getByCode(String)→

Best case de girdinin yanlış olma durumudur(yanlış kod) $O(n)$ olur. Worst case de yukarıdaki ile aynı şekilde iki fordada sonuna kadar dönmesidir ve $O(n^2)$ olur. Time complexity $\Theta(n^2)$ olur.

Public void printCourses()→ Toplam ders sayısı kadar işlem yapar yani ders sayısı n ise $\Theta(n)$ dir.

Part2→

Public boolean disabled(E)→ Best case ilk elemanın disabled edilme durumudur $\Omega(1)$ olur. Worst case de eleman disabled değildir. Ve diğer fordada remove add işlemleri yapılır ve $O(n^2)$ olur. Time complexity $O(n^2)$ olur.

Public boolean enabled(E)→ ()→ Toplam disable eleman sayısı kadar işlem yapar yani eleman sayısı n ise $\Theta(n)$ dir.

setEnabledIndex(int ,E)→ Best case for a girmediği durumdur ilk conditiondan $O(1)$ ikinci conditiondan $O(1)$ toplamda $\Theta(1)$ olur diyebiliriz. Worst case de for döngüsüne girer $\Theta(1)+O(n)$ time complexity $O(n)$ olur.

Public boolean showDisabled()→ Best case disable eleman olmaması $\Omega(1)$. Worst case de disableları yazdırması n tane eleman için $O(n)$ toplamda daha açık bir ifade ile $\Theta(n)$ diyebiliriz.

Public int EnableorDisabled(e)→ Gelen elemanı arayacağı için $\Theta(n^2)$ diyebiliriz çünkü iyi ihtimalde $O(n)$ kötü ihtimalde $O(n^2)$ çalışır $O(n^2)$ daha genel ve açık bir ifadedir.

Public void printItems(String)→ Toplam disable eleman sayısı kadar işlem yapar yani eleman sayısı n ise $\Theta(n)$ dir.

Part3→

İmplement edilen add remove get peek metodları Constant time da çalışır. Sadece contains metodu en iyi ihtimalle $\Omega(1)$ en kötü ihtimalle $\Theta(n)$ zaman çalışır. Ve biz contains için $\Theta(n)$ çalışır diyebiliriz.

Protected Node<E> connectSemester(int int arr)→ if veya else de birini sağlayacağı ve bu conditionların içindeki işlemler Constant time olduğu için $\Theta(1)$ diyebiliriz.

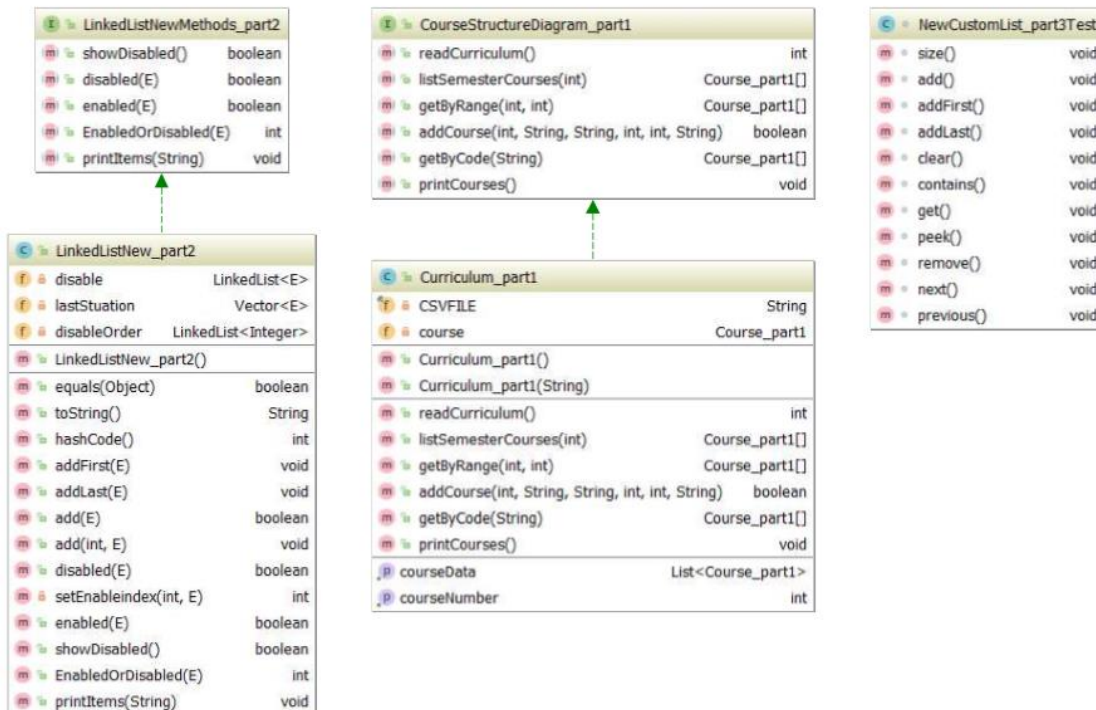
Private void addCourses(Curriculum)→ Semester sayısı kadar çalışacağı için $\theta(n)$ dir diyebiliriz

Public E nextSemester()→ connectSemester(int int arr)→ ı kullanır Constant time olduğu için $\theta(1)$ harcar bunun dışında herhangi pahalı bir işlem de olmadığı ve hepsi constant time de olduğu için $\theta(1)$ çalışır deriz.

Public E next()→ tek condition sağlayacağı için constant timedır ve Constant time olduğu için $\theta(1)$ dir

2 METHOD

2.1 Class Diagrams

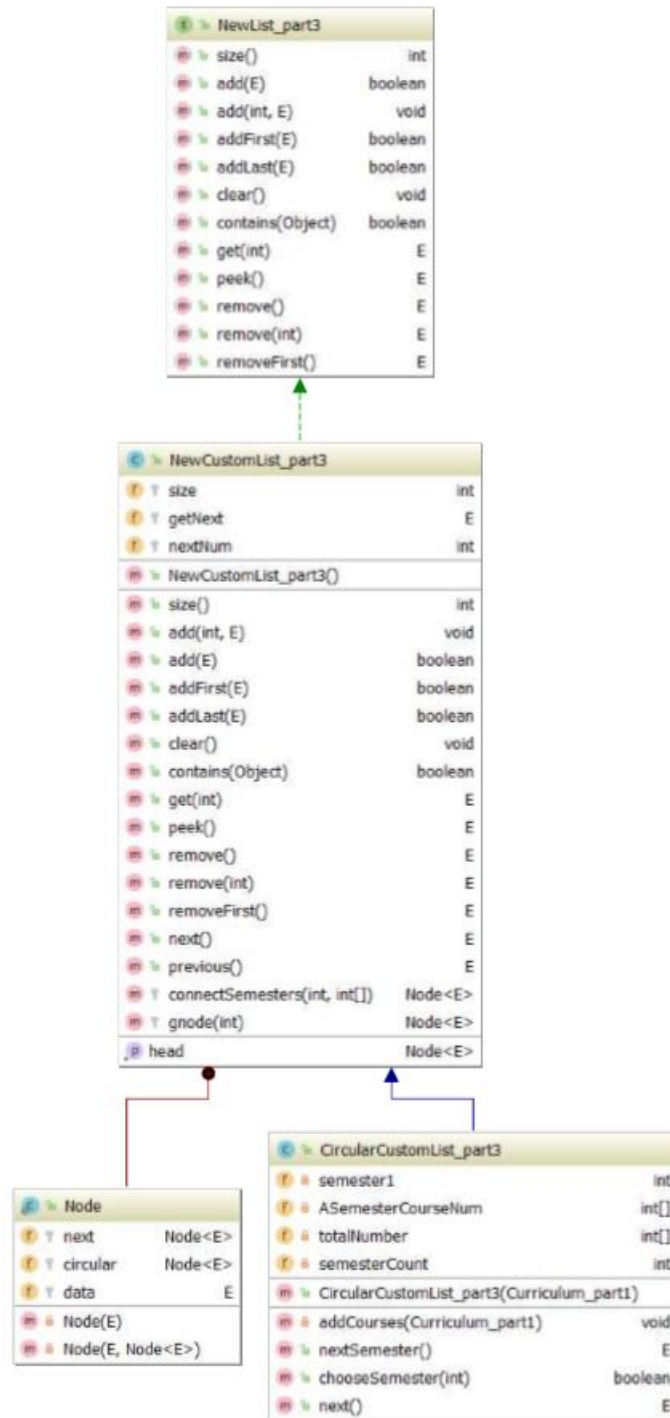


LinkedListNewMethods_part2 class ı bir interfacedir. LinkedList e ek olarak liste elemanlarının kullanımını kısıtlayan,tekrar kullanılabilir hale getiren,verilen bir elemanın

listede olup olmadığını, erişilip erişilmediğini kontrol eden, erişilemeyen elemanları gösteren ve isteğe göre erişilen veya erişilemeyen elemanları ekrana yazdıran methodlar oluşturulmuştur.

Bu interfaceden LinkedListNew_part2 classı oluşturulmuş ve interfacedeki tüm metodların implement edilmesi sağlanmıştır. Aralarında is a relation ilişkisi bulunmaktadır.

CourseStructureDiagram_part1 class ı bir interfacedir. Bir Bölümün dönemlere bölerek ders müfredatını oluşturur. İçinde verileri saklamak için LinkedList tutar. Bu class ta müfredatı dosyadan okumak için bir method, verilen dönemin derslerinin listelenmesi ve kaydedilmesi için bir method, verilen indexler arasındaki derslerin kaydedilmesi ve gösterilmesi için bir method, girilen ders kodunun müfredat içinde olup olmadığına bakan ve eşleşen koddaki dersleri gösteren bir method ve müfredatı ekrana yazdıran bir metod bulunmaktadır. Curriculum_part1 class ı ile CourseStructureDiagram_part1 arasında is a relation ilişkisi bulunmaktadır. Bu class interface in bütün metodlarını implement etmiştir. Bu iki class arasında OOP nin temel prensiplerinden biri olan inheritance bulunmaktadır.



Course_part1
Course_part1()
Course_part1(int, String, String, int, String)
toString()
name
gtu_credits
code
semester
hdi
ects

LinkedListNew_part2Test
add()
disabled()
enabled()
showDisabled()
enabledOrDisabled()

Curriculum_part1Test
readCurriculum()
listSemesterCourses()
getByRange()
getByCode()

CircularCustomList_part3Test
nextSemester()
chooseSemester()
next()

NewList_part3 bir interfacedir. Yeni bir list oluşturulmuştur. Bu interface classında linkedList metodlarının çoğu implement edilmek üzere yazılmıştır.Kendisiile arasında is a

relation ilişkisi bulunan ve tüm metodları implement edilmek üzere olan

NewCustomList_part3 static inner Node classı içermektedir. Bu Node classında generic tipte data,bir sonraki node a gitmek için next ve circular list yapısı oluşturmak için Node tipinde circular bulunur.Arasında is a relation ilişkisi bulunan CircularCustomList_part3 class NewCustomList_part3 ten extend edilmiştir. Curriculum(Müfredat) ı tekrar oluşturmak ve dönem dersleri arasında döngü oluşturmak için bu class oluşturulmuştur. Course_part1 metodu part1 ve part3 için kullanılmıştır.

2.2 Problem Solution Approach

İlk önce tüm partlar için olması gereken fonksiyonları ayrı ayrı interface oluşturarak belirledim. Daha sonra constructor ve local variableları oluşturdum. Extends ve implements işlemlerini gerçekleştirdim. Unit ve normal testlerle sonuçları kontrol ettim.

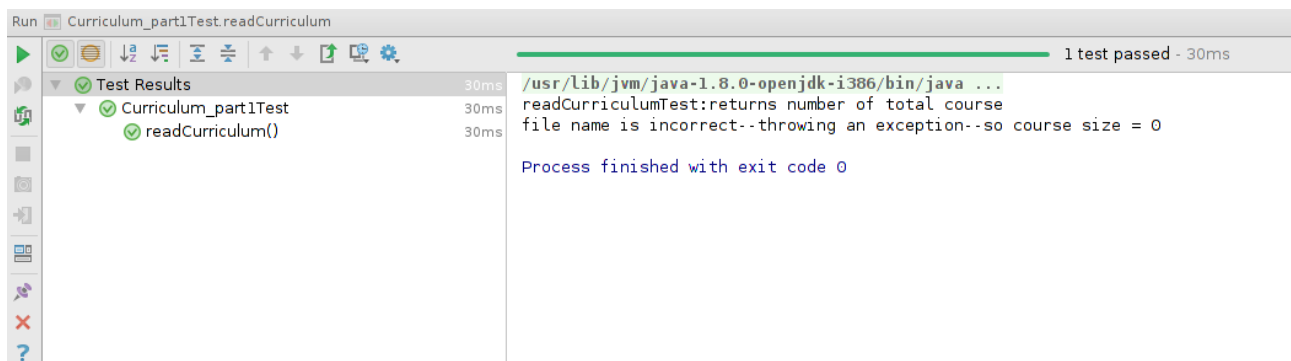
3 RESULT

3.1 Test Cases

Part1→(JUNIT5)

Dersleri dosyadan okuma ve linked liste ekleme:

```
@Test
void readCurriculum() throws IOException {
    Curriculum_part1 cr=new Curriculum_part1();
    Curriculum_part1 cr1;
    int size=0;
    System.out.println("readCurriculumTest:returns number of total course");
    assertEquals( expected: 54,cr.readCurriculum());
    try {
        cr1 = new Curriculum_part1( fileName: "aaaa");
        size=cr1.readCurriculum();
    }
    catch (IOException e){
        System.out.println("file name is incorrect"+"---"+"throwing an exception"+"---"+"so course size = 0");
        assertEquals( expected: 0,size);
    }
}
```



Verilen smesterleri listeleyen metod:

```
@Test
void listSemesterCourses() throws IOException {
    Curriculum_part1 cr=new Curriculum_part1();
    try {
        cr.listSemesterCourses(12);
    }
    catch (Exception e){
        System.out.println("listSemesterCourses() test: input semester num is out of range!! Throws exception. CourseArray cant initialize");
        assertEquals(e.getMessage(), actual: "There is no matching semester");
    }
    System.out.println("listSemesterCourses() test: input semester num is legal! listSemesterCourses. CourseArray size is 8 for semester 1");
    Curriculum_part1 cr1=new Curriculum_part1();
    cr1.listSemesterCourses(1);
    assertEquals( expected: 8,cr1.listSemesterCourses(1).length);
}
```

Curriculum_part1Test.listSemesterCourses

1 test passed - 25ms

Test Results	25ms	25ms	25ms
Curriculum_part1Test	25ms	25ms	25ms
listSemesterCourses()	25ms	25ms	25ms

```
/usr/lib/jvm/java-1.8.0-openjdk-i386/bin/java ...
listSemesterCourses() test: input semester num is out of range!! Throws exception. CourseArray cant initialize
listSemesterCourses() test: input semester num is legal! listSemesterCourses. CourseArray size is 8 for semester :
Course-->{
|
| semester-->1
| code-->XXX XXX
| name-->Teknik Olmayan Se0meli (SSC)
| ects-->2
| gtuCredits-->1
| htl-->2+0+0
|
| }
}

Course-->{
|
| semester-->1
| code-->CSE 101
| name-->Introduction To Computer Engineering
| ects-->8
| gtuCredits-->3
| htl-->3+0+0
|
| }
}

Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
| }
}
```

Test Results	25ms	/
Curriculum_part1Test	25ms	
listSemesterCourses()	25ms	<pre> Course-->{ semester-->1 code-->PHYS 121 name-->Physics I ects-->6 gtuCredits-->4 htl-->3+0+0 } Course-->{ semester-->1 code-->PHYS 151 name-->Physics Laboratory I ects-->1 gtuCredits-->1 htl-->0+0+2 } Course-->{ semester-->1 code-->SSTR 101 name-->Principles Of Atatürk And The History Of Turkish Revolution I ects-->2 gtuCredits-->2 htl-->2+0+0 } Course-->{ semester-->1 code-->TUR 101 name-->Turkish I ects-->2 gtuCredits-->2 htl-->2+0+0 } </pre>

Verilen indexler arasındaki dersleri return edip yazdıran method:

```

@Test
void getByRange() throws IOException {
    Curriculum_part1 cr=new Curriculum_part1();
    System.out.println("getByRange() test: input index num is out of range!! Throws exception. CourseArray cant initialize");
    try {
        cr.getByRange( start_index: -5, last_index: 3);
    }
    catch (Exception e){
        assertEquals( expected: "Wrong input in getByRange!!!!!!",e.getMessage());
    }
    System.out.println("-----test2-----");
    Curriculum_part1 cr1=new Curriculum_part1();
    assertEquals( expected: 4,cr.getByRange( start_index: 0, last_index: 3).length);
    System.out.println("True index");
}

```

Test Results	42ms	/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.0/bin/java ...
Curriculum_part1Test	42ms	getByRange() test: input index num is out of range!! Throws exception. CourseArray cant initialize
getByRange()	42ms	-----test2----- Course-->{ semester-->1 code-->XXX XXX name-->Teknik Olmayan Se0meli (SSC) ects-->2 gtuCredits-->1 htl-->2+0+0 } Course-->{ semester-->1 code-->CSE 101 name-->Introduction To Computer Engineering ects-->8 gtuCredits-->3 htl-->3+0+0 } Course-->{ semester-->1 code-->CSE 107 name-->Introduction To Computer Science Laboratory ects-->2 gtuCredits-->1 htl-->0+0+2 } Course-->{ semester-->1 code-->MATH 101 name-->Calculus I ects-->7 gtuCredits-->5 htl-->5+0+0 }

Verilen kodla eşleşen ders kodlarının bilgilerini yazdırıp return eden method:

```
@Test
void getByCode() throws IOException {
    Curriculum_part1 cr=new Curriculum_part1();
    System.out.println("getByCode() test: input code is no match anything!! Throws exception. CourseArray cant initialize");
    try{
        cr.getByCode("fsdffs");
    }
    catch (Exception e){
        assertEquals( expected: "There is no matching code!!!",e.getMessage());
    }
    System.out.println("-----test2----- (CSE 222) matched-----");
    assertEquals( expected: 1,cr.getByCode("CSE 222").length);
}
}
```

Test Results	29ms	/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.0/bin/java ...
Curriculum_part1Test	29ms	getByCode() test: input code is no match anything!! Throws exception. CourseArray cant initialize
getByCode()	29ms	-----test2----- (CSE 222) matched----- Course-->{ semester-->4 code-->CSE 222 name-->Data Structures And Algorithms ects-->9 gtuCredits-->5 htl-->4+2+0 } Process finished with exit code 0

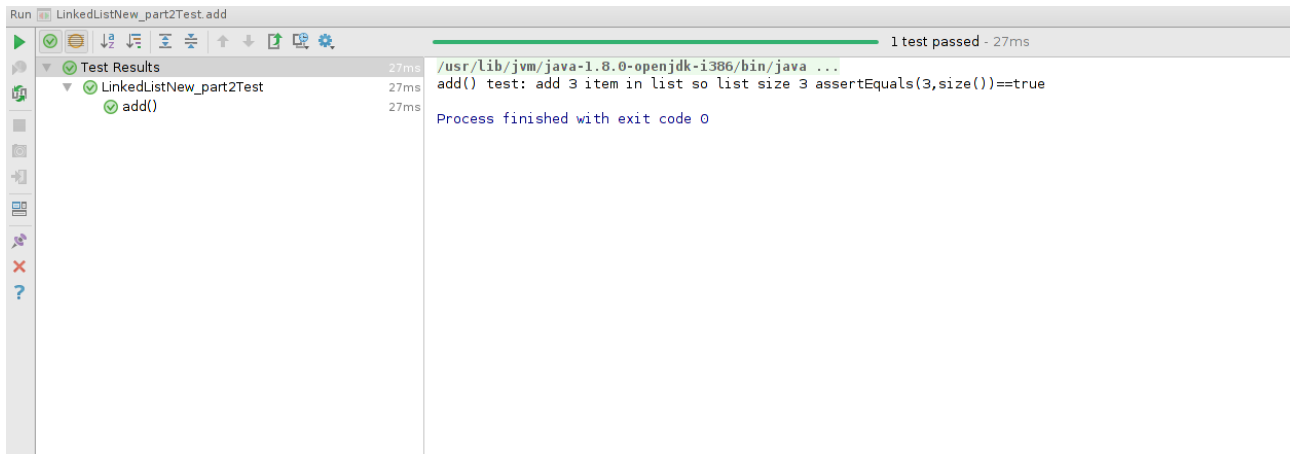
Part2→(JUNIT5)

Eleman ekleme:

```

    void add() {
        System.out.println("add() test: add 3 item in list so list size 3 assertEquals(3,size())==true");
        LinkedListNew_part2<String> ls=new LinkedListNew_part2<>();
        ls.add("akin");
        ls.add("cam");
        ls.add("151044007");
        assertEquals( expected: 3,ls.size());
    }

```

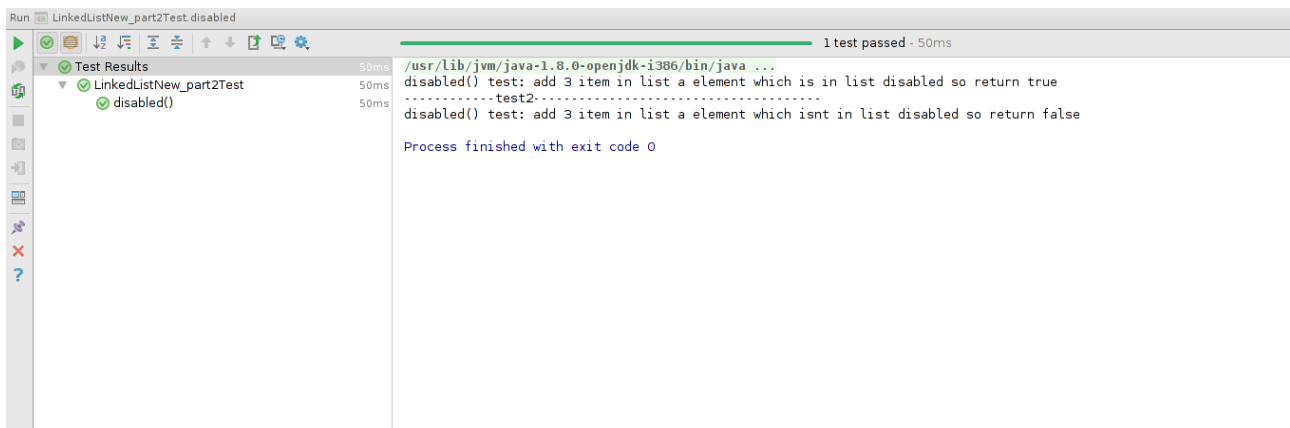


Disable metod:

```

@Test
void disabled() {
    LinkedListNew_part2<String> ls=new LinkedListNew_part2<>();
    ls.add("akin");
    ls.add("cam");
    ls.add("151044007");
    System.out.println("disabled() test: add 3 item in list a element which is in list disabled so return true");
    assertEquals( expected: true,ls.disabled("akin"));
    System.out.println("-----test2-----");
    System.out.println("disabled() test: add 3 item in list a element which isnt in list disabled so return false");
    assertEquals( expected: false,ls.disabled("aaa"));
}

```

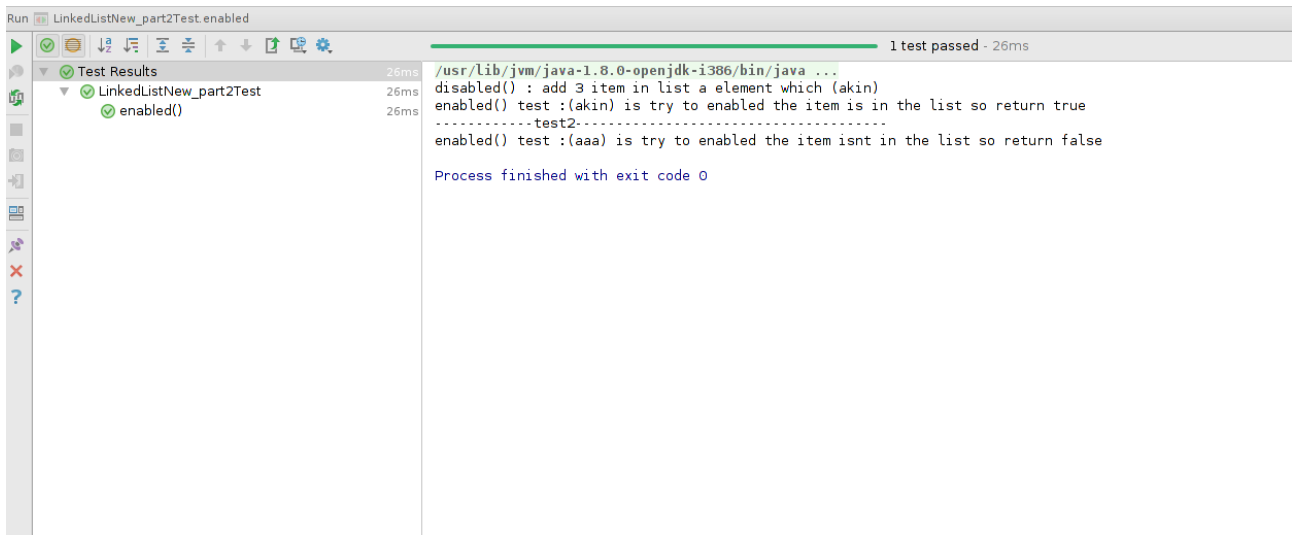


Enable metod:

```

@Test
void enabled() {
    LinkedListNew_part2<String> ls=new LinkedListNew_part2<>();
    ls.add("akin");
    ls.add("cam");
    ls.add("151044007");
    System.out.println("disabled() : add 3 item in list a element which (akin)");
    ls.disabled("akin");
    System.out.println("enabled() test : (akin) is try to enabled the item is in the list so return true");
    assertEquals( expected: true,ls.enabled("akin"));
    System.out.println("-----test2-----");
    System.out.println("enabled() test : (aaa) is try to enabled the item isnt in the list so return false");
    assertEquals( expected: false,ls.enabled("aaa"));
}

```

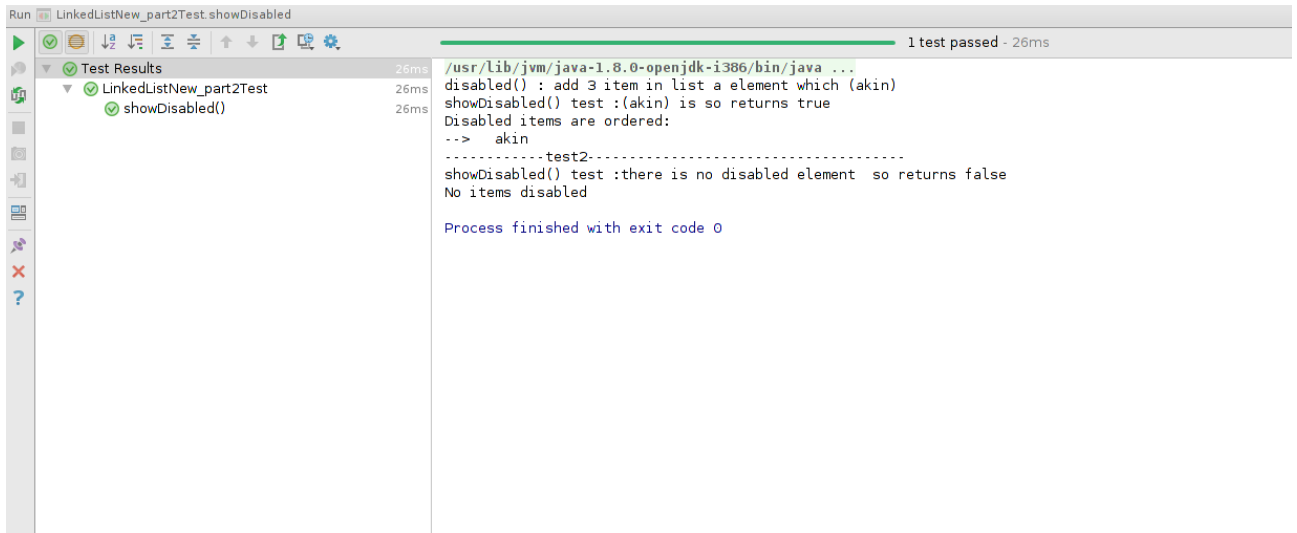


Disable elemanları gösteren method:

```

@Test
void showDisabled() {
    LinkedListNew_part2<String> ls=new LinkedListNew_part2<>();
    ls.add("akin");
    ls.add("cam");
    ls.add("151044007");
    System.out.println("disabled() : add 3 item in list a element which (akin)");
    ls.disabled("akin");
    System.out.println("showDisabled() test :(akin) is so returns true");
    assertEquals( expected: true,ls.showDisabled());
    ls.enabled("akin");
    System.out.println("-----test2-----");
    System.out.println("showDisabled() test :there is no disabled element so returns false");
    assertEquals( expected: false,ls.showDisabled());
}

```

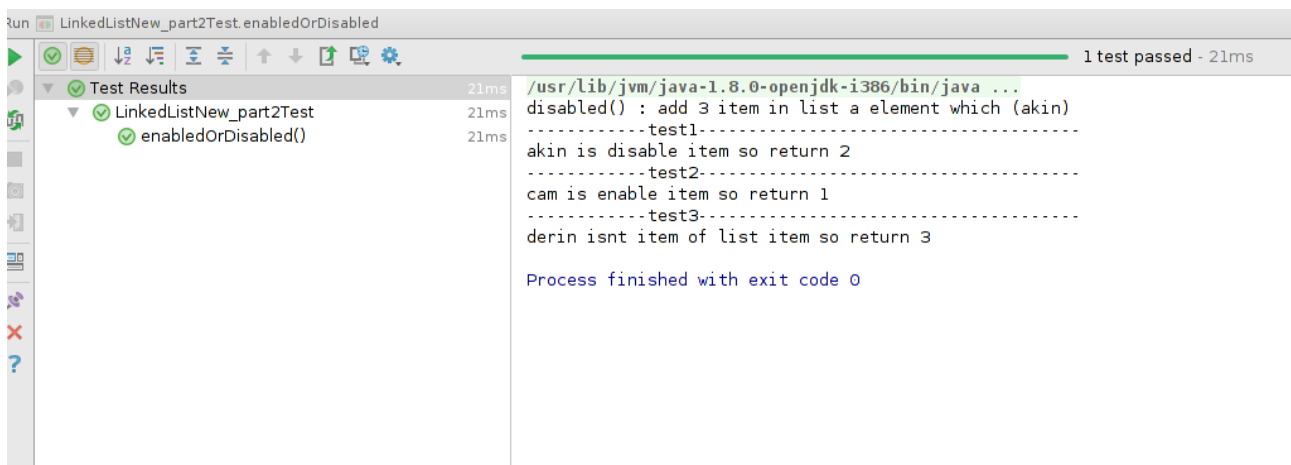


Verilen eleman enabled mı disabled mı yoksa böyle bir eleman mevcut mu listede :

```

@Test
void enabledOrDisabled() {
    LinkedListNew_part2<String> ls=new LinkedListNew_part2<>();
    ls.add("akin");
    ls.add("cam");
    ls.add("151044007");
    System.out.println("disabled() : add 3 item in list a element which (akin)");
    ls.disabled("akin");
    System.out.println("-----test1-----");
    System.out.println("akin is disable item so return 2");
    assertEquals( expected: 2,ls.EnabledOrDisabled( eORd: "akin"));
    System.out.println("-----test2-----");
    System.out.println("cam is enable item so return 1");
    assertEquals( expected: 1,ls.EnabledOrDisabled( eORd: "cam"));
    System.out.println("-----test3-----");
    System.out.println("derin isnt item of list item so return 3");
    assertEquals( expected: 3,ls.EnabledOrDisabled( eORd: "derin"));
}

```



Part3→(JUNIT5)

Circular için kullanılan method:

```

@Test
void nextSemester() throws IOException {
    Curriculum_part1 cr =new Curriculum_part1();
    CircularCustomList_part3<Course_part1> cr1=new CircularCustomList_part3(cr);
    cr1.chooseSemester( 2);
    Course_part1 c1=cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    cr1.nextSemester();
    System.out.println("nextSemester(): creates circular list loop continue relative items-same semestr courses");
    Course_part1 c2=cr1.nextSemester();
    assertEquals(c1,c2);
}

```

Test Results	70ms	70ms	70ms
<ul style="list-style-type: none"> ✓ CircularCustomList_part3Test <ul style="list-style-type: none"> ✓ nextSemester() 			<pre> Course-->{ semester-->8 code-->ENG 402 name-->? ects-->1 gtuCredits-->1 htl--> 0+0+0 } Course-->{ semester-->8 code-->XXX XXX name-->Teknik Olmayan Se0meli (SSB) ects-->3 gtuCredits-->2 htl-->2+0+0 } Course-->{ semester-->8 code-->XXX XXX name-->Teknik Olmayan Se0meli (SSA) ects-->3 gtuCredits-->2 htl-->2+0+0 } 2 2 2 2 2 2 2 2 nextSemester(): creates circular list loop continue relative items-same semestr courses Process finished with exit code 0 </pre>

Seçilmek istenen sömestr için:

```

@Test
void chooseSemester() throws IOException {
    Curriculum_part1 cr =new Curriculum_part1();
    CircularCustomList_part3<Course_part1> crl=new CircularCustomList_part3(cr);
    System.out.println("chooseSemester(): if index is legal return true else false");
    assertEquals( expected: false, crl.chooseSemester( 0: -1));
}

```

```

| htl-->2+0+0
|
|
}

chooseSemester(): if index is legal return true else false
Wrong semester!!! Enter the 1-8

Process finished with exit code 0

```

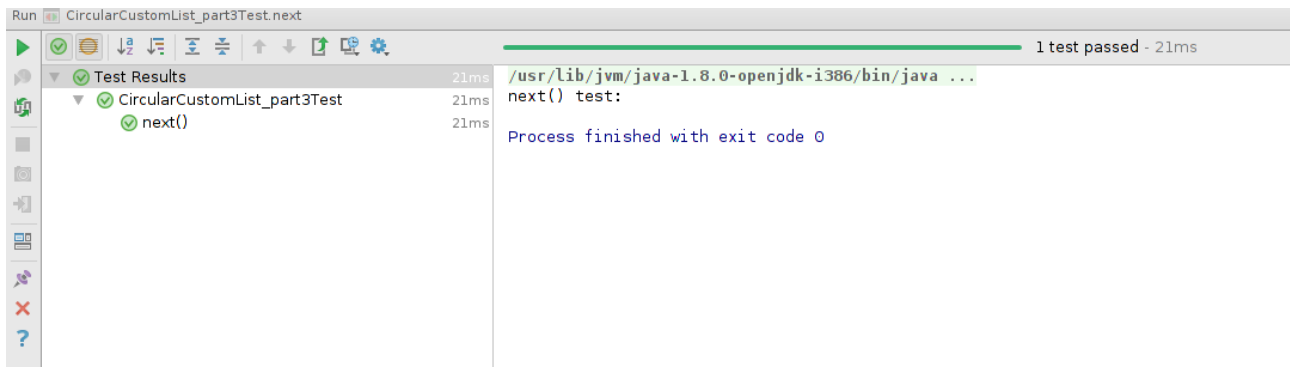
Tests Passed: 1 passed (a minute ago)

Next() methodu:

```

@Test
void next() {
    NewCustomList_part3<Integer> ns=new NewCustomList_part3<>();
    ns.add(1);
    ns.add(2);
    ns.add(3);
    System.out.println("next() test:");
    assertEquals( expected: 1, ns.next().intValue());
}

```

3.2 Running Results

Programı Debian Virtual Machinede(Moodledaki) yazdım ve test ettim.
Junit5 kullandım.

Test caseleri index ile gönderdim.

-----Part1 Test-----

```
public static void main(String[] args) throws IOException {
    String test1[]={"CSE 222","3","6","3","cse 2214","-1","3","-55","2","cse 112","Computer","4","7","2+2+0"};
    System.out.println("-----Part1 test-----");
    try {
        Curriculum_part1 cr = new Curriculum_part1();
        cr.printCourses();
        System.out.println("-----getByCode-----");
        cr.getByCode(test1[0]);
        System.out.println("-----getByRange-----");
        cr.getByRange(Integer.parseInt(test1[1]),Integer.parseInt(test1[2]));
        System.out.println("-----listSemesters-----");
        cr.listSemesterCourses(Integer.parseInt(test1[3]));
        System.out.println("-----addCourse-----");
        cr.addCourse(Integer.parseInt(test1[8]),test1[9],test1[10],Integer.parseInt(test1[11]),Integer.parseInt(test1[12]),test1[13]);
        System.out.println("-----getByCode(Exception)-----");
        cr.getByCode(test1[4]);
    }
    catch (Exception e){
        System.out.println(e.getMessage());
    }
    try{
        Curriculum_part1 cr = new Curriculum_part1();
        System.out.println("-----getByRange(Exception)-----");
        cr.getByRange(Integer.parseInt(test1[5]),Integer.parseInt(test1[6]));
    }
    catch (Exception e) {
        System.out.println(e.getMessage());
    }
    try{
        Curriculum_part1 cr = new Curriculum_part1();
        System.out.println("-----listSemester(Exception)-----");
        cr.listSemesterCourses(Integer.parseInt(test1[7]));
    }
    catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

Test result:

Semester	Course code	Course title	ECTS credits	GTU credits	H+T+L
1	XXX XXX	Teknik Olmayan Se0meli (SSC)	2	1	2+0+0
1	CSE 101	Introduction To Computer Engineering	8	3	3+0+0
1	CSE 107	Introduction To Computer Science Laboratory	2	1	0+0+2
1	MATH 101	Calculus I	7	5	5+0+0
1	PHYS 121	Physics I	6	4	3+0+0
1	PHYS 151	Physics Laboratory I	1	1	0+0+2
1	SSTR 101	Principles Of Atat0rk And The History Of Turkish Revolution I	2	2	2+0+0
1	TUR 101	Turkish I	2	2	2+0+0
2	XXX XXX	Teknik Olmayan Se0meli (SSC)	2	1	2+0+0
2	CSE 102	Computer Programming	8	4	4+0+0
2	CSE 108	Computer Programming Laboratory	2	1	0+0+2
2	MATH 102	Calculus II	7	5	5+0+0
2	PHYS 122	Physics II	6	4	3+0+0
2	PHYS 152	Physics Laboratory II	1	1	0+0+2
2	SSTR 102	Principles Of Atat0rk And The History Of Turkish Revolution II	2	2	2+0+0
2	TUR 102	Turkish II	2	2	2+0+0
3	CSE 241	Object Oriented Programming	9	5	3+2+0
3	CSE 211	Discrete Mathematics	6	3	3+0+0
3	CSE 231	Circuits And Electronics	8	4	4+0+0
3	CSE 233	Circuits And Electronics Laboratory	2	1	0+0+2
3	XXX XXX	Teknik Olmayan Se0meli (SSB)	3	2	2+0+0
3	EN 111	English For Business Life	2	2	2+0+0
4	CSE 222	Data Structures And Algorithms	9	5	4+2+0
4	MATH 217	Linearr Algebra And Differential Equations	8	5	4+2+0
4	CSE 232	Logic Circuits And Design	6	3	3+0+0
4	CSE 234	Logic Circuits And Design Laboratory	2	1	0+0+2
4	XXX XXX	Teknik Olmayan Se0meli (SSB)	3	2	2+0+0
4	EN 112	Academic English	2	2	2+0+0
5	CSE 343	Software Engineering	8	4	3+2+0
5	CSE 341	Programming Languages	6	3	3+0+0
5	CSE 331	Computer Organizations	7	4	3+2+0
5	CSE 321	Introduction To Algorithm Design	6	3	3+0+0
5	XXX XXX	Teknik Olmayan Se0meli (SSA)	3	2	2+0+0
5	ENG 300	Career Internship I	1	1	0+6+0
6	CSE 396	Computer Engineering Project	5	2	1+2+0
6	CSE 312	Operating Systems	6	3	3+0+2
6	MATH 118	Probability and Statistics	6	3	3+0+0
6	CSE 344	System Programming	3	2	0+0+0
6	CSE 351	Signals and Systems	6	3	0+0+0
6	XXX XXX	Teknik Olmayan Se0meli (SSA)	3	0	2+0+0
7	CSE 495	Graduation Project I	6	1	4+0+0
7	CSE 4XX	Department Elective I	6	3	3+0+0
7	CSE 4XX	B0l0m Se0meli II	6	3	3+0+0
7	CSE 4XX	B0l0m Se0meli (Temel Alan) I	6	3	3+0+0
7	ENG 401	Occupational Health and Safety	1	1	0+0+0
7	XXX XXX	Teknik Olmayan Se0meli (SSB)	3	2	2+0+0
7	ENG 400	Career Internship II	1	1	0+0+6
8	CSE 496	Graduation Project II	6	1	4+0+0
8	CSE 4XX	Department Elective III	6	3	3+0+0
8	CSE 4XX	B0l0m Se0meli IV	6	3	3+0+0
8	CSE 4XX	B0l0m Se0meli (Temel Alan) Se0meli II	6	3	3+0+0
8	ENG 402	?	1	1	0+0+0
8	XXX XXX	Teknik Olmayan Se0meli (SSB)	3	2	2+0+0
8	XXX XXX	Teknik Olmayan Se0meli (SSA)	3	2	2+0+0

-----getByCode-----

```
Course-->{
|
| semester-->4
| code-->CSE 222
| name-->Data Structures And Algorithms
| ects-->9
| gtuCredits-->5
| htl-->4+2+0
|
}
```

-----getByRange-----

```
Course-->{
|
| semester-->1
| code-->MATH 101
| name-->Calculus I
| ects-->7
| gtuCredits-->5
| htl-->5+0+0
|
}
```

```
Course-->{
|
| semester-->1
| code-->PHYS 121
| name-->Physics I
| ects-->6
| gtuCredits-->4
| htl-->3+0+0
|
}
```

```
Course-->{
|
| semester-->1
| code-->PHYS 151
| name-->Physics Laboratory I
| ects-->1
| gtuCredits-->1
| htl-->0+0+2
|
}
```

```
Course--->{
|
| semester-->1
| code-->SSTR 101
| name-->Principles Of Atatürk And The History Of Turkish Revolution I
| ects-->2
| gtuCredits-->2
| htl-->2+0+0
|
}
```

-----listSemesters-----

```
Course--->{
|
| semester-->3
| code-->CSE 241
| name-->Object Oriented Programming
| ects-->9
| gtuCredits-->5
| htl-->3+2+0
|
}
```

```
Course--->{
|
| semester-->3
| code-->CSE 211
| name-->Discrete Mathematics
| ects-->6
| gtuCredits-->3
| htl-->3+0+0
|
}
```

```
Course--->{
|
| semester-->3
| code-->CSE 231
| name-->Circuits And Electronics
| ects-->8
| gtuCredits-->4
| htl-->4+0+0
|
}
```

```

Course-->{

    semester-->3
    code-->CSE 233
    name-->Circuits And Electronics Laboratory
    ects-->2
    gtuCredits-->1
    htl-->0+0+2

```

```

Course-->{

    semester-->3
    code-->XXX XXX
    name-->Teknik Olmayan Se0meli (SSB)
    ects-->3
    gtuCredits-->2
    htl-->2+0+0

```

```

Course-->{

    semester-->3
    code-->EN 111
    name-->English For Business Life
    ects-->2
    gtuCredits-->2
    htl-->2+0+0

```

```

-----addCourse-----
Course-->{

    semester-->2
    code-->cse 112
    name-->Computer
    ects-->4
    gtuCredits-->7
    htl-->2+2+0

```

```

-----addCourse-----
Course-->{
| semester-->2
| code-->cse 112
| name-->Computer
| ects-->4
| gtuCredits-->7
| htl-->2+2+0
|
}

```

```

-----getByCode(Exception)-----
There is no matching code!!!
-----getByRange(Exception)-----
Wrong input in getByRange!!!!!!
-----listSemester(Exception)-----
There is no matching semester

```

-----Part2 Test-----

```

System.out.println("-----");
System.out.println("-----");
System.out.println("-----");
System.out.println("-----Part2 test-----");
String test2[]={"2","23","23"};
Curriculum_part1 cr1 = new Curriculum_part1();
LinkedListNew_part2<Course_part1> ls=new LinkedListNew_part2<>();
for(int i=0;i<cr1.getCourseNumber();i++)
    ls.add(cr1.getCourseData().get(i));
System.out.println("-----soon to be disabled element-----");
System.out.println(cr1.getCourseData().get(Integer.parseInt(test2[0])));
System.out.println("-----before disabled size-----");
System.out.println(ls.size());
ls.disabled(cr1.getCourseData().get(Integer.parseInt(test2[0])));
System.out.println("-----before disabled size-----");
System.out.println(ls.size());
System.out.println("-----soon to be disabled element-----");
System.out.println(cr1.getCourseData().get(Integer.parseInt(test2[0])));
System.out.println("-----before disabled size-----");
System.out.println(ls.size());
ls.disabled(cr1.getCourseData().get(Integer.parseInt(test2[1])));
System.out.println("-----after disabled size-----");
System.out.println(ls.size());
System.out.println("-----ShowsDisabled-----");
ls.showDisabled();
System.out.println("-----before Enable second disabled element index -----");
System.out.println(ls.get(Integer.parseInt(test2[2])));
System.out.println("-----after Enable second disabled element index -----");
ls.enabled(cr1.getCourseData().get(Integer.parseInt(test2[2])));
System.out.println(ls.get(Integer.parseInt(test2[2])-1));

```

Test result:

```

-----soon to be disabled element-----
Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
}

-----before disabled size-----
54
-----before disabled size-----
53
-----soon to be disabled element-----
Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
}

-----before disabled size-----
53
-----after disabled size-----
52
-----ShowsDisabled-----
Disabled items are ordered:
--> Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
}

```

```
--> Course-->{
|
| semester-->4
| code-->MATH 217
| name-->Linearr Algebra and Differantial Equations
| ects-->8
| gtuCredits-->5
| htl-->4+2+0
|
}
```

-----before Enable second disabled element index -----

```
Course-->{
|
| semester-->4
| code-->CSE 234
| name-->Logic Circuits And Design Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
}
```

-----after Enable second disabled element index -----

```
Course-->{
|
| semester-->4
| code-->MATH 217
| name-->Linearr Algebra and Differantial Equations
| ects-->8
| gtuCredits-->5
| htl-->4+2+0
|
}
```

-----Part2-Test2-----


```

System.out.println("-----");
System.out.println("-----");
System.out.println("-----");
System.out.println("-----Part2 test-----\n\n");
try {
    String test3[] = {"a", "e", "f", "enable", "e", "enable", "a", "enable", "disable"};
    LinkedListNew_part2<String> ls1 = new LinkedListNew_part2<>();
    ls1.add("a");
    ls1.add("b");
    ls1.add("c");
    ls1.add("d");
    ls1.add("e");
    ls1.add("f");
    ls1.add("g");
    System.out.println("-----ShowsSize-----");
    System.out.println(ls1.size());
    ls1.disabled(test3[0]);
    ls1.disabled(test3[1]);
    ls1.disabled(test3[2]);

    System.out.println("-----ShowsSizeafterdisable-----");
    System.out.println(ls1.size());
    System.out.println("-----ShowsDisabled-----");
    ls1.showDisabled();
    System.out.println("-----ShowsEnabled-----");
    ls1.printItems(test3[3]);
    ls1.enabled(test3[4]);
    System.out.println("-----ShowsEnabled-----");
    ls1.printItems(test3[5]);
    System.out.println("-----ShowsSize-----");
    System.out.println(ls1.size());
    ls1.enabled(test3[6]);
    System.out.println("-----ShowsEnabled-----");
    ls1.printItems(test3[7]);
    System.out.println("-----Showsdisabled-----");
    ls1.printItems(test3[8]);
}
catch (Exception e){
    e.getMessage();
}

```

Test2 result:

```
-----
-----
-----
-----Part2 test-----
```

```
-----ShowsSize-----
7
-----ShowsSizeafterdisable-----
4
-----ShowsDisabled-----
Disabled items are ordered:
-->  a
-->  e
-->  f
-----ShowsEnabled-----
b
c
d
g
-----ShowsEnabled-----
b
c
d
e
g
-----ShowsSize-----
5
-----ShowsEnabled-----
a
b
c
d
e
g
-----Showsdisabled-----
f
-----
-----
-----
-----Part3 test-----
```

-----Part3 Test----

```

int test4[] = {1,15,3,21};
try{
    Curriculum_part1 crs=new Curriculum_part1();
    CircularCustomList_part3 circular=new CircularCustomList_part3<>(crs);
    circular.chooseSemester(test4[0]);
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println(circular.nextSemester());
    System.out.println("-----returns first index-----");
    System.out.println(circular.nextSemester());
    System.out.println("-----wrong semester-----");
    circular.chooseSemester(test4[1]);
    System.out.println("-----normally next-----");
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println(circular.next());
    System.out.println("-----size-----");
    circular.size();
    System.out.println("-----get method-----");
    circular.get(test4[2]);
    System.out.println("-----clear all things looks size-----");
    circular.clear();
    System.out.println("-----remove element from index-----");
    System.out.println(circular.remove(test4[3]));
}
catch (Exception e){
    e.getMessage();
}

```

Test3 results:

Normal nextSemesterda aynı dönem derslerinde döner.

Next de farklı dönemlere geçer.

Ok ile işaretledim.

```
1
Course-->{
|
| semester-->1
| code-->XXX XXX
| name-->Teknik Olmayan Se0meli (SSC)
| ects-->2
| gtuCredits-->1
| htl-->2+0+0
|
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->CSE 101
| name-->Introduction To Computer Engineering
| ects-->8
| gtuCredits-->3
| htl-->3+0+0
|
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->MATH 101
| name-->Calculus I
| ects-->7
| gtuCredits-->5
| htl-->5+0+0
|
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->PHYS 121
| name-->Physics I
| ects-->6
| gtuCredits-->4
| htl-->3+0+0
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->PHYS 151
| name-->Physics Laboratory I
| ects-->1
| gtuCredits-->1
| htl-->0+0+2
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->SSTR 101
| name-->Principles Of Atatürk And The History Of Turkish Revolution I
| ects-->2
| gtuCredits-->2
| htl-->2+0+0
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->TUR 101
| name-->Turkish I
| ects-->2
| gtuCredits-->2
| htl-->2+0+0
|
}
```

-----returns first index-----

```
Course-->{
|
| semester-->1
| code-->XXX XXX
| name-->Teknik Olmayan Se0meli (SSC)
| ects-->2
| gtuCredits-->1
| htl-->2+0+0
|
}
```

-----wrong semester-----

Wrong semester!!! Enter the 1-8

-----normally next-----

```
1
Course-->{
|
| semester-->1
| code-->CSE 101
| name-->Introduction To Computer Engineering
| ects-->8
| gtuCredits-->3
| htl-->3+0+0
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->CSE 107
| name-->Introduction To Computer Science Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
}
```

.

```
Course-->1
|
| semester-->1
| code-->MATH 101
| name-->Calculus I
| ects-->7
| gtuCredits-->5
| htl-->5+0+0
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->PHYS 121
| name-->Physics I
| ects-->6
| gtuCredits-->4
| htl-->3+0+0
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->PHYS 151
| name-->Physics Laboratory I
| ects-->1
| gtuCredits-->1
| htl-->0+0+2
|
}
```

```
1
Course-->{
|
| semester-->1
| code-->SSTR 101
| name-->Principles Of Atatürk And The History Of Turkish Revolution I
| ects-->2
| gtuCredits-->2
| htl-->2+0+0
|
}
```

```

1
Course--->{
|
| semester-->1
| code-->TUR 101
| name-->Turkish I
| ects-->2
| gtuCredits-->2
| htl-->2+0+0
|
|
}

2
Course--->{
|
| semester-->2
| code-->XXX XXX
| name-->Teknik Olmayan Se0meli (SSC)
| ects-->2
| gtuCredits-->1
| htl-->2+0+0
|
|
}

2
Course--->{
|
| semester-->2
| code-->CSE 102
| name-->Computer Programming
| ects-->8
| gtuCredits-->4
| htl-->4+0+0
|
|
}

2
Course--->{
|
| semester-->2
| code-->CSE 108
| name-->Computer Programming Laboratory
| ects-->2
| gtuCredits-->1
| htl-->0+0+2
|
|
}

```

Simulation completed successfully in 1e 223ms (4 minutes ago)

```

-----size-----
54
-----get method-----
Course--->{
|
| semester-->1
| code-->MATH 101
| name-->Calculus I
| ects-->7
| gtuCredits-->5
| htl-->5+0+0
|
|
}

```

```

-----clear all things looks size-----
0
-----remove element from index-----

```

Process finished with exit code 0