

**Gebze Technical University
ComputerEngineering**

CSE 222 -2018 Spring

HOMEWORK 3 REPORT

**YUNUS ÇEVİK
141044080**

Course Assistant: MEHMET BURAK KOCA

1 INTRODUCTION

1.1 Problem Definition

1.1.1 Part1

“Courses(CSV).csv” adlı dosyada tutulan GTU Bilgisayar Mühendisliği Derslerinin, dosyadan okunarak LinkedList yapısı içerisine eklenerek Ders Kodu(Course Code), Dönem(Semester) ile sorgulama işlemi yaptıracağımız **getByCode** ve **listSemesterCourses** metodları ile birlikte liste yapısının indeksleri arasında bilgileri veren **getByRange** metodlarını implement etmemiz istenmektedir.

1.1.2 Part2

LinkedList yapısından extends ettiğimiz bir sınıf oluşturup, LinkedList yapısının özellikleri ile beraber elemanlar üzerinde enable, disable ve showDisable metodlarını implement etmemiz istenmektedir.

disable(): Disable edilen bir elemanın liste ile işi olmayıp listenin eleman sayısından da azaltılması sağlanmalıdır.

Örnek: list = { 1 2 3 4 5 } size = 5
list.disable(index) index => 2 (index 0 dan başlıyor varsayalım)
list = { 1 2 4 5 } size = 4
list.disable(index) index => 2 (2. indexe 3. indexdeki bilgi gelir)
list = { 1 2 5 }

showDisable(): Disable edilen elemanların kendi index değerleri ile tutulduğu bir başka LinkedList yapısında tutulması sağlanmalıdır.

Örnek: Yukarıda disable edilmiş listeyi göz önünde bulundurursak
list.showDisable()
disableList = { {2, 3} {3, 4} } size = 2
İlk değer eski listedeki index değerini, ikinci değer ise veriyi temsil eder.

enable(): Disable edilmiş bir elemanın enable edilerek eski konumuna geçmesi sağlanmalıdır.

Örnek: Yukarıda disable edilmiş listeyi göz önünde bulundurursak
list.showDisable()
disableList = { {2, 3} {3, 4} } size = 2

```
list.enable(2)
list = { 1 2 3 5 }    size = 4
list.showDisable()
disableList = { {3, 4} }    size = 1
list.enable(3)
list = { 1 2 3 4 5 }    size = 5
list.showDisable()
disableList = { }    size = 0
```

Disable edilmiş tüm elemanlar enable edildiğinde listenin ilk haline dönmesi beklenmektedir.

1.1.3 Part3

LinkedList yapısını kullanmadan kendi LinkedList yapısına benzer bir yapı yapmamız istenmiştir. Bu yapıyı oluşturabilmek için dizi(array) yapısı kullanılabilir. Ayrıca bizden istenen zorunlu metodları da implement etmemiz beklenmektedir.

Bunlar:

- 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

1.2 System Requirements

Proje, IntelliJ projesi olarak oluşturulmalıdır. Java 8 uyumlu olmalıdır. Windows ve Linux ortamlarında yüklü IntelliJ IDE' sinde çalışabilir olmalıdır. Ayrıca projeyi test etmek için bizlere sunulan Virtual Machine içinde bulunan IntelliJ IDE' sinde de test edilebilir olmalıdır. Benim oluşturmuş olduğum proje bu şartlara uymaktadır. Projede belirtilen part - part ayrılmış kısımların gereksinimleri aşağıda anlatılmaktadır.

1.2.1 Part1

Part1 için gerekli olanlar:

- Proje dosyamızın içinde Course.java class' ı oluşturarak GTU Bilgisayar Mühendisliğine ait ders yapılarının tutulacağı bir alan oluşturulmalıdır.

```

public class Course {
    private int semester;
    private String courseCode;
    private String courseTitle;
    private String ECTSCredits;
    private String GTUCredits;
    private String HTL;
}

```

- “Courses(CSV).csv” adlı dosyada bulunan ders bilgileri Course sınıfının verilerine aktarılmalıdır.
- Her dersin ayrı ayrı bir liste yapısında tutulması için ise LinkedList<Course> objesi oluşturularak içerisine add metodu ile tüm derslerin bilgileri aktarılmalıdır.
- Bu listenin kullanacağı ve bizden istenen getByCode, listSemesterCourses ve getByRange metodları implement edilmelidir.

Bu metodların prototipleri şöyle olmalıdır.

- `public LinkedList<Course> getByCode (String code) throws Exception;`
- `public LinkedList<Course> listSemesterCourses (int semester) throws Exception;`
- `public LinkedList<Course> getByRange(int start_index, int last_index) throws Exception;`

1.2.2 Part2

Part2 için gerekli olanlar:

- Proje dosyamızın içinde index ve herhangi bir object in tutulacağı bir yardımcı sınıf oluşturularak içerisinde generic olarak bilgi tutulması sağlanır.

Örnek: ClassHasIndexAndObject.java

```

public class ClassHasIndexAndObject {
    private int index;
    private Object object;
}

```

- Generic bir Part2 projesi oluşturulur ve LinkedList yapısından extends edilir.

```

public class Part2<E> extends LinkedList<ClassHasIndexAndObject>

```

- Bu listenin kullanacağı ve bizden istenen enable, disable ve showDisable implement edilmelidir.

Bu metodların prototipleri şöyle olmalıdır.

- `public void enable(int index);`
- `public void disable(int index);`
- `public void showDisable();`
-

Not: Listeye bir eleman add metodu ile eklendiğinde parametresinde ClassHasIndexAndObject tipinde veri alır ve bulunduğu index değerini bu class tipinde de tutar. Bu yüzden add metodu Override edilmelidir ve aynı indexe sahip başka bir veri geldiği zaman bu yardımcı class tipindeki index değerlerindeki birer artırmalıdır.

1.2.3 Part3

Part3 için gerekli olanlar:

- Part3 sınıfı oluşturup LinkedList yapısına benzer dizi(array)'den oluşan bir veri yapısı belirleyip bu dizi üzerine bazı metodlar implement ederek generic bir LinkedList yapısını oluşturulmalıdır.

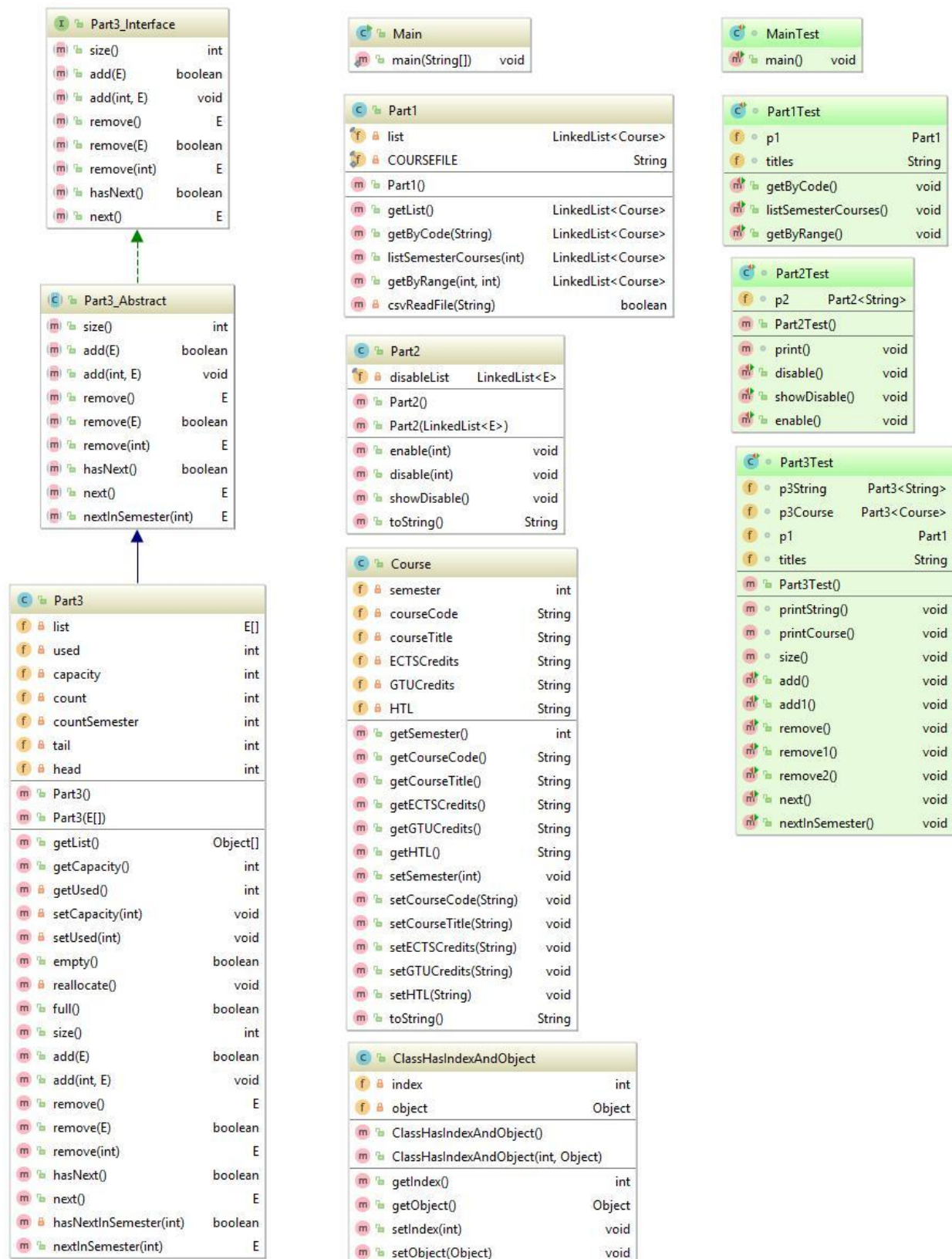
Bu metodların prototipleri şöyle olmalıdır.

- `public int size();`
- `public boolean add(E value);`
- `public void add(int index, E value);`
- `public E remove();`
- `public boolean remove(E value);`
- `public E remove(int index);`
- `public E next() throws NoSuchElementException;`
- `public E nextInSemester(int semester) throws NoSuchElementException;`

Not: Yukarıda belirtilen Part1, Part2 ve Part3 sınıflarının gerekli metodları ilerleyen bölümlerde detaylı anlatılacaktır.

2 METHOD

2.1 Class Diagrams



2.1.1 Class Diagram Documentantation:

2.1.1.1 Part1

public class Part1 => İçerisinde LinkedList objesi bulunan bu class, bu objenin içine "Courses(CSV).csv" dosyasından okunan bilgilerle doldurulmuş Course bilgilerini LinkedList metodları ve sonradan implement edilen getByCode, listSemesterCourses ve getByRange bu metodlarla kullanma imkanı sağlar.

public LinkedList<Course> getByCode (String code) throws Exception; => Parametresinde aldığı Course code ile Course' ların içinde tutulduğu listede arama yapar ve bu Course Code' una sahip Course' ları bir liste olarak return eder.

public LinkedList<Course> listSemesterCourses (int semester) throws Exception; => Parametresinde aldığı semester değeri ile Course' ların içinde tutulduğu listede arama yapar ve semester değerine sahip Course' lari bir liste olarak return eder.

public LinkedList<Course> getByRange(int start_index, int last_index) throws Exception; => Verilen indeksler arasında bulunan Course bilgilerini bir liste olarak return eder.

private boolean csvReadFile(String fileName); => CSV dosyasından okuma işlemi yaparak LinkedList objesinin içine ekler. İşlem başarılı ise true başarısız ise false return eder.

2.1.1.2 Part2

public class Part2<E> extends LinkedList<ClassHasIndexAndObject> => LinkedList' ten extends edilmiş bu generic class ile LinkedList metodları ile birlikte sonradan implement edilen enable, disable ve showDisable metodları kullanılır.

public void enable(int index); => Disable edilmiş olan veriyi index numarasına göre enable etme işine yarayan metoddur. Ayrıca enable edilmiş olan veri disable edilmeden önce olan konumuna tekrar geçer.

public void disable(int index); => Index numarası belirtilmiş listeden ders disable edilerek çıkarılır ve başka bir listeye eklenir.

public void showDisable(); => Disable edilmiş verileri ekran çıktısı olarak gösterir.

2.1.1.3 Part3

public interface Part3_Interface<E> => Objesi üretilmeyen ancak Polimorfik çağrılarda kullanılan interface.

public abstract class Part3_Abstract<E> implements Part3_Interface<E> => Objesi üretilmeyen ancak başka classlara miras ile aktardığı metodların kullanımı sağlayan abstract class. Bu abstract class Part3_Interface interface' inden implement edilmedir.

public class Part3<E> extends Part3_Abstract<E> => Generic bir array ile liste yapısı oluşturulur. Ayrıca implement edilen add, remove, next, nextInSemester ve size metodları kullanılır.

public boolean add(E value); => Listenin sonuna parametreden gelen veriyi ekleme işlemi yapar.

public void add(int index, E value); => Liste üzerinde parametreden gelen index değeri ve veriyi index değerine göre ekleme işlemi yapar. Ayrıca listede araya ekleme işlemi yapıldığında kaydırması gerekirse, kaydırma işlemi de gerçekleştirir.

public E remove(); => Listenin son elemanını silmeye yarar.

public boolean remove(E value); => Parametrede gelen değere göre liste üzerinde arama yapar ve listede varsa bu değer silinir.

public E remove(int index); => Parametrede gelen index değerine göre liste üzerinde arama yapar ve listede varsa o indexe sahip değer silinir.

public E next() throws NoSuchElementException; => Liste üzerinde next metodu listedeki verilere sırayla ulaşır ve ilerleme gerçekleştirilir.

public E nextInSemester(int semester) throws NoSuchElementException; => Semester değerine göre liste üzerinde bulunan o semestere sahip olan veriler üzerinde ilerlemek ve verileri return etmek için bu metod kullanılır. Ayrıca semester değerine göre son bulunan semesterdan sonra bu metod çağrıldığı takdirde ilk bulunan semester verisine gider. Circular bir yapı içermektedir.

public int size(); => Listenin içindeki eleman bilgisini verir.

2.2 Use Case Diagrams

Add use case diagrams if required.(Gerekmemektedir.)

2.3 OtherDiagrams (optional)

Add other diagrams if required. (Gerekmemektedir.)

2.4 Problem Solution Approach

2.4.1 Part1

- “Course(CSV).csv” dosyası BufferedReader ile okunarak, okunan veriler LinkedList<Course> objesine add metodu ile eklenir.
- Liste içerisine eklenen verileri sorgulama işlemini bizden istenen üç metod ile gerçekleştirilir.

Bunlar metodlar:

LinkedList<Course> getByCode (String code) => Parametre olarak Course Code alan bu metod, dersin kodunu aldıktan sonra bu ders kodu ile search işlemi gerçekleştirir. Bu ders koduna ait ders verileri metoddan geri döndürülecek LinkedList<Course> listesine eklenir ve geri döndürülür.

LinkedList<Course> listSemesterCourses (int semester) => Parametre olarak semester (dönem) alan bu metod, dersin bulunduğu dönemi aldıktan sonra bu dönem ile search işlemi gerçekleştirir. Bu döneme ait ders verileri metoddan geri döndürülecek LinkedList<Course> listesine eklenir ve geri döndürülür.

LinkedList<Course> getByRange(int start_index, int last_index) => Parametrelerde belirtilen start_index ve last_index değerlerini alarak listede üzerindeki bu index değerleri arasında bulun verileri metoddan geri döndürülecek LinkedList<Course> listesine eklenir ve geri döndürülür.

Yukarıda implement edilmiş metodlar ile, derslerin bulunduğu bir listede sorgula işlemleri çözülmüş olur.

2.4.2 Part2

- Liste içerisine eklenen verileri enable, disable ve showDisable etme metodları istenmişti. Bu metodlar aşağıdaki gibi tasarlanıp bizden istenen problemler çözülmüştür.

void disable(int index); => Bizden istenen bu metod gerekli görülen dersin devre dışı bırakılmasını sağlamaktadır. Bu metodun tasarımında bir verinin disable edilme işlemi silinerek başka generic bir LinkedList<E> yapısına aktarılması ile sağlanmıştır. Ana listede disable edildiği takdirde o veri herhangi bir işlem yapmayacağından listeden silinir ve doğal olarak liste içindeki eleman sayısı düşer böylece size() metodu çağrıldığında elemanların azaldığı görünür. Problem definition kısmında belirtildiği gibi yapılmıştır. Ayrıca Part2.java sınıfı LinkedList<ClassHasIndexAndObject> extends edildiğinden ClassHasIndexAndObject sınıfı gerekmektedir. Bu sınıf içerisinde index ve obje yapısını tutmaktadır. Disable edilen verinin ilk bulunduğu konum indexi ile birlikte veri değerini disable edilmişlerin olduğu listeye rahatça ekleyebilmemiz sağlanmıştır.

Örnek: liste = { { index: integer, object: Course} } ya da liste { { index: integer, object: String} }

void showDisable(); => Disable edilen veriler başka bir LinkedList<E> içerisine aktarılır. Bu metod ile disable edilen verilerin neler olduğunu görmemizi sağlayan bir metoddur. Ayrıca disable edilmeden showDisable metodu çağrıldığında kullanıcıya listede disable edilen elemanların olmadığını belirten bir hata mesajı verecek.

void enable(int index); => Disable edilen verilerin tekrardan etkinleştirilmesi için enable metodu çağrılır. Parametreye verilen index değeri ile hangi verinin enable edileceği belirlenir. showDisable metodunun çağrıldıktan sonra hangi elemanların disable edildiği görünür ve elemanların ilk index değerleri ({ index, object } olarak) ile enable metodu çağrıldığında disable edilenler listesinden ana listeye aktarılır ve disable edilenlerin listesinden silinir. Ana listeye aktarılan değer ilk index değerine bağlı olarak döngüler yardımıyla ana listeye taranır ve eski konumuna getirilir. Problem definition da belirtildiği gibi yapılmıştır. Sonuç olarak ana liste üzerinde disable yapılan veriler, daha sonra hepsi enable edilğinde showDisable içinde hiç eleman bulunmayacak ana liste ise başlangıçta olan şekline dönecektir. Bununla ilgili bir örnekte Problem definition da belirtilmiştir.

Bu metodların implementleri ile bizden istenen yapılar çözülmüş olur.

2.4.3 Part3

LinkedList yapısını kullanmadan kendi LinkedList yapısına benzer bir yapı ve bu yapının kullanabileceği metodlar oluşturmamız istemişti. LinkedList yapısını dizi(array) vs. yapılarla çözebiliriz. Part3 de dizi yapısını kullanarak LinkedList yapısına benzer yapı oluşturarak diğer metodların bu yapıya uygun bir şekilde implementi gerçekleştirilmiştir.

- Array yapısında bir capacity ve size belirlenir ve ilk olarak capacitylik yer kadar allocate edilir. Daha sonra capacity eğer size a eşit olduğunda reallocate metodu çağrılarak alınan capacitylik alan iki katına çıkarılarak hafızada yer artırılır.
- Array yapısının kullanacağı ve bizden istenen metodlar şu şekilde implement edilmiştir.
 - **boolean add(E value); =>** Add metodu parametresine gelen veriyi arrayin son elemanından sonrasına ekler. Eğer capacity size a eşit olursa reallocate metodu çağrılır ve içeride bulunan veriler geçici listeye alınır ve hafıza iki katına çıkarıldıktan sonra tekrar kopyalama işlemiyle yerine yerleştirilir ve parametreden gelen eleman sona eklenir. Ayrıca eleman eklendiğinde true döndürür. Bu metodun çözümünde sadece elemanın son elemanının bilinmesi yeterlidir. Çünkü son elemandan sonra yeni bir elemanın ekleneceği belirtilir.
 - **void add(int index, E value); =>** Overload edilmiş olan add metodu iki parametre alır. Bu parameterler index ve E tipinde bir veridir. Bu metod ile ekleme işlemi yapmak istediğimizde parametreye gelen veri index değerine göre array üzerinde indexin gösterdiği yere veri eklenir ve kayması gereken veriler kayar. Eğer capacity size a eşit olursa reallocate metodu çağrılır ve içeride bulunan veriler geçici listeye alınır ve hafıza iki katına çıkarıldıktan sonra tekrar kopyama işlemiyle yerine yerleştirilir ve parametreden gelen eleman indexin gösterdiği yere eklenir.
 - **E remove(); =>** Remove metodu çağrıldığında arrayin son elemanını silme işlemine yarar ve size değeri azaltılır. Silme işlemi sonrasında silinen değer return edilir.

- **boolean remove(E value);** => Remove metodu çağrıldığında aldığı parametrede bulunan veri array üzerinde search edilerek bulunur ve silme işlemi gerçekleştirilir. Silme sonrası kayması gereken elemanlar silinen veri üzerine kayma işlemini gerçekleştirir. Silme işlemi başarılı ise true, silme işlemi başarısız ise false return edilir.
- **E remove(int index);** => Remove metodu çağrıldığında aldığı parametrede bulunan index değerine göre array üzerinde search edilerek bulunur ve silme işlemi gerçekleştirilir. Silme sonrası kayması gereken elemanlar silinen veri üzerine kayma işlemini gerçekleştirir. Silme işlemi sonrasında silinen değer return edilir.
- **E next();** => Arrayin boyutuna göre ilerleme gerçekleştirilir. Ayrıca arrayin ilk elemanından son elemanına kadar her çağrılmada ilerleyerek hareketi sağlanır. Kısacası her bir next işlemi bir sonraki veriyi vermektedir.
- **E nextInSemester(int semester);** => Semester parametresine göre liste içinde bir search işlemi gerçekleştirip semester değerine sahip kaç eleman olduğu saptanır. Semester değerine sahip ilk ve son elemanların indexleri tutulur. Her bu metod çağrıldığında ilk elemandan son elemana sadece parametrede belirtilen ilerleme işlemi gerçekleştirilir ve her işlemde o semester değerine sahip veri return edilir. Ayrıca circular bir yapı vardır son elemanda iken bu metod çağrıldığında tekrardan semester değerine sahip ilk bulunan elemana geri döner.
- **int size();** => Listenin içindeki eleman bilgisini verir.

Yukarıda belirtilen metodların ne yaptığı ve nasıl yaptığı anlatılır. Bizden istenen metodların çözümleri belirtilmiştir. Ayrıca 4. Bölüm Complexity Analysis bölümünde tüm partlarda bulunan ve bizden istenen metodların analizleri yapılmıştır.

3 RESULT

3.1 Test Cases

3.1.1 Main Tets

Main Test için projede bulunan main metodu yazılmıştır. Tüm partlarda bizden istenen metodlar tek tek kullanılarak gösterilmiştir. Screen Shot olarak Running Results bölümünde gösterilecektir.

3.1.2 Unit Test

3.1.2.1 Part1

getByCodeTest

The screenshot shows the IntelliJ IDEA IDE with a Java test file named `Part1Test.java` open. The code includes a `getByRange` method and a `getByCode` method. The `getByCode` method is highlighted in blue. The `Run` button is also highlighted. The `Run` panel on the right shows the test results for `Part1Test`, indicating that all tests passed successfully. The `Test Results` panel shows the following test results:

- `listSemesterCourses()` - 31ms
- `getByCode()` - 31ms
- `getByRange()` - 31ms

The `Test Results` panel also shows a summary: `All 3 tests passed - 31ms`. Below the summary, the `Part1Test` class is listed with its methods and their execution times.

| Method | Execution Time |
|------------------------------------|----------------|
| <code>listSemesterCourses()</code> | 31ms |
| <code>getByCode()</code> | 31ms |
| <code>getByRange()</code> | 31ms |

The `Part1Test` class is also shown with its methods and their execution times:

| Method | Execution Time |
|------------------------------------|----------------|
| <code>listSemesterCourses()</code> | 31ms |
| <code>getByCode()</code> | 31ms |
| <code>getByRange()</code> | 31ms |

The `Part1Test` class is also shown with its methods and their execution times:

| Method | Execution Time |
|------------------------------------|----------------|
| <code>listSemesterCourses()</code> | 31ms |
| <code>getByCode()</code> | 31ms |
| <code>getByRange()</code> | 31ms |

listSemesterCoursesTest

HW3Project [C:\Users\Asus\Desktop\intelliProject\HW3Project] - ...test\HW3\Part1Test.java [HW3Project] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 Part1Test

Part1Test.java

```
30 System.out.println(plPrint.get(i).toString());
31 }
32 } catch (Exception e) {
33     System.out.println(e);
34 }
35 }
36
37 /**
38  * Parametre olarak girilen bir Semester degeri ile o semester e ait tum Course lari list
39  */
40 @Test
41 void listSemesterCourses() {
42     System.out.println("\n\t----- PART1_TEST <Course>-----\n");
43     try{
44         System.out.println("\n\t----- listSemesterCourses(semester) semester=> 2 -----
45         LinkedList<Course> plPrint = pl.listSemesterCourses(2);
46         System.out.println(titles);
47         for(int i = 0; i < plPrint.size(); i++) {
48             assertEquals( expected: 2,plPrint.get(i).getSemester());
49             System.out.println(plPrint.get(i).toString());
50         }
51     } catch (Exception e) {
52         System.out.println(e);
53     }
54 }
55
56 /**
57  * Parametre olarak girilen baslangic ve bitis indexlerinin arasinda bulunan tum Course l
58  */
59 @Test
60 void getByRange() {
61     System.out.println("\n\t----- PART1_TEST <Course>-----\n");
62     try{
63         System.out.println("\n\t----- getByRange(start_index, last_index) start_index=
64         LinkedList<Course> plPrint = pl.getByRange( start_index: 10, last_index: 20);
65         Part1Test > listSemesterCourses()
```

Run Part1Test

Test Results

- Part1Test
- listSemesterCourses()
- getByCode()
- getByRange()

All 3 tests passed - 31ms

----- PART1_TEST <Course>-----

----- listSemesterCourses(semester) semester=> 2 -----

| Semester | Course Code | Course Title | ECTS | C | GTU | C | H+I+L |
|----------|-------------|---------------------------------|------|---|-----|-----|-------|
| 2 | XXXX XXXX | Teknik Olmayan Se | 2 | 1 | 2 | 0 | 0+0 |
| 2 | CSE 102 | Computer Programming | 8 | 4 | 4 | 0 | 0+0 |
| 2 | CSE 108 | Computer Programming Laboratory | 2 | 1 | 0 | 0 | 0+2 |
| 2 | MATH 102 | Calculus II | 7 | 5 | 5 | 0 | 0+0 |
| 2 | PHYS 122 | Physics II | 6 | 4 | 3 | 0 | 0+0 |
| 2 | PHYS 152 | Physics Laboratory II | 1 | 1 | 0 | 0+2 | |
| 2 | SSIR 102 | Principles Of Atat | 2 | 2 | 2 | 0 | 0+0 |
| 2 | TUR 102 | Turkish II | 2 | 2 | 2 | 0 | 0+0 |

Tests Passed: 3 passed (4 minutes ago)

41:10 CRLF: UTF-8: 04:16

getByRangeTest

HW3Project [C:\Users\Asus\Desktop\intelliProject\HW3Project] - ...test\HW3\Part1Test.java [HW3Project] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 Part1Test

Part1Test.java

```
46 System.out.println(titles);
47 for(int i = 0; i < plPrint.size(); i++) {
48     assertEquals( expected: 2,plPrint.get(i).getSemester());
49     System.out.println(plPrint.get(i).toString());
50 }
51 } catch (Exception e) {
52     System.out.println(e);
53 }
54 }
55
56 /**
57  * Parametre olarak girilen baslangic ve bitis indexlerinin arasinda bulunan tum Course lari listele
58  */
59 @Test
60 void getByRange() {
61     System.out.println("\n\t----- PART1_TEST <Course>-----\n");
62     try{
63         System.out.println("\n\t----- getByRange(start_index, last_index) start_index=> 10, last_index=> 20
64         LinkedList<Course> plPrint = pl.getByRange( start_index: 10, last_index: 20);
65         System.out.println(titles);
66         for(int i = 0; i < plPrint.size(); i++)
67             System.out.println(plPrint.get(i).toString());
68         assertEquals( expected: 10,plPrint.size());
69     } catch (Exception e) {
70         System.out.println(e);
71     }
72 }
73 }
74 }
```

Run Part1Test

Test Results

- Part1Test
- listSemesterCourses()
- getByCode()
- getByRange()

All 3 tests passed - 31ms

----- PART1_TEST <Course>-----

----- getByRange(start_index, last_index) start_index=> 10, last_index=> 20 -----

| Semester | Course Code | Course Title | ECTS | C | GTU | C | H+I+L |
|----------|-------------|-------------------------------------|------|---|-----|-----|-------|
| 2 | CSE 108 | Computer Programming Laboratory | 2 | 1 | 0 | 0 | 0+0 |
| 2 | MATH 102 | Calculus II | 7 | 5 | 5 | 0 | 0+0 |
| 2 | PHYS 122 | Physics II | 6 | 4 | 3 | 0 | 0+0 |
| 2 | PHYS 152 | Physics Laboratory II | 1 | 1 | 0 | 0+2 | |
| 2 | SSIR 102 | Principles Of Atat | 2 | 2 | 2 | 0 | 0+0 |
| 2 | TUR 102 | Turkish II | 2 | 2 | 2 | 0 | 0+0 |
| 3 | CSE 241 | Object Oriented Programming | 9 | 5 | 3 | 2 | 0+0 |
| 3 | CSE 211 | Discrete Mathematics | 6 | 3 | 3 | 0 | 0+0 |
| 3 | CSE 231 | Circuits And Electronics | 8 | 4 | 4 | 0 | 0+0 |
| 3 | CSE 233 | Circuits And Electronics Laboratory | 2 | 1 | 0 | 0 | 0+0 |

Tests Passed: 3 passed (5 minutes ago)

41:10 CRLF: UTF-8: 04:17

3.1.2.2 Part2

disableTest

The screenshot shows the IntelliJ IDEA IDE with the 'Part2Test.java' file open. The code defines a 'disableTest' method that calls 'p2.disable(index 2)' and 'p2.showDisable()'. The 'showDisable()' method prints the state of the 'Test_List' after disabling an element. The 'Test Results' window shows that all tests passed, including 'enable()', 'disable()', and 'showDisable()'. The output of 'showDisable()' shows the 'Test_List' with the element at index 2 (Akaminko) disabled.

```
43 System.out.println("ndisable(2)");
44 p2.disable(index 2);
45 System.out.println("disable(2)");
46 p2.disable(index 2);
47 assertEquals( expected: 4,p2.size()); // assertEquals ile size lari test ettim
48
49 print();
50 System.out.println("-----\n");
51
52 /**
53  * Liste uzerinde index degerine gore disable edilmis verilerin bir baska disable ed
54  * listeye gecip gecmedigini test etmeye yarayan unitTest metodudur.
55  */
56
57 @Test
58 void showDisable() {
59     System.out.println("-----\n");
60     System.out.println("nSHOWDISABLE_TEST");
61     print();
62     System.out.println("ndisable(1)");
63     p2.disable(index 1);
64     assertEquals( expected: 5,p2.size()); // assertEquals ile size lari test ettim
65     System.out.println("n ----- Disable are those (showDisable) -----n");
66     p2.showDisable();
67
68     print();
69     System.out.println("-----\n");
70
71 /**
72  * Liste uzerinden disable edilmis olan bir verinin enable edilmesi sonucu,
73  * disable edilmislerin bulunduugu listeden ana listede eski konumuna getirmesini tes
74  */
75
76 @Test
77 void enable() {
78     Part2Test p2 = new Part2Test();
79     p2.showDisable();
80 }
```

Test Results:

- Test Results: 25ms
- Part2Test: 25ms
- enable(): 22ms
- disable(): 1ms
- showDisable(): 2ms

All 3 tests passed - 25ms

DISABLE_TEST

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

disable(2)

disable(2)

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Akaminko
- 3 - Alex De Souza

Tests Passed: 3 passed (moments ago)

showDisableTest

The screenshot shows the IntelliJ IDEA IDE with the 'Part2Test.java' file open. The code defines a 'showDisableTest' method that calls 'p2.showDisable()' and 'p2.enable()'. The 'showDisable()' method prints the state of the 'Test_List' after disabling an element. The 'Test Results' window shows that all tests passed, including 'enable()', 'disable()', and 'showDisable()'. The output of 'showDisable()' shows the 'Test_List' with the element at index 2 (Akaminko) disabled.

```
46 p2.disable(index 2);
47 assertEquals( expected: 4,p2.size()); // assertEquals ile size lari test ettim
48
49 print();
50 System.out.println("-----\n");
51
52 /**
53  * Liste uzerinde index degerine gore disable edilmis verilerin bir baska disable ed
54  * listeye gecip gecmedigini test etmeye yarayan unitTest metodudur.
55  */
56
57 @Test
58 void showDisable() {
59     System.out.println("-----\n");
60     System.out.println("nSHOWDISABLE_TEST");
61     print();
62     System.out.println("ndisable(1)");
63     p2.disable(index 1);
64     assertEquals( expected: 5,p2.size()); // assertEquals ile size lari test ettim
65     System.out.println("n ----- Disable are those (showDisable) -----n");
66     p2.showDisable();
67
68     print();
69     System.out.println("-----\n");
70
71 /**
72  * Liste uzerinden disable edilmis olan bir verinin enable edilmesi sonucu,
73  * disable edilmislerin bulunduugu listeden ana listede eski konumuna getirmesini tes
74  */
75
76 @Test
77 void enable() {
78     System.out.println("-----\n");
79     System.out.println("nENABLE_TEST");
80     print();
81 }
```

Test Results:

- Test Results: 25ms
- Part2Test: 25ms
- enable(): 22ms
- disable(): 1ms
- showDisable(): 2ms

All 3 tests passed - 25ms

SHOWDISABLE_TEST

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

disable(1)

----- Disable are those (showDisable) -----

1 -> Ali Lukunku

Test_List

- 0 - Muhittin
- 1 - Ayten
- 2 - Pakize
- 3 - Akaminko
- 4 - Alex De Souza

Tests Passed: 3 passed (4 minutes ago)

enableTest

The screenshot shows the IntelliJ IDEA IDE with the HW3Project open. The file Part2Test.java is selected, and the enable() method is being edited. The method contains several assertions and print statements. The Run window on the right shows the test results for Part2Test, indicating that all 3 tests passed in 25ms. The output of the test is displayed in the Run window, showing the Test_List and the output of the enable() method.

```
74 // disable edilmislerin bulunduugu listeden ana listede eski konumuna getirmesini
75 //
76 @Test
77 void enable() {
78     System.out.println("-----");
79     System.out.println("nENABLE_TEST");
80     print();
81     System.out.println("ndisable(2)");
82     p2.disable( index 2);
83     System.out.println("disable(2)");
84     p2.disable( index 2);
85     assertEquals( expected: 4,p2.size()); // assertEquals ile size leri test ettim
86     System.out.println("n ----- Disable are those (showDisable) -----n");
87     p2.showDisable();
88
89     print();
90
91     System.out.println("nenable(2)");
92     p2.enable( index 2);
93     assertEquals( expected: 5,p2.size()); // assertEquals ile size leri test ettim
94     print();
95
96     System.out.println("n ----- Disable are those (showDisable) -----n");
97     p2.showDisable();
98
99     System.out.println("nenable(3)");
100    p2.enable( index 3);
101    assertEquals( expected: 6,p2.size()); // assertEquals ile size leri test ettim
102    print();
103
104    System.out.println("n ----- Disable are those (showDisable) -----n");
105    p2.showDisable();
106
107    System.out.println("-----n");
108 }
```

Test Results: All 3 tests passed - 25ms

ENABLE_TEST

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

disable(2)

disable(2)

----- Disable are those (showDisable) -----

2 -> Ayten

3 -> Pakize

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Akaminko
- 3 - Alex De Souza

enable(2)

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

The screenshot shows the IntelliJ IDEA IDE with the HW3Project open. The file Part2Test.java is selected, and the enable() method is being edited. The method contains several assertions and print statements. The Run window on the right shows the test results for Part2Test, indicating that all 3 tests passed in 25ms. The output of the test is displayed in the Run window, showing the Test_List and the output of the enable() method.

```
74 // disable edilmislerin bulunduugu listeden ana listede eski konumuna getirmesini
75 //
76 @Test
77 void enable() {
78     System.out.println("-----");
79     System.out.println("nENABLE_TEST");
80     print();
81     System.out.println("ndisable(2)");
82     p2.disable( index 2);
83     System.out.println("disable(2)");
84     p2.disable( index 2);
85     assertEquals( expected: 4,p2.size()); // assertEquals ile size leri test ettim
86     System.out.println("n ----- Disable are those (showDisable) -----n");
87     p2.showDisable();
88
89     print();
90
91     System.out.println("nenable(2)");
92     p2.enable( index 2);
93     assertEquals( expected: 5,p2.size()); // assertEquals ile size leri test ettim
94     print();
95
96     System.out.println("n ----- Disable are those (showDisable) -----n");
97     p2.showDisable();
98
99     System.out.println("nenable(3)");
100    p2.enable( index 3);
101    assertEquals( expected: 6,p2.size()); // assertEquals ile size leri test ettim
102    print();
103
104    System.out.println("n ----- Disable are those (showDisable) -----n");
105    p2.showDisable();
106
107    System.out.println("-----n");
108 }
```

Test Results: All 3 tests passed - 25ms

enable(2)

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Akaminko
- 4 - Alex De Souza

----- Disable are those (showDisable) -----

3 -> Pakize

enable(3)

Test_List

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

----- Disable are those (showDisable) -----

Disable courses are not available.

3.1.2.3 Part3

addTest

The screenshot shows the IntelliJ IDEA IDE with the `Part3Test.java` file open. The file contains two test methods: `size()` and `add()`. The `size()` method prints the size of the list and asserts it is 6. The `add()` method adds five elements to the list and asserts the size is 6. The test results pane on the right shows that all 7 tests passed in 32ms. The test output shows the list contents: Muhittin, Ali Lukunku, Ayten, Pakize, Akaminko, and Alex De Souza.

```
45 // String tipindeki listenin icindeki elemanların sayısını verir.
46 //
47
48 void size() {
49     System.out.println("\n\t----- size() -----");
50     System.out.println("size => " + p3String.size());
51 }
52
53 // Object tipinde veriyi parametrik olarak, listeye eklenip eklenmediğini test eden unitTest metodudur.
54 //
55 @Test
56 void add() {
57     System.out.println("-----");
58     System.out.println("\nadd(E e) _TEST\n");
59     p3String.add("Muhittin");
60     p3String.add("Ali Lukunku");
61     p3String.add("Ayten");
62     p3String.add("Pakize");
63     p3String.add("Akaminko");
64     p3String.add("Alex De Souza");
65     printString();
66     assertEquals( expected: 6, p3String.size());
67     size();
68     System.out.println("-----");
69 }
70
71 // Index ve Object tipinde veriyi parametrik olarak, liste üzerinde eklenip eklenmediğini test eden unitTest metodudur.
72 //
73 @Test
74 void addI() {
75     System.out.println("-----");
76     System.out.println("\nadd(index, E e) _TEST\n");
77 }
```

Test Results: All 7 tests passed - 32ms

add(E e) _TEST

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

----- size() -----

size => 6

The screenshot shows the IntelliJ IDEA IDE with the `Part3Test.java` file open. The file contains two test methods: `size()` and `addI()`. The `size()` method prints the size of the list and asserts it is 6. The `addI()` method adds five elements to the list and asserts the size is 6. The test results pane on the right shows that all 7 tests passed in 32ms. The test output shows the list contents: Muhittin, Ali Lukunku, Ayten, Pakize, Akaminko, and Alex De Souza.

```
65 p3String.add("Alex De Souza");
66 printString();
67 assertEquals( expected: 6, p3String.size());
68 size();
69 System.out.println("-----");
70 }
71
72 // Index ve Object tipinde veriyi parametrik olarak, liste üzerinde eklenip eklenmediğini test eden unitTest metodudur.
73 //
74 @Test
75 void addI() {
76     System.out.println("-----");
77     System.out.println("\nadd(index, E e) _TEST\n");
78     p3String.add("Muhittin");
79     p3String.add("Ali Lukunku");
80     p3String.add("Ayten");
81     p3String.add("Pakize");
82     p3String.add("Akaminko");
83     p3String.add("Alex De Souza");
84     printString();
85     System.out.println("\nadd(3, Hayrullah)");
86     p3String.add( index: 3, value: "Hayrullah");
87     printString();
88     assertEquals( expected: 7, p3String.size());
89     size();
90     System.out.println("-----");
91 }
92
93 // Listenin son elemanını silip silmediğini test eden unitTest metodudur.
94 //
95 @Test
96 void remove() {
97     System.out.println("-----");
98     System.out.println("\nremove() _TEST\n");
99     p3String.remove("Alex De Souza");
100     printString();
101     assertEquals( expected: 5, p3String.size());
102     size();
103     System.out.println("-----");
104 }
```

Test Results: All 7 tests passed - 32ms

add(index, E e) _TEST

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

add(3, Hayrullah)

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Hayrullah
- 4 - Pakize
- 5 - Akaminko
- 6 - Alex De Souza

----- size() -----

size => 7

removeTest

The screenshot shows the IntelliJ IDEA interface with the `Part3Test.java` file open. The `remove()` method is highlighted, which is a JUnit test. The test prints the initial list, removes the element "Alex De Souza", and then asserts that the list size is 5. The Run window on the right shows the test results, indicating that all 7 tests passed in 32ms. The output of the test is displayed in the console, showing the list state before and after the removal.

```
100 void remove() {
101     System.out.println("-----");
102     System.out.println("remove() _TEST\n");
103     p3String.add("Muhittin");
104     p3String.add("Ali Lukunku");
105     p3String.add("Ayten");
106     p3String.add("Pakize");
107     p3String.add("Akaminko");
108     p3String.add("Alex De Souza");
109     printString();
110     System.out.println("remove() => " + p3String.remove());
111     printString();
112     assertEquals( expected: 5, p3String.size());
113     size();
114     System.out.println("-----");
115 }
116
117 /**
118  * Object tipinde veriyi parametrik olarak, listeden silip silmedigini test eden unitTest
119  */
120 @Test
121 void remove1() {
122     System.out.println("-----");
123 }
```

Test Results:

- Test Results (32ms)
- Part3Test (32ms)
- remove() (32ms)
- nextInSemester()
- add()
- add1()
- add1()

All 7 tests passed - 32ms

remove() _TEST

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

remove() => Alex De Souza

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko

size()

size => 5

The screenshot shows the IntelliJ IDEA interface with the `Part3Test.java` file open. The `remove1()` and `remove2()` methods are highlighted. The `remove1()` method is a JUnit test that removes the element "Akaminko" and asserts that the list size is 5. The `remove2()` method is a JUnit test that removes the element "Akaminko" and asserts that the list size is 5. The Run window on the right shows the test results, indicating that all 7 tests passed in 32ms. The output of the tests is displayed in the console, showing the list state before and after the removal.

```
121 void remove1() {
122     System.out.println("-----");
123     System.out.println("remove(E value) _TEST\n");
124     p3String.add("Muhittin");
125     p3String.add("Ali Lukunku");
126     p3String.add("Ayten");
127     p3String.add("Pakize");
128     p3String.add("Akaminko");
129     p3String.add("Alex De Souza");
130     printString();
131     System.out.println("remove(\"Akaminko\") => " + p3String.remove( value: "Akaminko"));
132     printString();
133     assertEquals( expected: 5, p3String.size());
134     size();
135     System.out.println("-----");
136 }
137
138 /**
139  * Index degerini parametrik olarak, listeden silip silmedigini test eden unitTest
140  */
141 @Test
142 void remove2() {
143     System.out.println("-----");
144 }
```

Test Results:

- Test Results (32ms)
- Part3Test (32ms)
- remove1() (32ms)
- remove2() (32ms)
- nextInSemester()
- add()
- add1()
- add1()
- next()
- remove1()
- remove2()

All 7 tests passed - 32ms

remove(E value) _TEST

Test_StringList

- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Akaminko
- 5 - Alex De Souza

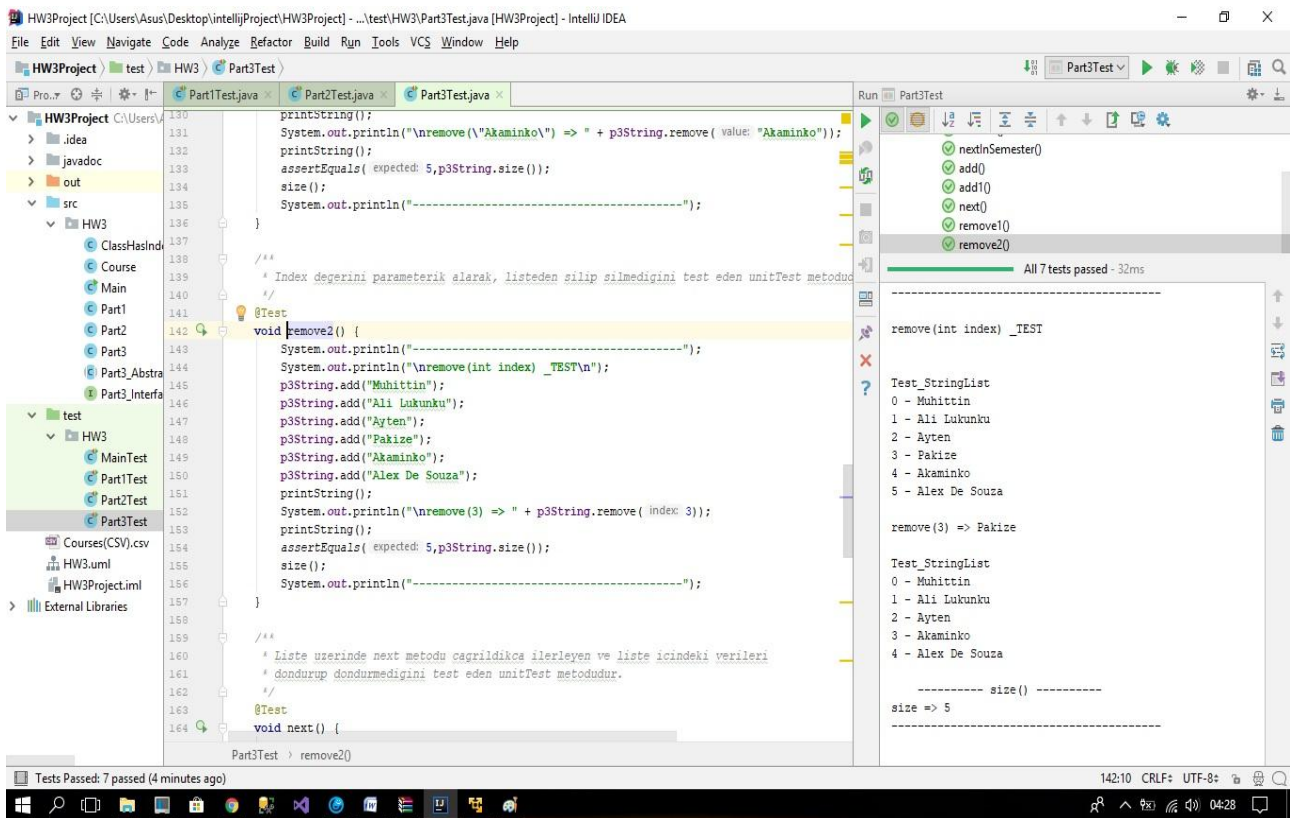
remove("Akaminko") => true

Test_StringList

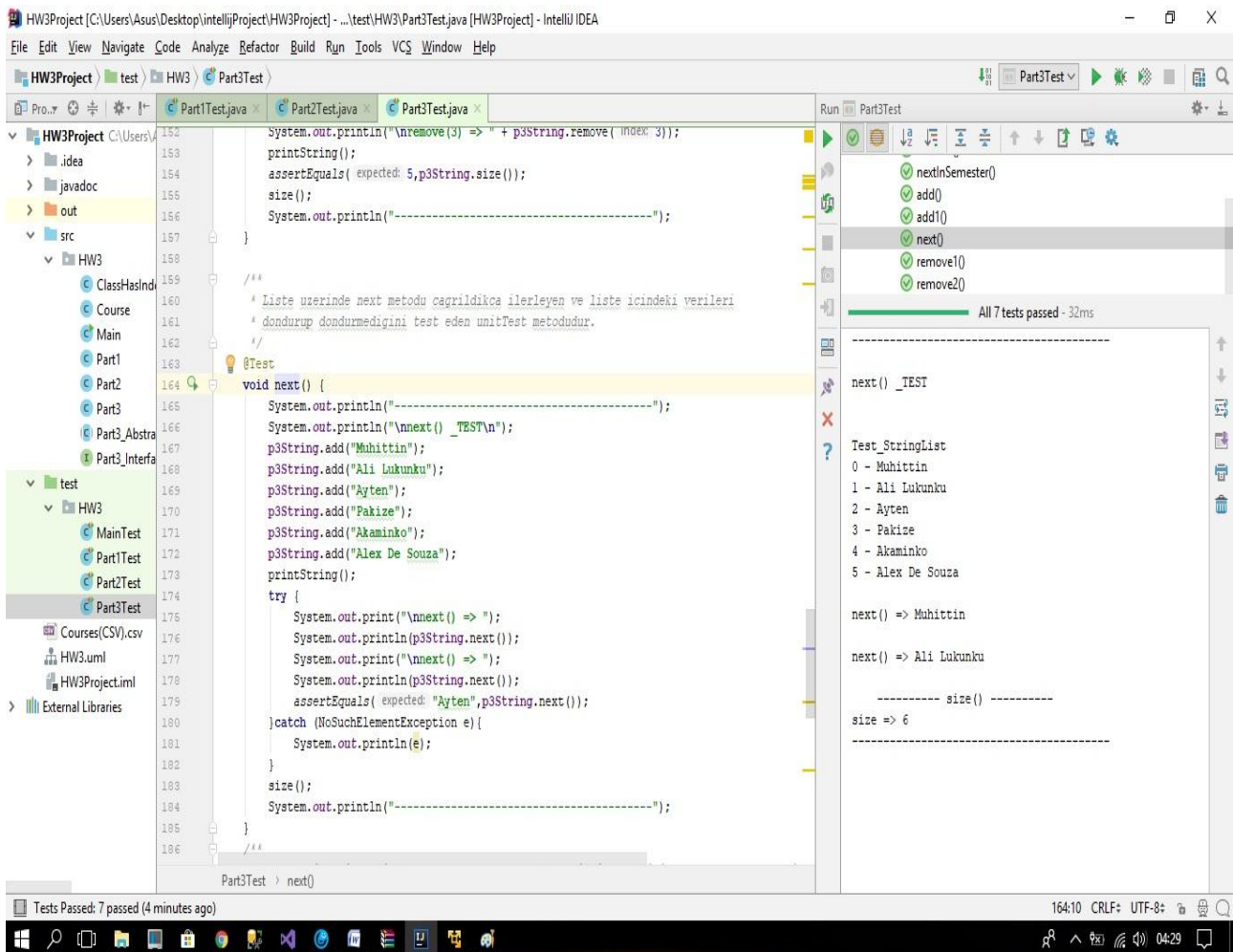
- 0 - Muhittin
- 1 - Ali Lukunku
- 2 - Ayten
- 3 - Pakize
- 4 - Alex De Souza

size()

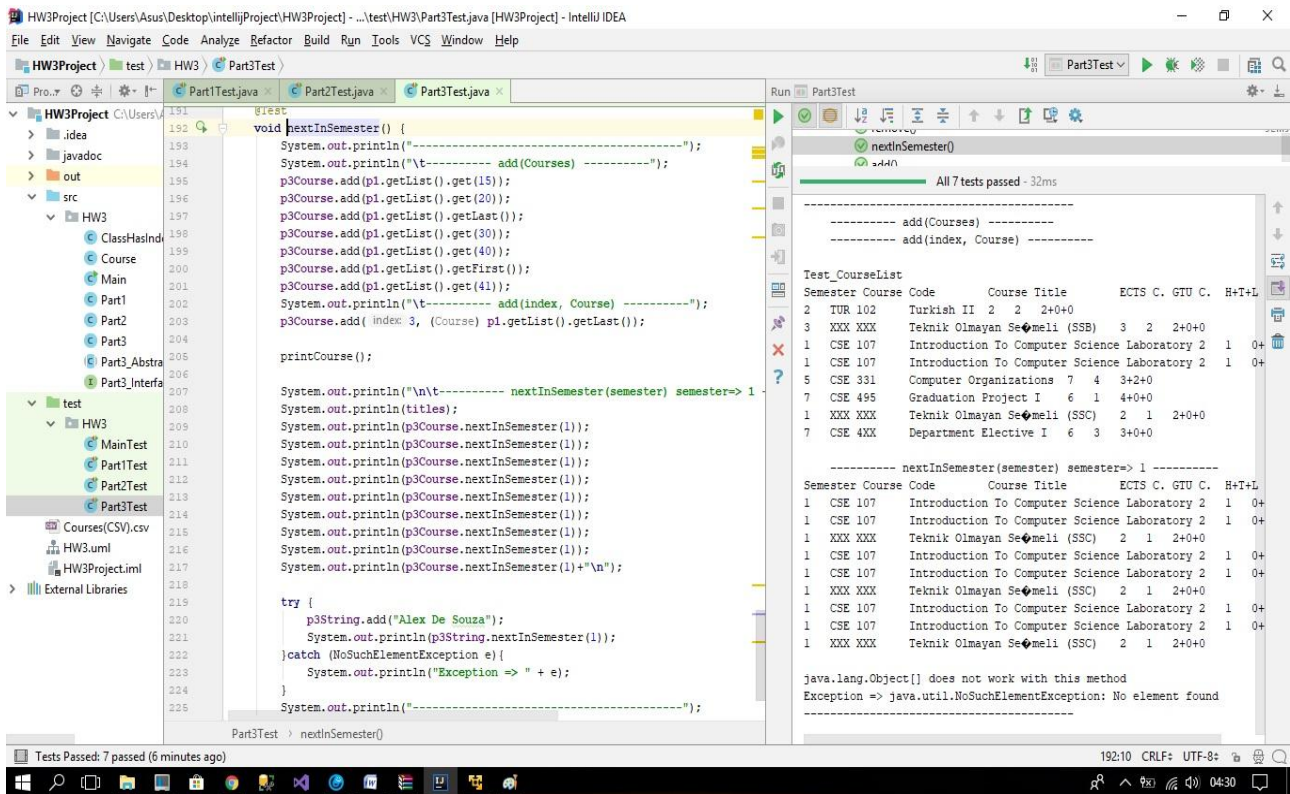
size => 5



nextTest

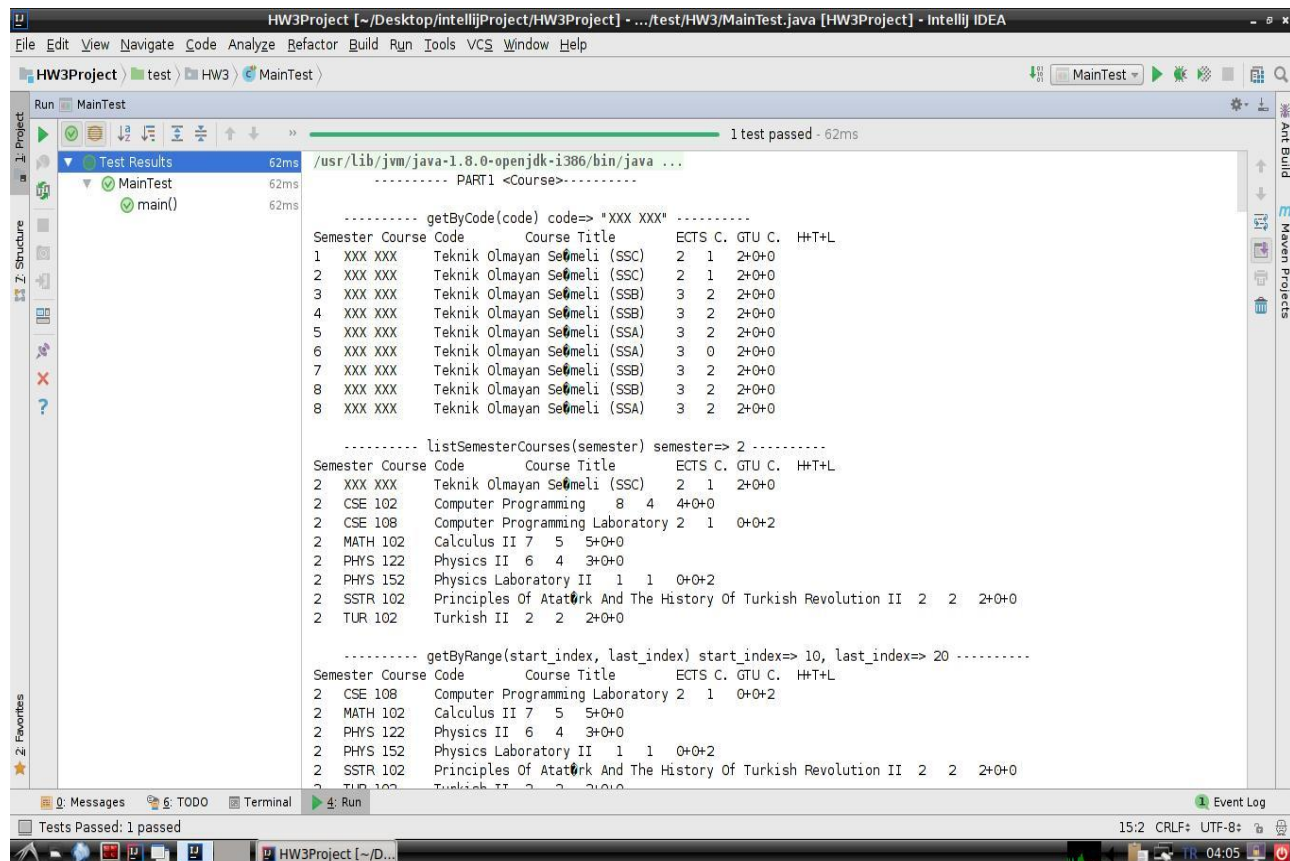


nextInSemester



3.2 Running Results

Mainde yazılan test amaçlı tüm metodların çalıştığını gösteren ekran çıktıları.



```
HW3Project [~/Desktop/intelijProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest
Run MainTest
1 test passed - 62ms

Test Results 62ms
  MainTest 62ms
    main() 62ms

----- PART2 - Generic <E> ( This Example <Course> ) -----

----- GTU Computer Engineering Course -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 XXX XXX Teknik Olmayan Seömeli (SSC) 2 1 2+0+0
1 -> 1 CSE 101 Introduction To Computer Engineering 8 3 3+0+0
2 -> 1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
3 -> 1 MATH 101 Calculus I 7 5 5+0+0
4 -> 1 PHYS 121 Physics I 6 4 3+0+0
5 -> 1 PHYS 151 Physics Laboratory I 1 1 0+0+2
6 -> 1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
7 -> 1 TUR 101 Turkish I 2 2 2+0+0
8 -> 2 XXX XXX Teknik Olmayan Seömeli (SSC) 2 1 2+0+0
9 -> 2 CSE 102 Computer Programming 8 4 4+0+0
10 -> 2 CSE 108 Computer Programming Laboratory 2 1 0+0+2
11 -> 2 MATH 102 Calculus II 7 5 5+0+0
12 -> 2 PHYS 122 Physics II 6 4 3+0+0
13 -> 2 PHYS 152 Physics Laboratory II 1 1 0+0+2
14 -> 2 SSTR 102 Principles Of Atatürk And The History Of Turkish Revolution II 2 2 2+0+0
15 -> 2 TUR 102 Turkish II 2 2 2+0+0
16 -> 3 CSE 241 Object Oriented Programming 9 5 3+2+0
17 -> 3 CSE 211 Discrete Mathematics 6 3 3+0+0
18 -> 3 CSE 231 Circuits And Electronics 8 4 4+0+0
19 -> 3 CSE 233 Circuits And Electronics Laboratory 2 1 0+0+2
20 -> 3 XXX XXX Teknik Olmayan Seömeli (SSB) 3 2 2+0+0
21 -> 3 EN 111 English For Business Life 2 2 2+0+0
22 -> 4 CSE 222 Data Structures And Algorithms 9 5 4+2+0
23 -> 4 MATH 217 Linearr Algebra And Differential Equations 8 5 4+2+0
24 -> 4 CSE 232 Logic Circuits And Design 6 3 3+0+0
25 -> 4 CSE 234 Logic Circuits And Design Laboratory 2 1 0+0+2
26 -> 4 XXX XXX Teknik Olmayan Seömeli (SSB) 3 2 2+0+0
27 -> 4 EN 112 Academic English 2 2 2+0+0
28 -> 5 CSE 343 Software Engineering 8 4 3+2+0
29 -> 5 CSE 341 Programming Languages 6 3 3+0+0

0: Messages 6: TODO Terminal 4: Run
Tests Passed: 1 passed (2 minutes ago) 15:2 CRLF+ UTF-8: 04:06
```

```
HW3Project [~/Desktop/intelijProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest
Run MainTest
1 test passed - 62ms

Test Results 62ms
  MainTest 62ms
    main() 62ms

----- disable(index) index=> 0, 5, 13 -----
----- showDisable() -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 XXX XXX Teknik Olmayan Seömeli (SSC) 2 1 2+0+0
6 -> 1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
15 -> 2 TUR 102 Turkish II 2 2 2+0+0

----- GTU Computer Engineering Course -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 CSE 101 Introduction To Computer Engineering 8 3 3+0+0
1 -> 1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
2 -> 1 MATH 101 Calculus I 7 5 5+0+0
3 -> 1 PHYS 121 Physics I 6 4 3+0+0
4 -> 1 PHYS 151 Physics Laboratory I 1 1 0+0+2
5 -> 1 TUR 101 Turkish I 2 2 2+0+0
6 -> 2 XXX XXX Teknik Olmayan Seömeli (SSC) 2 1 2+0+0
7 -> 2 CSE 102 Computer Programming 8 4 4+0+0
8 -> 2 CSE 108 Computer Programming Laboratory 2 1 0+0+2
9 -> 2 MATH 102 Calculus II 7 5 5+0+0
10 -> 2 PHYS 122 Physics II 6 4 3+0+0
11 -> 2 PHYS 152 Physics Laboratory II 1 1 0+0+2
12 -> 2 SSTR 102 Principles Of Atatürk And The History Of Turkish Revolution II 2 2 2+0+0
13 -> 3 CSE 241 Object Oriented Programming 9 5 3+2+0
14 -> 3 CSE 211 Discrete Mathematics 6 3 3+0+0
15 -> 3 CSE 231 Circuits And Electronics 8 4 4+0+0
16 -> 3 CSE 233 Circuits And Electronics Laboratory 2 1 0+0+2
17 -> 3 XXX XXX Teknik Olmayan Seömeli (SSB) 3 2 2+0+0
18 -> 3 EN 111 English For Business Life 2 2 2+0+0
19 -> 4 CSE 222 Data Structures And Algorithms 9 5 4+2+0
20 -> 4 MATH 217 Linearr Algebra And Differential Equations 8 5 4+2+0
21 -> 4 CSE 232 Logic Circuits And Design 6 3 3+0+0
22 -> 4 CSE 234 Logic Circuits And Design Laboratory 2 1 0+0+2
23 -> 4 XXX XXX Teknik Olmayan Seömeli (SSB) 3 2 2+0+0

0: Messages 6: TODO Terminal 4: Run
Tests Passed: 1 passed (2 minutes ago) 15:2 CRLF+ UTF-8: 04:07
```


HW3Project [~/Desktop/intelijProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest

Run MainTest 1 test passed - 62ms

Test Results 62ms

- MainTest 62ms
- main() 62ms

```
----- disable(index) index=> 21, 7, 17 -----
----- showDisable() -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1 2+0+0
6 -> 1 SSTR 101 Principles Of Atat0rk And The History Of Turkish Revolution I 2 2 2+0+0
15 -> 2 TUR 102 Turkish II 2 2 2+0+0
24 -> 4 CSE 232 Logic Circuits And Design 6 3 3+0+0
9 -> 2 CSE 102 Computer Programming 8 4 4+0+0
21 -> 3 EN 111 English For Business Life 2 2 2+0+0

----- enable(index) index=> 6, 24, 21 -----
----- showDisable() -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1 2+0+0
15 -> 2 TUR 102 Turkish II 2 2 2+0+0
9 -> 2 CSE 102 Computer Programming 8 4 4+0+0

----- GTU Computer Engineering Course -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 CSE 101 Introduction To Computer Engineering 8 3 3+0+0
1 -> 1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
2 -> 1 MATH 101 Calculus I 7 5 5+0+0
3 -> 1 PHYS 121 Physics I 6 4 3+0+0
4 -> 1 PHYS 151 Physics Laboratory I 1 1 0+0+2
5 -> 1 SSTR 101 Principles Of Atat0rk And The History Of Turkish Revolution I 2 2 2+0+0
6 -> 1 TUR 101 Turkish I 2 2 2+0+0
7 -> 2 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1 2+0+0
8 -> 2 CSE 108 Computer Programming Laboratory 2 1 0+0+2
9 -> 2 MATH 102 Calculus II 7 5 5+0+0
10 -> 2 PHYS 122 Physics II 6 4 3+0+0
11 -> 2 PHYS 152 Physics Laboratory II 1 1 0+0+2
12 -> 2 SSTR 102 Principles Of Atat0rk And The History Of Turkish Revolution II 2 2 2+0+0
```

0: Messages 6: TODO Terminal 4: Run 1: Event Log

Tests Passed: 1 passed (3 minutes ago) 15:2 CRLF: UTF-8: 04:07

HW3Project [~/Desktop/intelijProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest

Run MainTest 1 test passed - 62ms

Test Results 62ms

- MainTest 62ms
- main() 62ms

```
----- enable(index) index=> 0, 15, 9 -----
----- showDisable() -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
Disable courses are not available.
----- GTU Computer Engineering Course -----
Index Semester Course Code Course Title ECTS C. GTU C. H+T+L
0 -> 1 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1 2+0+0
1 -> 1 CSE 101 Introduction To Computer Engineering 8 3 3+0+0
2 -> 1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
3 -> 1 MATH 101 Calculus I 7 5 5+0+0
4 -> 1 PHYS 121 Physics I 6 4 3+0+0
5 -> 1 PHYS 151 Physics Laboratory I 1 1 0+0+2
6 -> 1 SSTR 101 Principles Of Atat0rk And The History Of Turkish Revolution I 2 2 2+0+0
7 -> 1 TUR 101 Turkish I 2 2 2+0+0
8 -> 2 XXX XXX Teknik Olmayan Se0meli (SSC) 2 1 2+0+0
9 -> 2 CSE 102 Computer Programming 8 4 4+0+0
10 -> 2 CSE 108 Computer Programming Laboratory 2 1 0+0+2
11 -> 2 MATH 102 Calculus II 7 5 5+0+0
12 -> 2 PHYS 122 Physics II 6 4 3+0+0
13 -> 2 PHYS 152 Physics Laboratory II 1 1 0+0+2
14 -> 2 SSTR 102 Principles Of Atat0rk And The History Of Turkish Revolution II 2 2 2+0+0
15 -> 3 CSE 241 Object Oriented Programming 9 5 3+2+0
16 -> 2 TUR 102 Turkish II 2 2 2+0+0
17 -> 3 CSE 211 Discrete Mathematics 6 3 3+0+0
18 -> 3 CSE 231 Circuits And Electronics 8 4 4+0+0
19 -> 3 CSE 233 Circuits And Electronics Laboratory 2 1 0+0+2
20 -> 3 XXX XXX Teknik Olmayan Se0meli (SSB) 3 2 2+0+0
21 -> 3 EN 111 English For Business Life 2 2 2+0+0
22 -> 4 CSE 222 Data Structures And Algorithms 9 5 4+2+0
23 -> 4 MATH 217 Linearr Algebra and Differential Equations 8 5 4+2+0
24 -> 4 CSE 234 Logic Circuits And Design Laboratory 2 1 0+0+2
25 -> 4 XXX XXX Teknik Olmayan Se0meli (SSB) 3 2 2+0+0
26 -> 4 CSE 232 Logic Circuits And Design 6 3 3+0+0
27 -> 4 EN 112 Academic English 2 2 2+0+0
```

0: Messages 6: TODO Terminal 4: Run 1: Event Log

Tests Passed: 1 passed (4 minutes ago) 15:2 CRLF: UTF-8: 04:08

```
HW3Project [~/Desktop/intelliJProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest
Run MainTest
1 test passed - 62ms

Test Results 62ms
  MainTest 62ms
    main() 62ms

PART3 Generic <E> ( This Example <Course> )

add(Courses)
add(index, Course)
hasNext() - next()

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 XXX XXX Teknik Olmayan Seimeli (SSC) 2 1 2+0+0
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
2 CSE 102 Computer Programming 8 4 4+0+0

size()
size => 5

remove()
remove(Course)
remove(index) index=> 2
hasNext() - next()

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0

add(Courses)
hasNext() - next()

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
2 TUR 102 Turkish II 2 2 2+0+0
3 XXX XXX Teknik Olmayan Seimeli (SSB) 3 2 2+0+0
1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
5 CSE 331 Computer Organizations 7 4 3+2+0
7 CSE 495 Graduation Project I 6 1 4+0+0

0: Messages 6: TODO Terminal Run Event Log
Tests Passed: 1 passed (4 minutes ago) 15:2 CRLF+ UTF-8+
HW3Project [~/D...
```

```
HW3Project [~/Desktop/intelliJProject/HW3Project] - .../test/HW3/MainTest.java [HW3Project] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW3Project test HW3 MainTest
Run MainTest
1 test passed - 62ms

Test Results 62ms
  MainTest 62ms
    main() 62ms

remove(Course)
remove(index) index=> 2
hasNext() - next()

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0

add(Courses)
hasNext() - next()

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
2 TUR 102 Turkish II 2 2 2+0+0
3 XXX XXX Teknik Olmayan Seimeli (SSB) 3 2 2+0+0
1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
5 CSE 331 Computer Organizations 7 4 3+2+0
7 CSE 495 Graduation Project I 6 1 4+0+0
1 XXX XXX Teknik Olmayan Seimeli (SSC) 2 1 2+0+0
7 CSE 4XX Department Elective I 6 3 3+0+0

nextInSemester(semester) semester=> 1

Semester Course Code Course Title ECTS C. GTU C. H+T+L
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
1 XXX XXX Teknik Olmayan Seimeli (SSC) 2 1 2+0+0
1 MATH 101 Calculus I 7 5 5+0+0
1 SSTR 101 Principles Of Atatürk And The History Of Turkish Revolution I 2 2 2+0+0
1 CSE 107 Introduction To Computer Science Laboratory 2 1 0+0+2
1 XXX XXX Teknik Olmayan Seimeli (SSC) 2 1 2+0+0
1 MATH 101 Calculus I 7 5 5+0+0

Process finished with exit code 0

0: Messages 6: TODO Terminal Run Event Log
Tests Passed: 1 passed (5 minutes ago) 15:2 CRLF+ UTF-8+
HW3Project [~/D...
```

4 Complexity Analysis

4.1 Part1

GetByCode ve ListSemesterCourses: Bu metod içerisinde iterator oluşturularak LinkedList içerisinde search edilmesi ile bir if şartına bağlı olarak return edilecek LinkedList yapısına eleman eklenecektir ve metod sonunda return edilecektir. Bu nedenle iteratorun liste içindeki n kadarlık tüm elemanları tek tek dolanması worst-case durumudur ve linear time $Q(n)$ kadar sürmektedir. Ancak liste içerisinde tek bir elemanın varsayım ve bu elemana bakma durumu best-case durumudur ve constant time $Q(1)$ kadar sürmektedir.

GetByRange: Bu metod içerisinde for döngüsü ile parametreden gelen index değerleri arasında bulunan elemanlar başka bir LinkedList yapısına sahip listeye eklenerek return edecektir. Bu nedenle parametreden gelen index aralığı 0 ile n kadarlık büyük bir diğer aralığı verildiğinde worst-case durumunda linear time $Q(n)$ kadar döngü dönecektir. Ancak parametre değerleri kısa bir aralıkta verilir ise best-case durumunda constant time $Q(1)$ kadar sürecektir.

4.2 Part2

Enable: Bu metod aldığı index parametresi ile disable edilmiş liste üzerinde iterator ile dolanarak if şartına göre disable edilmiş listeden çıkarma işlemi yapmaktadır. Parametrede gönderilen index değeri disable edilmişlerden en sonda n . Indexde olduğunu varsayarsak worst-case durumu linear time $Q(n)$ kadar sürmektedir. Disable edilmişler arasında ilk sıralarda bulunduğu takdirde ise constant-time $Q(1)$ kadar sürecektir.

Disable: Bu metod aldığı index parametresi ile ana liste üzerinde search yapar ve if şartına bağlı olarak o index değerine sahip veriyi siler ve disable edilmişler listesine alır. Bu işlem for döngüsü ile yapılacağından ve index değerine sahip elemanın n . Indexde olduğunu varsayarsak worst-case durumu linear time $Q(n)$ kadar sürmektedir. Ana liste de ilk sıralarda bulunduğu takdirde ise constant-time $Q(1)$ kadar sürecektir.

ShowDisable: Disable edilenler listesi üzerinde bir iterator yardımı ile dolanarak verileri ekrana çıktı olarak verir. Bu nedenle disable edilmişler n kadar olduğunu varsayarsak worst-case durumu linear time $O(n)$ kadar sürmektedir. Disable edilecekler listesinde tek eleman olduğunu varsayarsak best-case durumu constant time $O(1)$ kadar sürecektir.

4.3 Part3

Add(Object): Bu metod arrayin son elemanından sonrasına veriyi ekleyeceğinden içyapısı gereği kapasite ile eleman sayıları aynı olduğu takdirde reallocate metodu çağrılacaktır ve reallocate metodu içyapısı gereği sırasıyla bir arrayden diğer arraye kopyalama işlemi yaptığından add metodu worst-case durumunda linear time $O(n)$ kadar sürecektir. Ancak kapasite ile eleman sayıları aynı olmadığı takdirde best-case durumunda constant time $O(1)$ kadar sürecektir.

Add(index,Object): Bu metod aldığı index değeri ile array üzerinde linear bir search işlemi gerçekleştirir. Bulunan index değerine ait yere parametrede belirtilen object yerleştirilir ve diğer elemanlar bir ilerideki indexe kaydırılır. Lineer search haricinde diğerleri constant time sürdüğünden bu metodun worst-case durumu linear time $O(n)$ kadar sürecektir. Index değerinin arrayin ilk başlarında bulunması ve elemanın yerleştirilmesi durumunda ise constant time $O(1)$ sürmektedir.

Remove(): Arrayin boş olmadığı durumda arrayin son elemanını silen metoddur. Bu nedenle tek bir işlem yapılacaktır ve worst ve best case durumlarında constant time $O(1)$ sürecektir.

Remove(Object): Arrayin içerisinde boş olup olmadığı kontrol edilir array içerişi eğer boş ise best-case durumu constant time $O(1)$ dir. Ancak array boş ise iki durum çıkıyor. Metodda bulunan objenin olup olmaması için bir for döngüsü ve arrayin n kadar elemanının olduğunu var sayarsak linear search yapacaktır. Eğer eleman arrayde yok ise best-case durumunda önceden yapılan linear search işleminden dolayı best-case durumu linear time $O(n)$ kadar sürecektir. Diğer bir yonden array içerisinde aradığımız eleman var ve array sisteminde silme işlemi ya üzerine bir şey yazarak ya da o elemanı es geçip size değerini bir azaltarak yapılır. Böylece bu işlemde de kullanılacak linear

ilerleyen bir for döngüsünden dolayı worst-case durumunda ise her durumda linear time $O(n)$ kadar bir süre geçecektir.

Remove(index): Metodun içerisinde yapılan index kontrollerinden dolayı silinecek index değerinin sıfır ile arrayin eleman sayısı aralığında olmama durumunu temel alırsak worst-case constant time $O(1)$ kadar sürecektir. Eğer parametrede belirtilen index değerinde bir sıkıntı yok ise array üzerinde silme işlemini bir üst kısımda belirttiğim gibi olacağından ve n elemanlı bir array var sayıp n . elemana kadar for döngüsü döneceğinden worst-case durumu linear time $O(n)$ kadar sürecektir.

Next(): Next metodunun yapısı gereği sadece bir sonraki veriyi vereceğinden ve içerisinde herhangi bir döngülük iş olmadığından ya da başka işlemler gerçekleşmediğinden worst ve best case durumlarında constant time $O(1)$ kadar sürecektir.

NextInSemester(semester): Parametre olarak almış olduğu semester değerine göre liste üzerindeki sadece o semester değerine sahip olan verileri veren bu metodun iç yapısında hasNextInSemester adında bir metod çağırılmaktadır. Bu metod yardımcı bir metod olup array üzerinde aldığı semester değerine göre kaç tane bu semester değerine sahip verinin olduğunu ve head-tail değerini belirleyen bir metoddur ve arrayin ilk ve son elemanında aynı semester değerinin olduğunu varsayarsak ve n kadar elemana sahip bir array olduğunu varsayarsak bu yardımcı metodun worst-case durumu linear time $O(n)$ sürer best-case durumu ise parametrede belirtilen semester değerinin sadece iki tane olup ve yan yana olması ile gerçekleşir constant time $O(1)$ kadar sürer. Gelelim nextInSemester metoduna, bu metod içerisinde yardımcı metoddan gelen $O(n)$ lik bir kısım çalışır ayrıca her nextInSemester işleminde head-tail arasındaki parametreden gelen semester bilgisiyle eşleşmeyen verileri es geçeceğimiz için worst-case durumunda bir while döngüsü ile linear time $O(n)$ kadar süre geçecektir. While döngüsüne girmez ise best-case durumunda constant time sürmektedir.