

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

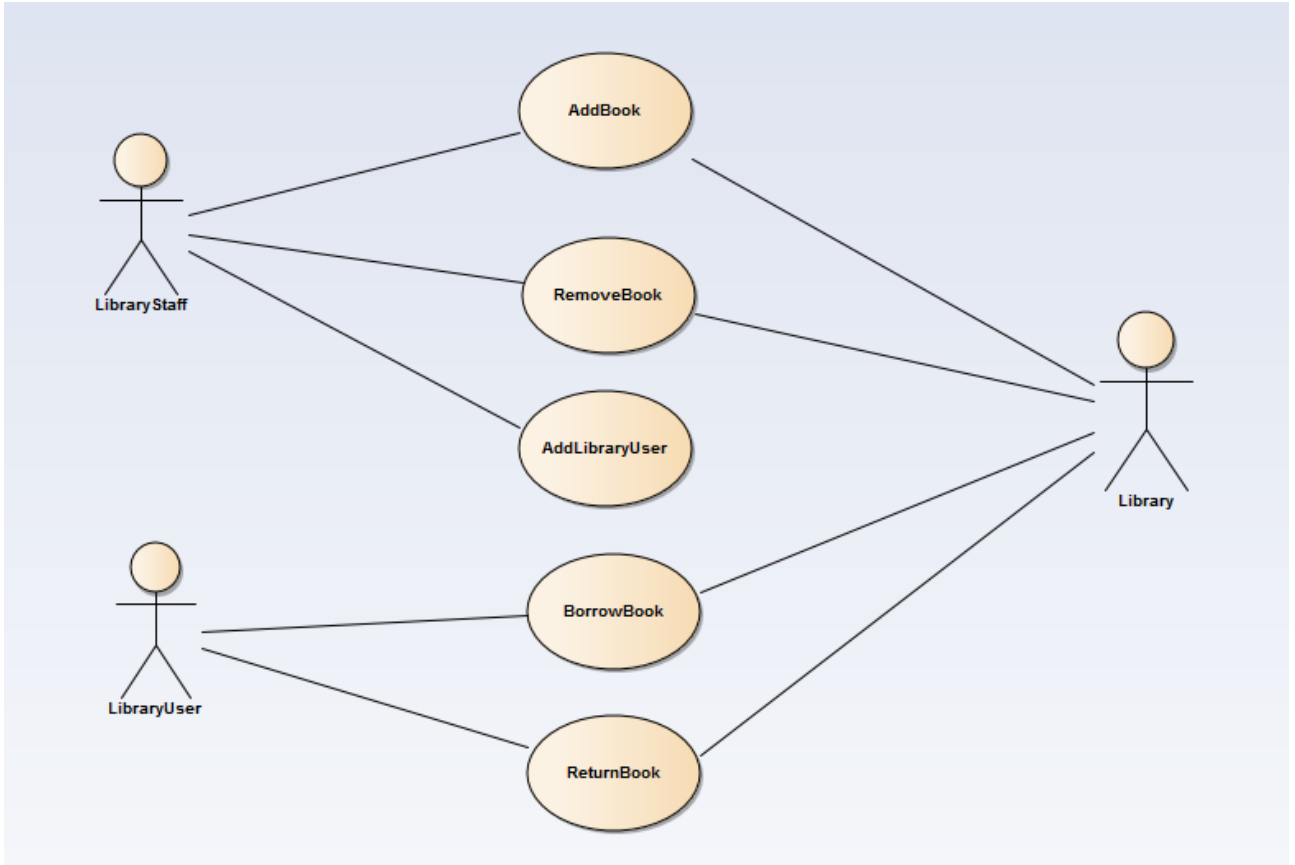
HOMEWORK 01 REPORT

Büşra ARSLAN
131044021

1. System Requirements

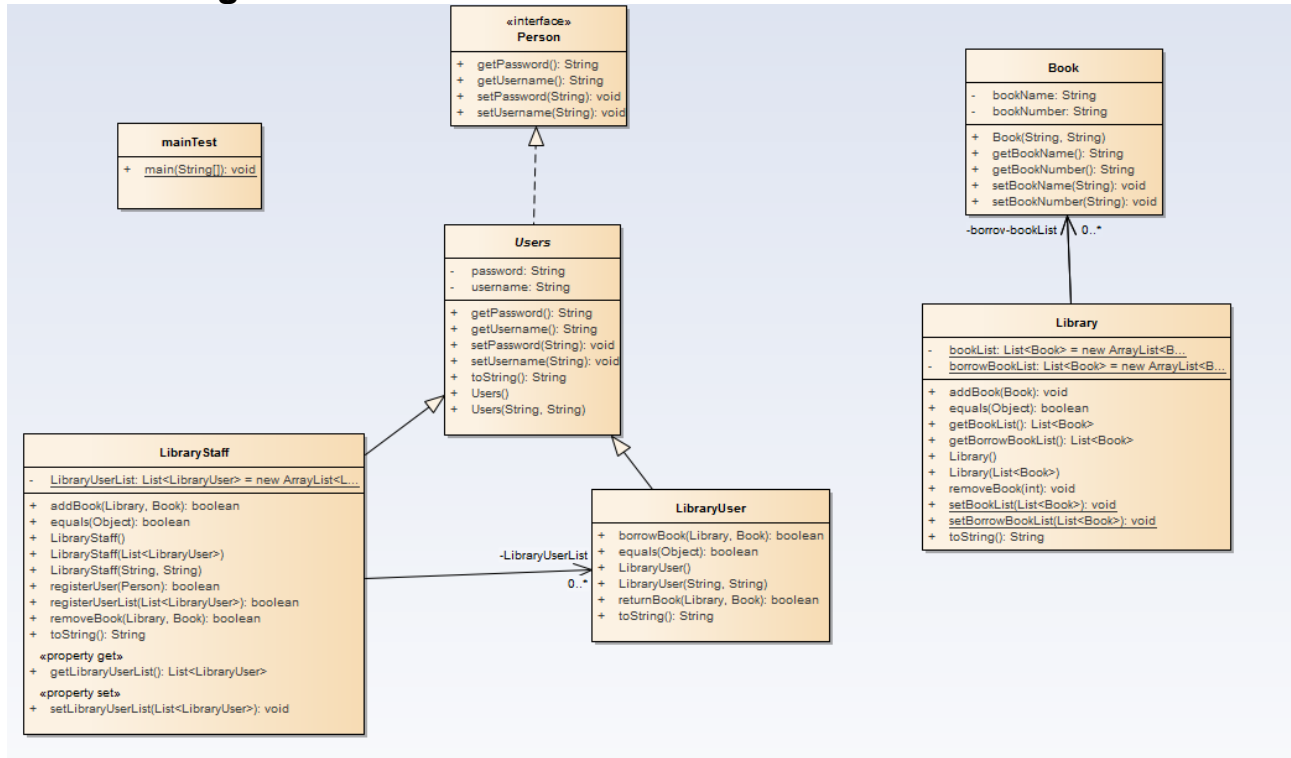
- İnterface bir Person classından implements edilen Users abstract classı kişi username ve password değerlerini alır. Users abstract classı yapılmasının sebebi Persondaki isim ve şifreyi LibraryStaff ve LibraryUser da iki kere implement etme gereksizliğini yapmamak için yapılmıştır.
- Users classında methodlar Override edilmiştir.
- Users abstract classından LibraryUser ve LibraryStaff classları extends edilmektedir.
- LibraryUser classı içerisinde borrowBook ve returnBook işlemleri yapılmaktadır.
- LibraryStaff içerisinde addBook ,removeBook ve addUser işlemleri yapılmaktadır.
- Book classında kitap isim ve numara değerlerinin getter ve setter methodları yer almaktadır.
- Library classında Book tipinde iki tane private static arrayList tutulmuştur. Biri book listesini diğeri ise LibraryUser'ın kitap ödünç alma işlemi için borrowBook listesidir. Bu listeye göre staff ödünç alınan bir kitabı remove etmek istediğinde kullanıcıya kitabın ödünç alındığı ve iade edilince remove edilebileceği bilgisi uyarı olarak verilir.
- Ödevde arrayList kullanılmıştır.
- Tüm methodların görevleri yerine getirilmiştir.
- Dosyada(.csv) ilk başta staff,user ve book listesi alınarak okunur. Her işlemde listede olduğu gibi dosyada da güncelleme yapılır.Result çıktılarında örnekleri mevcuttur.
- Staff içinde LibraryUser tipinde private bir liste tutulmuştur bu liste LibraryUser listesidir.Bu listede user çıkarma işlemini sadece Staff yapmaktadır.
- Ödevde belirtilmediği için LibraryUser remove işlemi ve add/remove Staff işlemleri yapılmamaktadır.
- Gerekli görülen yerlerde Exception kullanılmıştır.

2. Use Case Diagrams



- Kullanıcı olarak Staff ve User bulunmaktadır.
- Staff User ekleyebilir. Staff kitap ekleyip çıkarabilir.
- User kitap ödünç alır ve geri iade edebilir.
- Tüm bu kitap alıp verme işlemleri kütüphane class'ın içindeki kitap listelerinde tutulmaktadır.

3. Class Diagrams



4. Problem Solutions Approach

Bilgileri kullanıcıdan olabildiğince saklanmak amaçlanmıştır. Encapsulation amaçlanarak oluşturulan hiyerarşide **interface** kullanılmıştır.

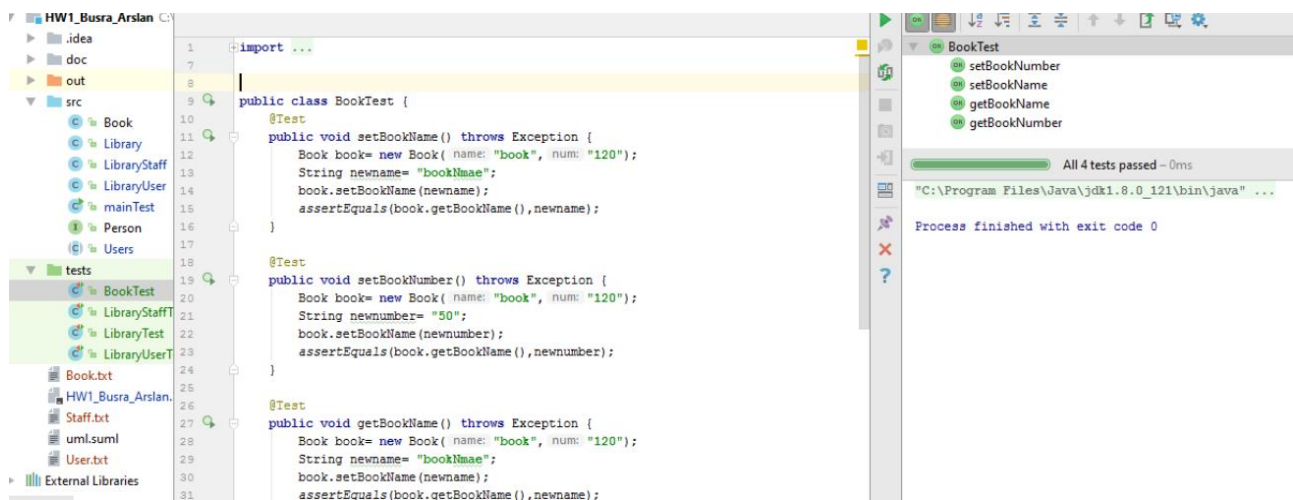
İnterface mantığı ile oluşturduğum ödevde methodlar **Override** edilmiştir.

Inheritance ile değişik kullanıcılar eklenebilmesi amaçlanmıştır. Böylelikle sistem üzerinde sonradan değişiklik olur.

Polimorfizm düzeni kullanılarak da encapsulation sağlanmıştır.

5. Test Cases

Book Class Test



The screenshot displays an IDE interface with a project named 'HW1_Busra_Arslan'. The left sidebar shows a file explorer with a 'tests' folder containing 'BookTest', 'LibraryStaffT', 'LibraryTest', and 'LibraryUserT'. The main editor shows the 'BookTest' class with three test methods: 'setBookName', 'setBookNumber', and 'getBookName'. Each method is annotated with '@Test' and 'throws Exception'. The code for each test method is as follows:

```
1  import ...
2
3  public class BookTest {
4
5      @Test
6      public void setBookName() throws Exception {
7          Book book= new Book( name: "book", num: "120");
8          String newname= "bookNmae";
9          book.setBookName(newname);
10         assertEquals(book.getBookName(),newname);
11     }
12
13     @Test
14     public void setBookNumber() throws Exception {
15         Book book= new Book( name: "book", num: "120");
16         String newnumber= "50";
17         book.setBookName(newnumber);
18         assertEquals(book.getBookName(),newnumber);
19     }
20
21     @Test
22     public void getBookName() throws Exception {
23         Book book= new Book( name: "book", num: "120");
24         String newname= "bookNmae";
25         book.setBookName(newname);
26         assertEquals(book.getBookName(),newname);
27     }
28 }
```

The right sidebar shows the 'BookTest' class with four methods: 'setBookNumber', 'setBookName', 'getBookName', and 'getBookNumber'. Below this, a progress bar indicates 'All 4 tests passed - 0ms'. The command prompt shows the execution path: 'C:\Program Files\Java\jdk1.8.0_121\bin\java' and the message 'Process finished with exit code 0'.

Library Class Test

HW1_Busra_Arslan C:\Users\bs...

idea

doc

out

src

Book

Library

LibraryStaff

LibraryUser

mainTest

Person

Users

tests

BookTest

LibraryStaffTest

LibraryTest

LibraryUserTest

Book.txt

HW1_Busra_Arslan.iml

Staff.txt

uml.suml

User.txt

LibraryTest

removeBook()

```

7
8
9 public class LibraryTest {
10     @Test
11     public void addBook() throws Exception {
12         Book book=new Book( name: "booo555k", num: "23");
13         Library library=new Library();
14
15         library.addBook(book);
16         List<Book> forTest =new ArrayList<>();
17         forTest.add(book);
18         assertEquals(library.getBookList(), forTest);
19     }
20
21     @Test
22     public void removeBook() throws Exception {
23         Book book=new Book( name: "boook", num: "23");
24         Library library=new Library();
25         LibraryStaff staff= new LibraryStaff( name: "bus", passW: "123");
26
27         staff.addBook(library,book);
28         staff.removeBook(library,book);
29
30         List<Book> forTest =new ArrayList<>();

```

LibraryTest

setBorrowBookList

addBook

getBorrowBookList

setBookList

getBookList

removeBook

All 6 tests passed - 0ms

"C:\Program Files\Java\jdk1.8.0_121\bin\java"

Book already exist.Can't add book

Book removed

Process finished with exit code 0

LibraryStaff Class Test

HW1_Busra_Arslan C:\Users\bs...

idea

doc

out

src

Book

Library

LibraryStaff

LibraryUser

mainTest

Person

Users

tests

BookTest

LibraryStaffTest

LibraryTest

LibraryUserTest

Book.txt

HW1_Busra_Arslan.iml

Staff.txt

uml.suml

User.txt

LibraryStaffTest

removeBook()

```

48         assertEquals(staff.getLibraryUserList(),forTest);
49     }
50
51     @Test
52     public void addBook() throws Exception {
53         Book book=new Book( name: "boook", num: "23");
54         LibraryStaff staff= new LibraryStaff( name: "bulut", passW: "123");
55         Library library= new Library();
56         int i=0;
57
58         staff.addBook(library,book);
59         List<Book> forTest =new ArrayList<>();
60         forTest.add(book);
61
62         assertEquals(library.getBookList(), forTest);
63     }
64
65     @Test
66     public void removeBook() throws Exception {
67
68         Book book=new Book( name: "boook", num: "23");
69         LibraryStaff staff= new LibraryStaff( name: "bulut", passW: "123");
70         Library library= new Library();
71
72         List<Book> forTest =new ArrayList<>();
73
74         staff.addBook(library,book);
75         staff.removeBook(library,book);
76         assertEquals(library.getBookList(), forTest);

```

LibraryStaffTest

registerUser

setLibraryUserList

addBook

registerUserList

removeBook

getLibraryUserList

All 6 tests passed - 0ms

"C:\Program Files\Java\jdk1.8.0_121\bin\java" ...

User added by staff

Book added

Book already exist.Can't add book

Book removed

Process finished with exit code 0

LibraryUser Class Test

The screenshot shows an IDE with the `LibraryUserTest` class open. The class contains two test methods: `borrowBook()` and `returnBook()`. The `borrowBook()` method creates a `Book` object, a `LibraryUser` object, and a `Library` object. It then calls `user.borrowBook(library, book)` and asserts that the `Library` object's `getBookList()` method returns the expected list. The `returnBook()` method creates a `Book` object, a `LibraryUser` object, and a `Library` object. It then calls `user.returnBook(library, book)` and asserts that the `Library` object's `getBookList()` method returns the expected list.

The test results panel on the right shows that all 2 tests passed in 16ms. The output of the tests is as follows:

```
LibraryUserTest
- returnBook
- borrowBook

All 2 tests passed - 16ms

"C:\Program Files\Java\jdk1.8.0_121\bin\java" ...
Book returned
Book already exist.Can't return book
Book borrowed

Process finished with exit code 0
```

6. Running and Results

- Eklenecek user eğer dosyada yani listemizde varsa listede mevcuttur bilgisi verir.

The screenshot shows a Java application running in a terminal window. The application is named `131044021_hw1` and is located in the `~/IdeaProjects/131044021_hw1` directory. The application is running the `mainTest` method. The output of the application is as follows:

```
131044021_hw1 > User.csv >
Project: 131044021_hw1
131044021_hw1 ~/IdeaProjects/131044021_hw1
1 userA,123,
2 userB,124,
3 userC,125,
4 userD,126,
5 userE,130,
6

Run: mainTest
mainTest
mainTest

/usr/lib/jdk1.8.0_121/bin/java ...
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
1
Please you enter username and password
Ayse
5555
You entered the system as staff.
Select an operation: [1] or [2] or [3]
[1] Register a new library user
[2] Add a new book
[3] Remove a book
1
Enter a username and password to add a LibraryUser
userA
123
User already exists
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
```

- Exit komutu ile sistemden çıkılır.Exit denmediği sürece sistem işlem bittikten sonra sistemi en baştan çalıştırarak menüye devam eder.

```

/usr/lib/jvm/java-11/bin/java ...
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
1
Please you enter username and password
Ayse
5555
You entered the system as staff.
Select an operation: [1] or [2] or [3]
[1] Register a new library user
[2] Add a new book
[3] Remove a book
1
Enter a username and password to add a LibraryUser
userA
123
User already exists
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
3

Process finished with exit code 1

```

- User bir kitap ödünç aldığı anda(borrow) dosyada alınan kitap güncellenir.Kitap listesinden silinir ve ödünç alınan kitap listesine eklenir.
- HeidiDagda,123 numaralı kitap dosyadan silindi ve artık dosyada yok.

The screenshot shows an IDE with a project named "131044021_hw1". The console window displays the following output:

```

/usr/lib/jdk1.8.0_121/bin/java ...
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
2
Please you enter username and password
UserD
126
This user does not exist on the list
Please you repeat entered bookname and number
userD
126
You entered the system as user.
Select an operation: [1] or [2]
[1] Borrow book
[2] Return book
1
Enter a bookName and number to borrow a Book
HeidiDagda
123
Book borrowed
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit

```

The text editor shows a file named "Book.txt" with the following content:

```

BurdaMevsimKis,563,
Shrek,40,
BuzDevri,30,
BaliklarNasilYuzer,89,

```


- Bir üstteki resimde HeidiDagda isimli 123 numaralı kitabı user ödünç almıştı. Eğer user bir kitabı ödünç aldıktan sonra staff aynı kitabı remove etmek isterse ; o kitap borrowBookList adlı ayrı bir listede tutulduğu için bunun kontrolü yapılır ve uyarı yazısı verilir.

```
Run: mainTest mainTest mainTest
Select an operation: [1] or [2]
[1] Borrow book
[2] Return book
1
Enter a bookName and number to borrow a Book
HeidiDagda
123
Book borrowed
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
1
Please you enter username and password
Ayse
5555
You entered the system as staff.
Select an operation: [1] or [2] or [3]
[1] Register a new library user
[2] Add a new book
[3] Remove a book
3
Enter a bookName and number to remove a Book
HeidiDagda
123
Book can't be removed because it was borrowed by the user.
Not found book. Can't remove book
Which user are you? Please choose: [1] or [2]
[1] Staff
```

- Staff kitap eklemek isterse ama kitap listede varsa listeye kitap eklenmez.
- Ama listede yani dosyada olmayan bir kitap eklenirse kitap hem listeye hem dosyaya eklenir.(Listede değişen kitap verisi sürekli olarak dosyaya update edilir.)

The screenshot shows an IDE with a project named "131044021_hw1". The project structure includes a "main" directory. The console output shows the following sequence of events:

```

[2] Add a new book
[3] Remove a book
2
Enter a bookName and number to add a Book
Shrek
40
Book already exist.Can't add book
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
1
Please you enter username and password
Veli 111
This staff does not exist on the list
Please you repeat entered username and password
Veli 1111
You entered the system as staff.
Select an operation: [1] or [2] or [3]
[1] Register a new library user
[2] Add a new book
[3] Remove a book
2
Enter a bookName and number to add a Book
GTU
36
Book added
Which user are you? Please choose: [1] or [2]
[1] Staff

```

- Staff listede yani dosyada da olmayan bir kitap eklendiğinde dosyayı günceller.Yeni kitabı dosyaya ekler.

The screenshot shows the same IDE with the project "131044021_hw1". The console output shows the following sequence of events:

```

[3] Remove a book
3
Enter a bookName and number to remove a Book
GTU
36
Book removed
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User
[3] Exit
1
Please you enter username and password
Ayse
5555
You entered the system as staff.
Select an operation: [1] or [2] or [3]
[1] Register a new library user
[2] Add a new book
[3] Remove a book
2
Enter a bookName and number to add a Book
GTU
36
Book added
Which user are you? Please choose: [1] or [2]
[1] Staff
[2] User

```

The CSV file "Book.csv" is shown with the following content:

```

1 HeidiDagda,123,
2 BurdaMevsimKis,563,
3 Shrek,40,
4 BuzDevri,30,
5 BaliklarNasilYuzer,89,

```

The console output shows that the book "GTU" with number "36" was added, and the CSV file was updated accordingly.

7)Ödev Part 2

C'de Encapsulation Nasıl Yapılır?

Java da Abstract yapmak için interface kavramından yararlandık.Burada sadece fonksiyonların prototypeları yani fonksiyon isimleri, parametreleri ve döndürdükleri değer verilerrek implemet bilgisi abstract yapılmaya çalışıldı.

C'de .h ve .c dosyaları olarak kodumuzu parçalara ayırırız.

.h'da prototypelar bulunur. İmplement bilgisi ise .c dosyasında bulunur..c de include edilerek kullanılır.Böylelikle kullanıcının görmemesi gereken implement bilgileri saklanmış olur.

```
Typedef struct list * List;
```

```
Struct list{  
    Unsigned length;  
    char* value;  
};
```

Structlarla yardımı ile de bir struct içerisinde başka bir struct yapıyı kullanırız. Bu classların inheritance mantığına benzer bir çalışma gösterip encapsulation sağlar.

Örnek:

```
Struct Node{  
    Node* next;  
    Node* prev;  
};  
Struct List{  
    İnt size;  
    Node* head;  
    Node* tail;  
};
```

Yararlanılan kaynaklar:

<http://inst.eecs.berkeley.edu/~selfpace/studyguide/9C.sg/Output/ADTs.in.C.html>

<http://stackoverflow.com/questions/5997584/how-to-achieve-encapsulation-in-c>