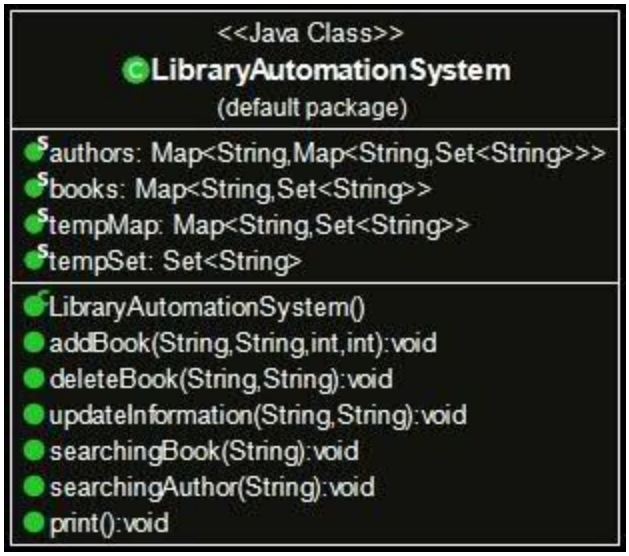
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 6 Report

161044063

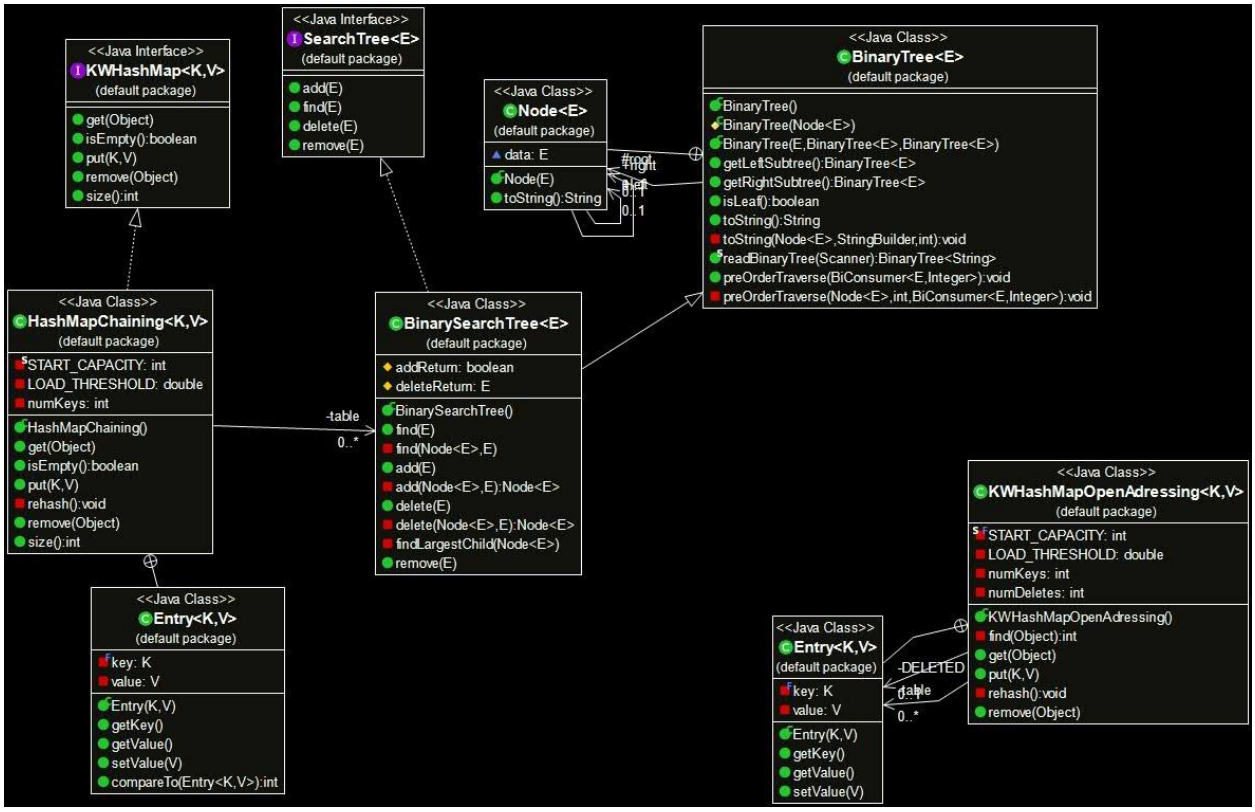
ZAFER ALTAY

1-)CLASS DIAGRAM

FOR PART3:



FOR PART4:



2-)TEST CASES

****Comparisons of Part4 and Part2 will be discarded as separate files.**

For Library Automation System:

TEST ID	T.SCENERIO	TEST DATA	EXPECTED R.	PASS/FAIL
T01	**add Book and control it using print**	**"Halit Ziya Usakligil" "Aşk-ı Memnu" c:2 s:5**	**Author Name: Halit Ziya Usakligil Aşk-ı Memnu Locations of Aşk-ı Memnu [c2s5.6241]**	**PASS**

T02	**add same book**	**"Halit Ziya Usakligil" "Aşk-ı Memnu" c:2 s:5**	**Author Name: Halit Ziya Usakligil Aşk-ı Memnu Locations of Aşk-ı Memnu [c2s5.9054, c2s6.1093]**	**PASS**
T03	**add another book and print all**	**"R.Mahmut Ekrem" "Araba Sevdasi" c3 s4**	**Author Name: R.Mahmut Ekrem Araba Sevdasi Locations of Araba Sevdasi [c3s4.926] Author Name: Halit Ziya Usakligil Aşk-ı Memnu Locations of Aşk-ı Memnu [c2s5.9054, c2s6.1093]**	**PASS**
T04	**delete valid book**	**"Tolstoy" "Anna Karaninna"**	**The book removed succesfully**	**PASS**
T05	**delete invalid book**	**"Ali" "BBBBB"**	**The book already deleted or never added**	**PASS**
T06	**searching author**	**“İskender Pala”**	**OD Location of OD's [c5s5.7261]**	**PASS**
T07	**searching valid book**	**“NUTUK”**	**Author of the Nutuk Kemal Atatürk Locations of the Nutuk [c1s1.7370]**	**PASS**
T08	**update information and print it**	**“OD”**	**Author of the OD İskender Pala Locations of the OD [c9s10.1684]**	**PASS**

For HashMapChaining:

TEST ID	T.SCENERIO	TEST DATA	EXPECTED R.	PASS/FAIL
T0	add a value and check it with get method	"a1","Ali"	Ali	PASS
T1	add 3 element with a key that will have the same index and check it	"a125","Ayse" “a322","Alihan" "4164","Burak"	Ayse Alihan Burak	PASS
T2	adding a new item with the same key and check it	"4164","Not burak is changed"	Not burak is changed	PASS
T3	valid remove and check it	"c2"	null	PASS
T4	invalid remove	“afda”	null	PASS
T5	getting removed element	“c2”	null	PASS

3-) Running command and results

For Library Automation System:

```
Welcome to Library Automation Sysytem
First ,I creating an object for our test
Since we will test all the methods, we assume that it is entered with a password.
We are now testing the addBook and print method. First, we will add the book using the add method, then we will display it with the print method.
We put the book on the 2nd corridor and on the 5th shelf. We will print the book's information using the print function.
The part before the dot sign gives the information of the corridor and shelf

Author Name: Halit Ziya Usakligil
Aşk-ı Memnu
Locations of Aşk-ı Memnu
[c2s5.1093]
Now we are testing to see what happens if there is more than one of the same book,For this we are adding again from the same book
Corridor and Shelf information must be c2s5 and c2s6

Author Name: Halit Ziya Usakligil
Aşk-ı Memnu
Locations of Aşk-ı Memnu
[c2s6.9054, c2s5.1093]
Let's test the situation of having different books
Using addBook method ,i adding another book
Now i'm printing all

Author Name: R.Mahmut Ekrem
Araba Sevdasi
Locations of Araba Sevdasi
[c3s4.926]
Author Name: Halit Ziya Usakligil
Aşk-ı Memnu
Locations of Aşk-ı Memnu
[c2s6.9054, c2s5.1093]
I add a lot of books for better testing
TESTING T05
The book already deleted or never added
```

```
TESTING T04
Please select the book which you want delete
[c2s5.4886]
Your choice(Please enter all code ex: c1s1.1111):
c2s5.4886
The book removed succesfully
TESTING T06
OD
Location of OD's
[c5s5.7261]
TESTING T07
Author of the Nutuk Kemal Atatürk
Locations of the Nutuk
[c1s1.7370]
TESTING T08
Please select the book which you want to update
[c5s5.7261]
Your choice(Please enter all code ex: c1s1.1111):
c5s5.7261
Please enter new locations
Corridor :
9
Shelf :
10
Author of the OD İskender Pala
Locations of the OD
[c9s10.1684]
```

For HashMapChaining:

```
First of all I add a value and check it with get method
Ali
When the mod operation is done, I add 3 element with a key that will have the same index.
I controlled all, sequently
Ayse
Alihan
Burak
I am adding a new item with the same key.
Then I check the change
Not burak is changed
I add a few different elements
Now ı trying remove method
null
null
Now I'm checking the key I removed
null
```

4-)Problem Solution Approach

For Library Automation System:

First of all, I have set a general password for this system. If the user chose admin login when he\she ran the application, I asked for a password and compared it to the general password. I listed the tasks he\she can do if the passwords match. if the user chose guest login, I just listed the search functions, the user can select one of them and run it.

If we talk about methods;

addBook: In this method, I first asked the user for information about the author, the book and the location. Generating a random number, I prevented books from mixing in the same hallway and same shelf. Since the author name is our key, I called if there is a book by the same author in the external hash. If this work is the first book of the author, I first created a set for the location of the book and put that set in the inner map, and then I put the inner map in the outher map. The name of the book is the key to the inner map, and the set with the location is the value of the inner map. This map is the value of the outer map, and the key of the outer map is the name of the author. But if the author's name matches one of the keys of our external map, that author already exists. In this case, this time we compare the book names that are the key to our internal map. If there is a match, the book is already added, I add the locations with a set of values into the set, if there is no match, the book is not attached. I create a new map and add the matching author key inside the outher map. I increase by 1 the number of books after adding.

deleteBook: In this method, I got the name of the author and the book name from the user. I first searched for the name of the author on the external map, if I could not find it, the book is not in the library. I used the get method when searching. I saved a temporary mape. If temp is null, it means it doesn't exist. If the author key matches, I called the get method using the name of the book for the inner map. This is the key to our internal map. If the get method did not return a null value, it means there is a book. I printed the book's locations on the screen and ask the user in which location to remove the book. I removed the user-selected location from the set using the remove method, and then I checked the set's size, if it is zero, I set that set to null.

UpdateInformation: I got the name of the book and author from the user. As in the previous methods, I searched and printed on the screen the locations of the book, if any. I asked the user to select and write the location she\he wants to change. Then I asked him\her to enter the new location. And I deleted the old location in the embankment inside and added a new one. I used the java set and map functions for deleting, adding and getting. Note that I created a temporary reference with get and made transactions on it.

SearchingBook: I created an iterator for our external map, using Map.entry. Since the value set of each element of the operator is the books of an author, I got the book name with get in each iterator element. If the item I received with get is null, I went to the next iterator element. If I can't find the book when the items are finished, it means the book is not in the library. If the item I received with get is not null, it means that I found the book. I wrote the elements in the whole set of the

book on the screen, that is, I printed their location. I used getValue and getKey instead of get because Map.entry turned the elements into a set set.

SearchingAuthor: I called the author's name using the Get method and brought the temporary reference. If the reference is not null, I visited the elements with the iterator I created for the inner map and took the values of all of them with getValue and printed them. So I printed all the books and locations on the screen. I used getKey for book titles.

Print: I created iterator for both outher and internal map. I used a new inner map iterator for each value object of the iterator of the outer map, so I got around all the books. The set of values for each object of the inner map of the iterator indicates the locations of a book in the library.I printed them.

For KWHashMapChaining:

First of all, I added the binary search tree class that I used in the previous assignment. I used this because Java have not got bst class.After that ı created an entry inner class in hash map chaining. This class hold keys and value. I created a bst array for hashTable. Each of the index will hold a bst.

Get: If the index given by the hash code is null, null means null, I returned null. If there is none, there is a tree in the index. We create an entry object whose key is

the same as we are looking for, and we are looking for it with the find method. And I returned the getValue of the value returned by this method**.**

Put: First of all, I found the index of the values we got by applying our formula. If the index is null, I first created a tree in the index. I created an entry element with the values we received and added it to the tree we created using the tree's add method. If the directory I checked is not empty, I created an entry item with the value we received and added it to the tree using the add method of the tree.

In the meantime, if the tree finds a match with the same values in the add method, I replaced the new one to replace the old one and return the old value.

I increased the number of additions after each addition. If it provides features for rehash, I rehash the table

Rehash: In this function, I created a new table that is twice the size of the old table. Then I copied the indexes from the old table to the new table.

Remove : I found the index to remove using our formula. If our index is null, the table is empty. If our index is not empty, I used the delete method of the tree.

After removal, if the data of the tree in the index is null, the tree has no element. In this case, I made that index a null reference. And I returned the value returned by the remove method.

Size: I returned numKeys.

isEmpty: I trace through all the table elements to see if they were null. If all of them are null, the tree is empty.

For KWHashMapOpenAdressing:

Unlike the class in the book in this class, I used hash2 method, which is a very common method for finding index. I used prime- (prime% hashCode) logic in this method. I combined this with the books method and obtained a new index method.