Department of Computer Technology and Information Systems

**CTIS221 – Object Oriented Programming**

Fall 2019 - 2020

**Lab Guide 20**

| **OBJECTIVES:** MAPS |
|---|
| **Instructor** : Burcu LİMAN<br>**Assistants**: Burcu ALPER, Leyla SEZER |

**Q1.** Write a java program that counts the number of occurrences for all letters that are given as a string array.

In the main class implement the following array;

```
String names[] = {"ali", "veli", "ceren", "leyla", "hatice", "burcu", "neşe", "ceren",
"ahmet", "leyla"};
```

Then create a HashMap object to hold words as key and count as value:

```
HashMap<Character,Integer> hm = new HashMap();
```

- Then get the words and count the letters by using get() and put() functions of the hashmap.
- Write a **printMap()** method that gets the hashMap and displays the string, size and isEmpty property. Also display the key and values of the hashMap separetly.

**Example Output:**

```
Number of occurences of each letter: {a=5, b=1, c=4, e=11, h=2, i=3, l=6, m=1, n=3, r=3, t=2, u=2,
v=1, y=2, ş=1}
size: 15
isEmpty: false
Key Set: [a, b, c, e, h, i, l, m, n, r, t, u, v, y, ş]
Values: [5, 1, 4, 11, 2, 3, 6, 1, 3, 3, 2, 2, 1, 2, 1]
```

**hm HasMap structure:**

Key: Character Value: Integer

- Key: Character that will be counted
- Value: The number of occurrence of a character

| | |
|---|---|
| a | 5 |
| b | 1 |
| c | 4 |
| d | 0 |
| e | 11 |
| . . . | . . . |

**Q2.** Write a Java program that reads the student names from the files and creates a map to put the students in it according to their section numbers.

**PART A: <u>Implement your classes</u>**

Your program will get input from unknown number of text file named with section1, section2, …, section# with the following structures;
The files includes name and surname of the students.

section1.txt

```
Ali Beyhan
Yavuz Selim Ekin
Jaehon Park
Ali Yagiz Mumcu
Suheera Tanvir
```

section2.txt

```
Turan Yücel
Elchin Latifli
Baris Aksakal
Furkan Kilic
Cem Kurulay
```

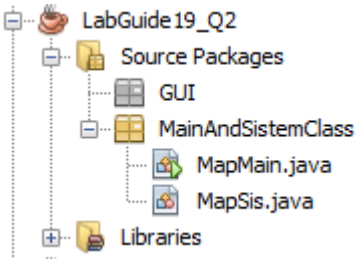
section3.txt

```
Nil Mumcu
Ya-Ting Yang
Murat Alkan
Ercan Emre Celik
Osman Cem Arslan
```

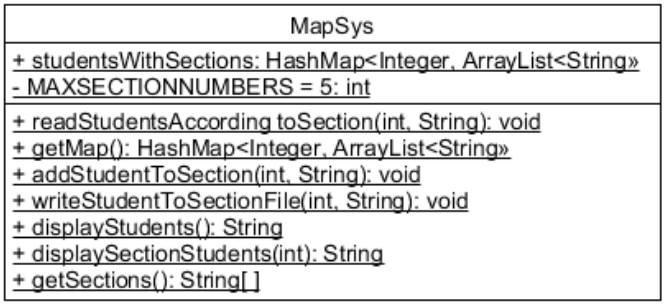
section#.txt

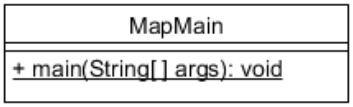
```
...
```

You are going to create each class in separate java files. Packages and the file names are as follows;




➔ Check the UML class diagram, implement your class according to it, and do not change visibility modifiers.




***<u>Information of the MapSys class structure:</u>***

- Crete the data member;
  A HashMap object that contains Integer key which corresponds to String Array List:

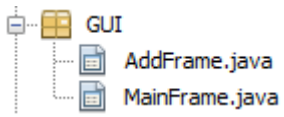
```
HashMap<Integer,ArrayList<String>> studentsWithSections = new HashMap<Integer,ArrayList<String>>();
```

studentsWithSections HasMap structure:

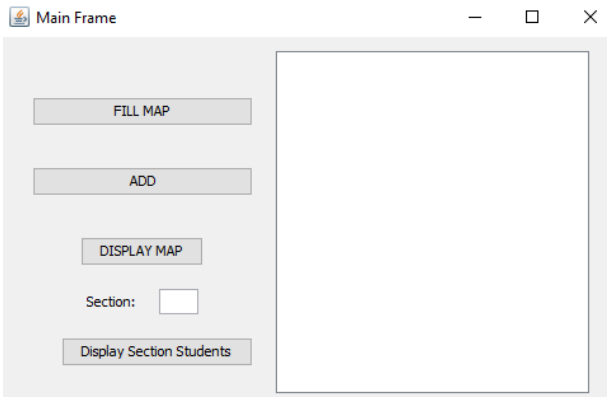
| Section number (key) | studentList (value) |
|---|---|
| 1 | studentList:<br>Ali Beyhan<br>Yavuz Selim Ekin<br>Jaehon Park<br>Ali Yagiz Mumcu<br>Suheera Tanvir<br>. . . |
| 2 | studentList:<br>Turan Yücel<br>Elchin Latifli<br>Baris Aksakal<br>Furkan Kilic<br>Cem Kurulay<br>. . . |
| 3 | studentList:<br>Nil Mumcu<br>Ya-Ting Yang<br>Murat Alkan<br>Ercan Emre Celik<br>Osman Cem Arslan<br>. . . |

- Implement a **readStudentsAccordingToSections()** method that reads the all files from section1.txt to section#.txt into the Hash Map**. #of files is not known.**
- Implement a **getMap()** method that will return the map.
- Implement a **addStudentToSection()** method that gets the section number and student name as parameters then the method checks if the section number exist or not. If section exists, get the ArrayList which includes student names from the map according to the section number and add the new student to the section. If the section number is not exist the method adds the new section number with new student list to the map.
- Implement a w**riteStudentToSectionFile(…)** method that gets the section number and the student name as parameters. The function appends the student name to end of the file with the specified section number.
- Implement a **displayStudents()** method that returns the map content.
- Implement a **displaySectionStudents()** method that gets the section number as parameter to returns the student list as a string.
- Implement a **getSections()** method that returns a String array that contains section numbers.

**PART B: Implement your GUI**

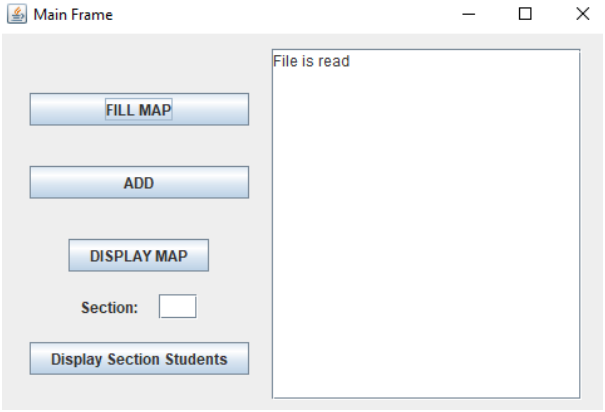



**1)** When program starts in main class;
   **a.** The start-up frame should be created and sets its visibility to true;
      i. There are 3 buttons, 1 text field, 1 label and 1 text area as shown below.
      ii. Set the title to the “Main Frame”.




**b.** When the user clicks on the **"FILL MAP"** button,

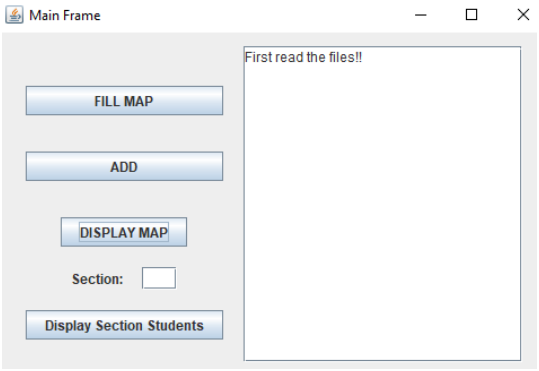
i. Read the files by invoking the **readStudentsAccordingToSections()** method from the MapSys and display a message on the text area.

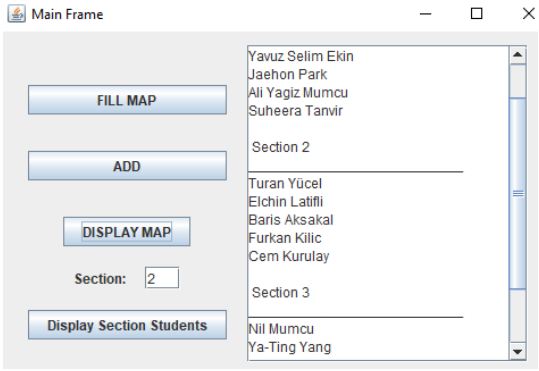



**c.** When the user clicks on the **"DISPLAY MAP"** button,

i. If the map is empty, give a warning message.

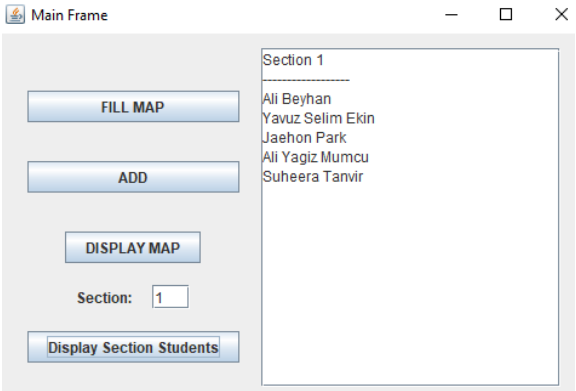
ii. Display the content of the map on the textArea by invoking the **displayStudents()** method.

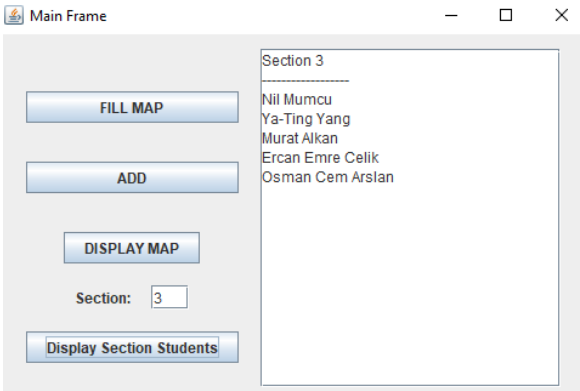



**d.** When the user clicks on the **"Display Section Students"** button,

i. Display the selected section's student list by invoking the **displaySelectionStudents()**



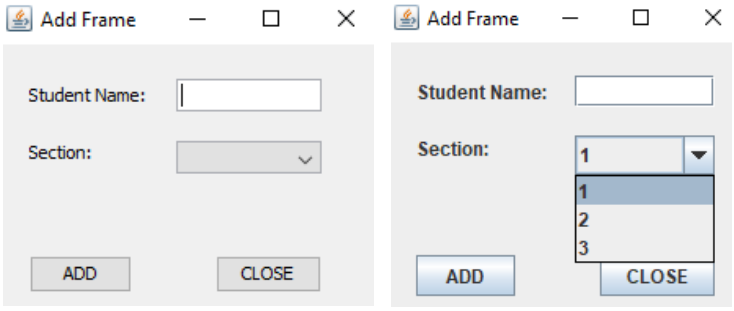
**e.** When the user clicks on the **"ADD"** button;

i. A frame should be created and sets its visibility to true;

ii. Set the title to the "Add Frame"

iii. There are 1 buttons, 1 combo box, 1 text area, 2 labels as shown below.

iv. All section numbers in the map will be shown on the combo box.




v. When a section is selected from the combo box, add the student to the student list in that section by invoking the **addStudentToSection()** method and display a message on the label. Also write the new student name to the file with the specified section number by invoking the **writeStudentToSectionFile()** method.

vi. When "CLOSE" button is clicked clear the text field and the message label and set the selected item as the first one then dispose the frame.