Requirements in details

4.1 Non-functional Requirements

- Java as a programming language.
- The system runs offline.
- Bug/Log report system to keep track of what types of errors occurred.
- The system will be scalable and maintainable.

4.2 Functional Requirements

- The method that checks if the given element is in the heap tree.
-A method that merges two Heap Trees.
-A method to remove the ith largest element in a heap tree.
-A method to set the value (value passed as parameter) of the last element returned by the next methods.
-The methods where the placement rule of the Min Heap Tree's elements will not break.
-Some auxiliary methods (size, toString ... etc.) used to perform main operations.
-Main methods of the iterator class.

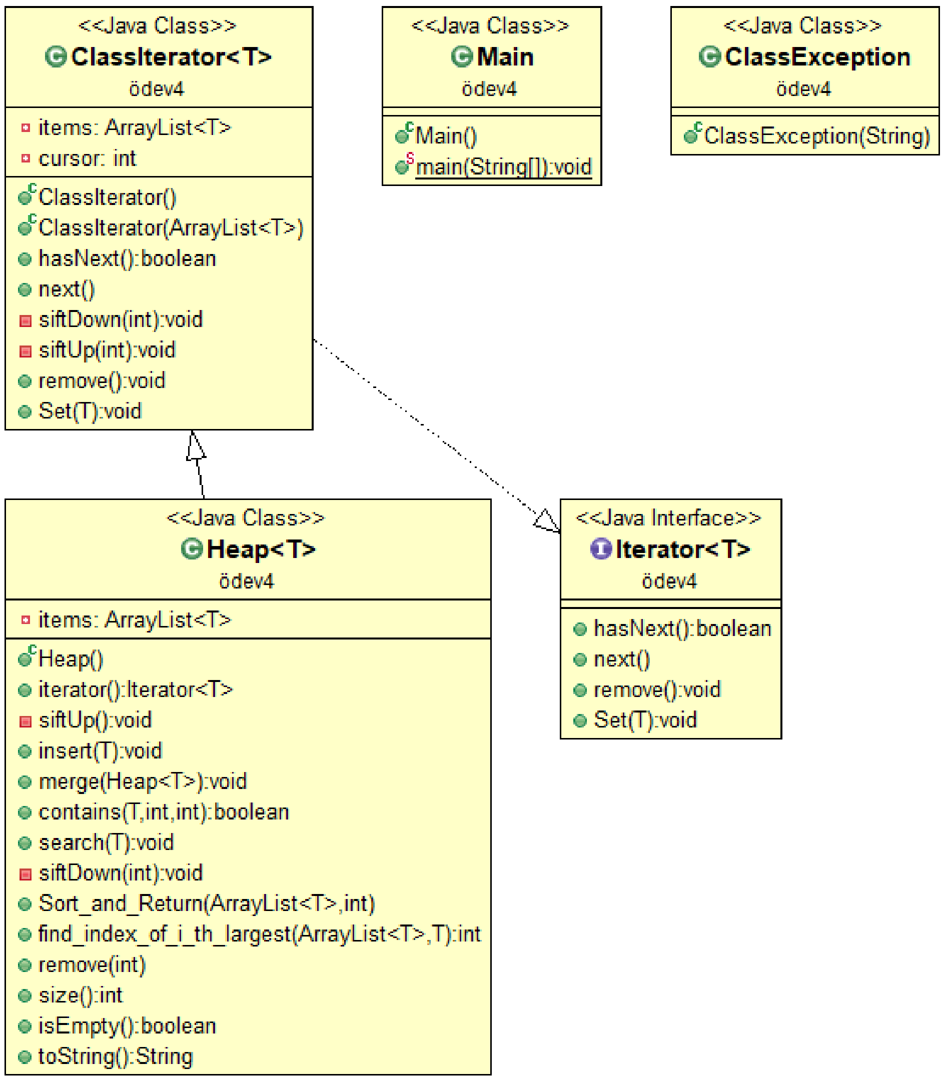
4.3 User Requirements

- User needs to enter the value he wants to search in the heap tree.
-It needs to enter the ith largest element to remove it from the Heap tree.
-It needs to enter the value that it wants the iterator to set to the Heap tree.

4.4 System requirements

- Checks if the Heap Tree is empty or not.
-The ith checks that the i value of the largest element to be removed is not greater than the size of the heap tree.

Class Diagram



Problem solutions approach

Before designing the desired methods, it was necessary to add elements in the Min Heap Tree.For this, I made adding elements to the node at the end of Tree because the Heap Tree logic would be appropriate.When adding this element, I did a check as it has to have a smaller element than parent's his children.

I recursively browsed the Min Heap Tree for the search process, accessing the right and left children of the parents.

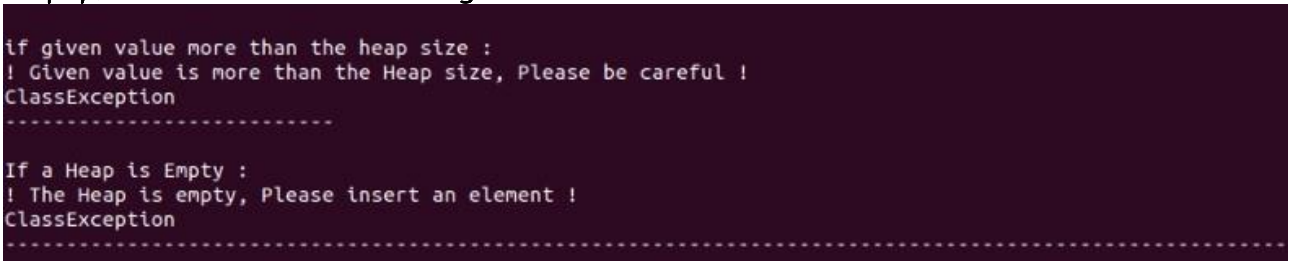
I have created a method in my Heap Class for the merge method. This method takes a Heap Tree. I add it to the other Heap Tree by browsing through the Heap Tree given in the method one by one.

When I am finding the ith largest element in the heap Tree, I first order the Heap Tree in ascending order using an auxiliary function and retry to be in the i^th place.Later, using another helper function, I search for this element in unordered the Heap Tree and return its index.and I remove the element in this index from Tree. However, it is not enough to do this.Finally, according to different situations, I make a comparison and arrangement by using an auxiliary function so that the structure of the Heap Tree is not damaged.

In the part of adding set method to the iterator class, I first created an iterator interface for test cases in my driver function, implements this interface to my iterator class and overrided the interface's methods in the class.In other words, I created my iterator class. Since the set operation will be applied to the last value returned by the next method of the iterator, I created a cursor variable in the iterator class and thus changed the last location of my iterator object.Lastly, I made an arrangement and comparison in parents and childs according to the added value in order not to break the structure of the Min Heap Tree.

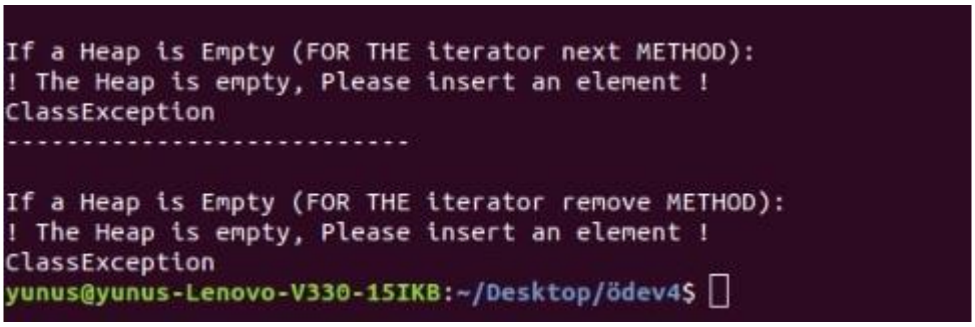
Test cases

When removing the i'th largest element from the Min Heap Tree, if the value entered by the user is more than the size of the Tree and if the Min Heap Tree is empty, I throw an error message to the user.



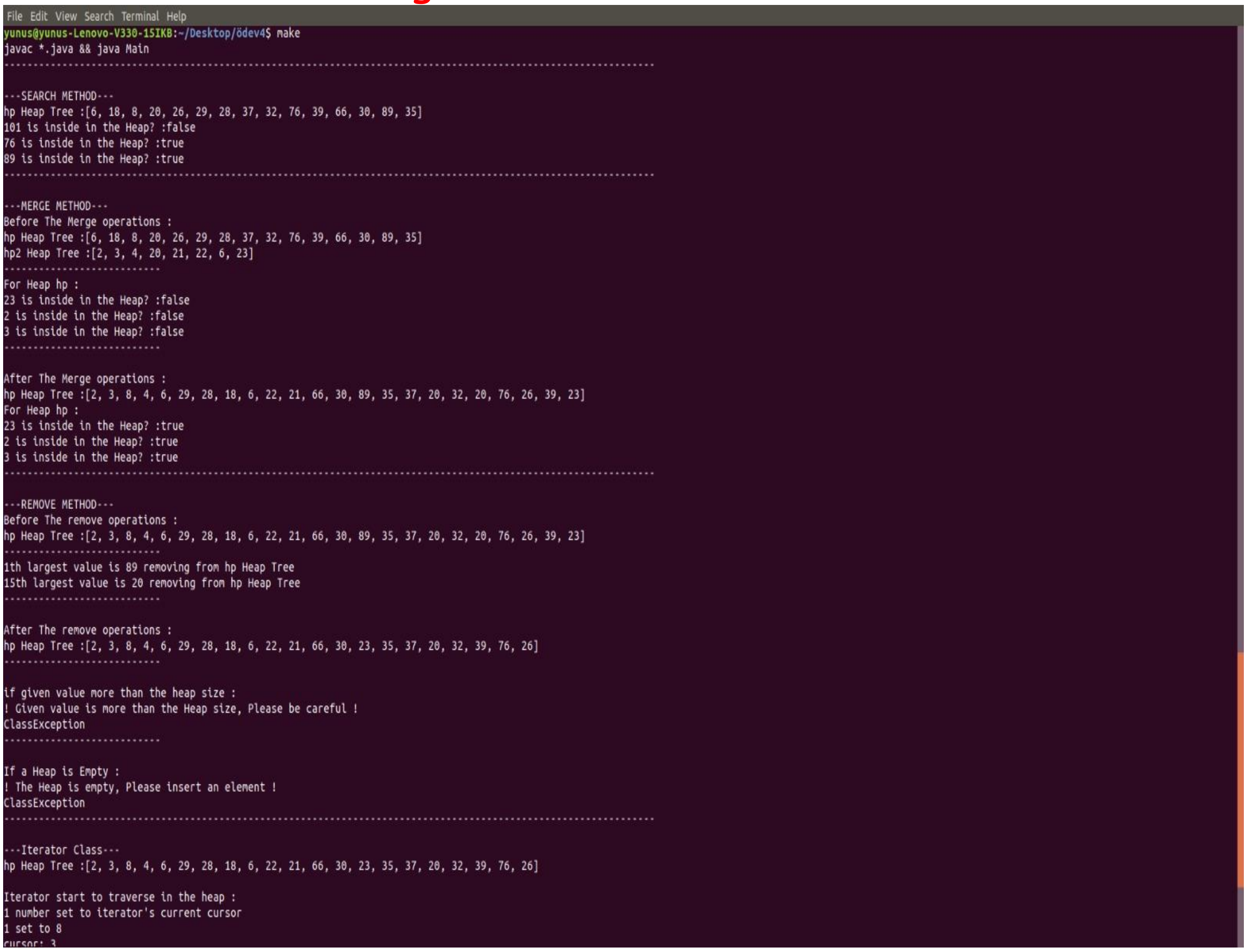
```
if given value more than the heap size :
! Given value is more than the Heap size, Please be careful !
ClassException
---------------------------

If a Heap is Empty :
! The Heap is empty, Please insert an element !
ClassException
-----------------------------------------------------------------------------------------------------------
```

If the Min Heap Tree is empty, I throw an error message to the user in the next and remove methods of the iterator class.

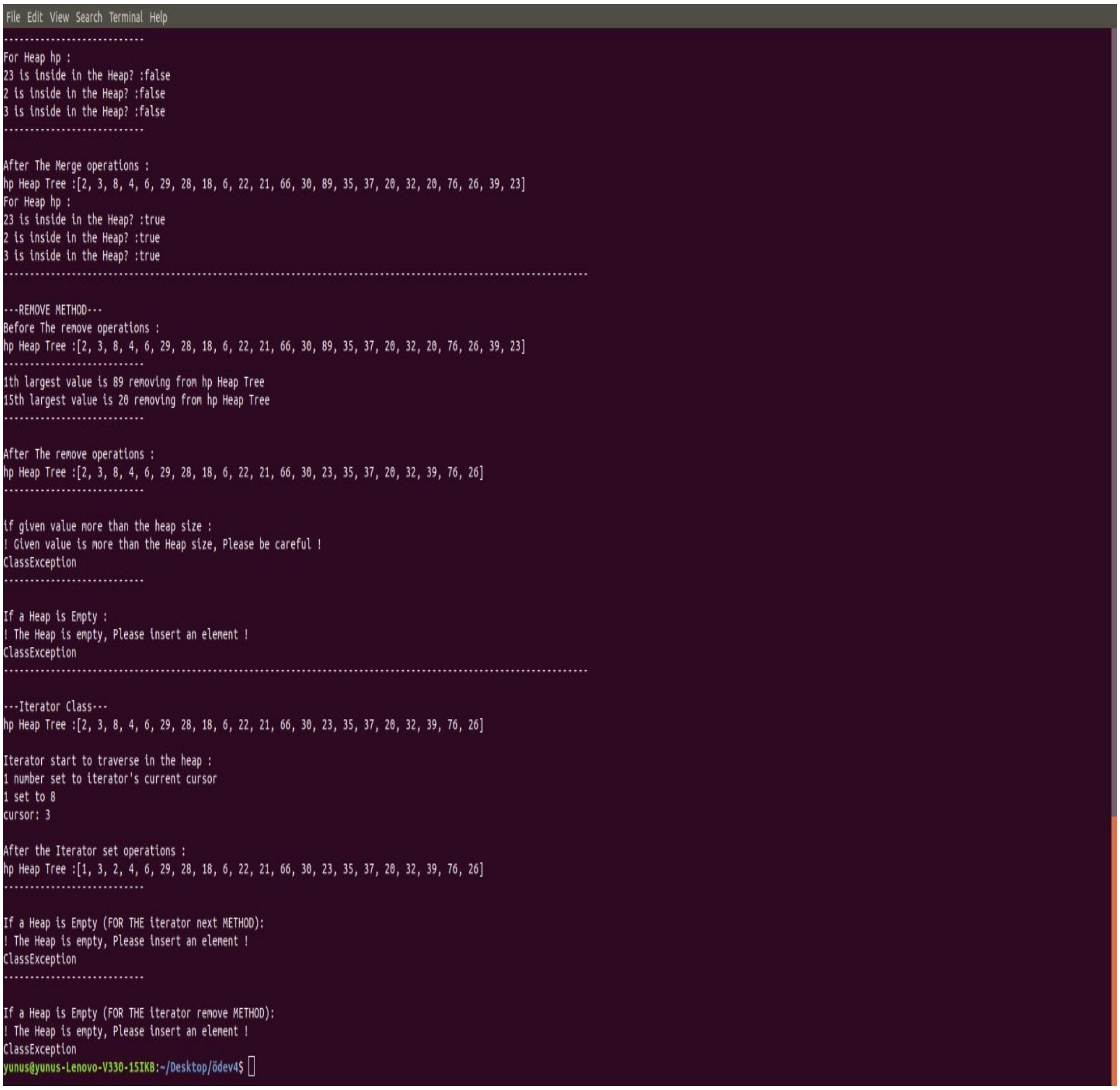


```
If a Heap is Empty (FOR THE iterator next METHOD):
! The Heap is empty, Please insert an element !
ClassException
---------------------------

If a Heap is Empty (FOR THE iterator remove METHOD):
! The Heap is empty, Please insert an element !
ClassException
yunus@yunus-Lenovo-V330-15IKB:~/Desktop/ödev4$ 
```

Running Command and Results



```
yunus@yunus-Lenovo-V330-15IKB:~/Desktop/ödev4$ make
javac *.java && java Main
------------------------------------------------------------------------------------------------------------


---SEARCH METHOD---
hp Heap Tree :[6, 18, 8, 20, 26, 29, 28, 37, 32, 76, 39, 66, 30, 89, 35]
101 is inside in the Heap? :false
76 is inside in the Heap? :true
89 is inside in the Heap? :true
------------------------------------------------------------------------------------------------------------


---MERGE METHOD---
Before The Merge operations :
hp Heap Tree :[6, 18, 8, 20, 26, 29, 28, 37, 32, 76, 39, 66, 30, 89, 35]
hp2 Heap Tree :[2, 3, 4, 20, 21, 22, 6, 23]
--------------------------
For Heap hp :
23 is inside in the Heap? :false
2 is inside in the Heap? :false
3 is inside in the Heap? :false
--------------------------


After The Merge operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 89, 35, 37, 20, 32, 20, 76, 26, 39, 23]
For Heap hp :
23 is inside in the Heap? :true
2 is inside in the Heap? :true
3 is inside in the Heap? :true
------------------------------------------------------------------------------------------------------------


---REMOVE METHOD---
Before The remove operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 89, 35, 37, 20, 32, 20, 76, 26, 39, 23]
--------------------------
1th largest value is 89 removing from hp Heap Tree
15th largest value is 20 removing from hp Heap Tree
--------------------------


After The remove operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 23, 35, 37, 20, 32, 39, 76, 26]
--------------------------


if given value more than the heap size :
! Given value is more than the Heap size, Please be careful !
ClassException
--------------------------


If a Heap is Empty :
! The Heap is empty, Please insert an element !
ClassException
------------------------------------------------------------------------------------------------------------


---Iterator Class---
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 23, 35, 37, 20, 32, 39, 76, 26]

Iterator start to traverse in the heap :
1 number set to iterator's current cursor
1 set to 8
cursor: 3
```



```
--------------------------
For Heap hp :
23 is inside in the Heap? :false
2 is inside in the Heap? :false
3 is inside in the Heap? :false
--------------------------

After The Merge operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 89, 35, 37, 20, 32, 20, 76, 26, 39, 23]
For Heap hp :
23 is inside in the Heap? :true
2 is inside in the Heap? :true
3 is inside in the Heap? :true
------------------------------------------------------------------------------------------------------------

---REMOVE METHOD---
Before The remove operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 89, 35, 37, 20, 32, 20, 76, 26, 39, 23]
--------------------------
1th largest value is 89 removing from hp Heap Tree
15th largest value is 20 removing from hp Heap Tree
--------------------------

After The remove operations :
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 23, 35, 37, 20, 32, 39, 76, 26]
--------------------------

if given value more than the heap size :
! Given value is more than the Heap size, Please be careful !
ClassException
--------------------------

If a Heap is Empty :
! The Heap is empty, Please insert an element !
ClassException
------------------------------------------------------------------------------------------------------------

---Iterator Class---
hp Heap Tree :[2, 3, 8, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 23, 35, 37, 20, 32, 39, 76, 26]

Iterator start to traverse in the heap :
1 number set to iterator's current cursor
1 set to 8
cursor: 3

After the Iterator set operations :
hp Heap Tree :[1, 3, 2, 4, 6, 29, 28, 18, 6, 22, 21, 66, 30, 23, 35, 37, 20, 32, 39, 76, 26]
--------------------------

If a Heap is Empty (FOR THE iterator next METHOD):
! The Heap is empty, Please insert an element !
ClassException
--------------------------

If a Heap is Empty (FOR THE iterator remove METHOD):
! The Heap is empty, Please insert an element !
ClassException
yunus@yunus-Lenovo-V330-15IKB:~/Desktop/ödev4$ 
```