

HW5

Part 1: Implement the Towers of Hanoi problem iteratively

Part 2: Implement the remove procedure from LinkListRec class but your procedure removes all duplicate elements in linked list

Part 3: Implement a class that have two sorted array, list1 and list2. You should write 3 recursive procedure that do some operations on these lists

Adı Soyadı : Mercan Karacabey

No: 131044034

System Requirement

- 1- Implement the Towers of Hanoi problem iteratively

Bu problemin çözümü için

- Kaç tane disk olduğu
- Source
- Destination

Gerekmektedir

- 2- Implement the remove procedure from LinkListRec class but your procedure removes all duplicate elements in linked list

Bu problemin çözümü için

- Bir liste ve onun tekrar eden elemanlarını oluşturmak ve sonrasında tüm duplicate Elemanları silmek gerekiyordu.

- 3-: Implement a class that have two sorted array, list1 and list2. You should write 3 recursive procedure that do some operations on these lists

Bu problemin çözümü için

public List intersectionOfLists()

public List unionOfLists()

ilk iki method için iki liste alıp gerekli

işlemlerden sonra yenisini oluşturmak gerekmektedir.

public boolean isSubset()

Bu methodda ise yine iki list olarak birincisinin ikincisinin alt kümesi olup olmadığı kontrol edilecektir.

CLASS DIAGRAM

TowerOfHanoi

-Execute()
-Move()

RecursiveList

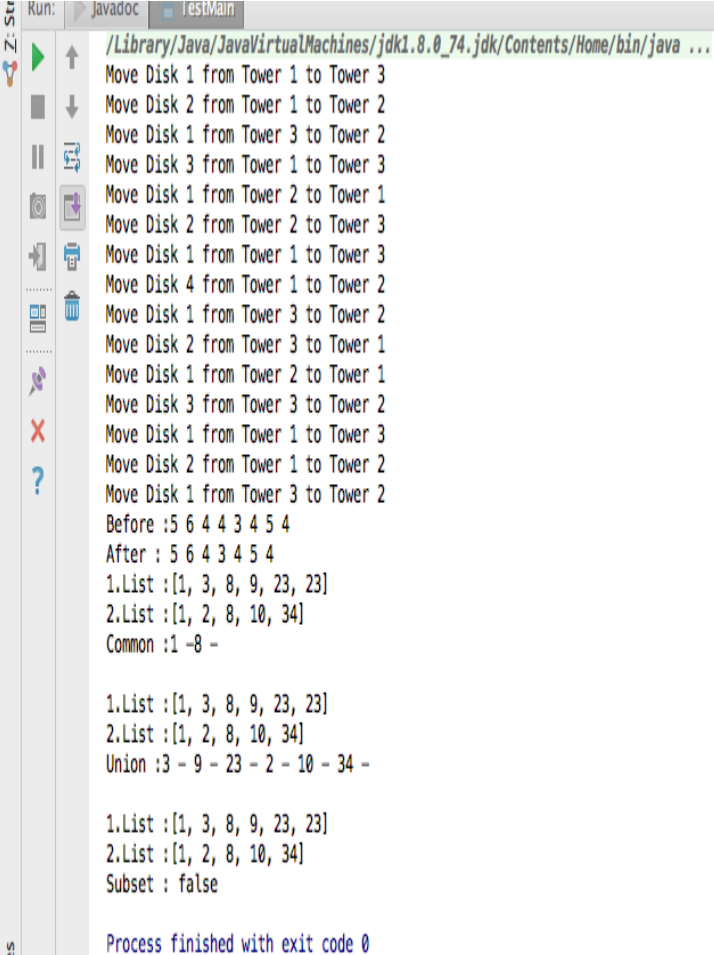
Node<E> inner class
-next
-data

```
private boolean remove(Node < E > head, Node < E > pred, E outData)  
public boolean remove(E outData)
```

part3

```
public List<E> findCommonElement(List<E> list1, List<E> list2, List<E>  
resultArray)  
public List<E> unionOfList(List<E> list1, List<E> list2, List<E> resultArray)  
public boolean isSubset(List<E> list1, List<E> list2)
```

Running Command



```
Run: javadoc TestMain
/Library/Java/JavaVirtualMachines/jdk1.8.0_74.jdk/Contents/Home/bin/java ...
Move Disk 1 from Tower 1 to Tower 3
Move Disk 2 from Tower 1 to Tower 2
Move Disk 1 from Tower 3 to Tower 2
Move Disk 3 from Tower 1 to Tower 3
Move Disk 1 from Tower 2 to Tower 1
Move Disk 2 from Tower 2 to Tower 3
Move Disk 1 from Tower 1 to Tower 3
Move Disk 4 from Tower 1 to Tower 2
Move Disk 1 from Tower 3 to Tower 2
Move Disk 2 from Tower 3 to Tower 1
Move Disk 1 from Tower 2 to Tower 1
Move Disk 3 from Tower 3 to Tower 2
Move Disk 1 from Tower 1 to Tower 3
Move Disk 2 from Tower 1 to Tower 2
Move Disk 1 from Tower 3 to Tower 2
Before :5 6 4 4 3 4 5 4
After : 5 6 4 3 4 5 4
1.List :[1, 3, 8, 9, 23, 23]
2.List :[1, 2, 8, 10, 34]
Common :1 -8 -

1.List :[1, 3, 8, 9, 23, 23]
2.List :[1, 2, 8, 10, 34]
Union :3 - 9 - 23 - 2 - 10 - 34 -

1.List :[1, 3, 8, 9, 23, 23]
2.List :[1, 2, 8, 10, 34]
Subset : false

Process finished with exit code 0
```