

2 KthSmallestElement.java

```
import java.util.ArrayList;
import util.Collections;
import util.Scanner;
public class KthSmallestElement {
    public static void int findKthSmallest(ArrayList<Integer> list, int k) {
        if (k <= 0 || k > list.size()) {
            throw new IllegalArgumentException("k is out of bound");
        }
        Collection.sort(list);
        return list.get(k - 1);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Integer> list = new ArrayList<>();
        System.out.println("Enter the numbers of elements : ");
        int n = scanner.nextInt();
        System.out.println("Enter the elements : ");
    }
}
```

```
for(int i=0; i<n; i++)  
    list.add(scanner.nextInt());
```

System.out.println("Enter the value of k:");

```
int k = scanner.nextInt();
```

```
try{
```

```
int kthSmallest = findKthSmallest(list, k);
```

System.out.println("The " + k + "th smallest element is: " + kthSmallest);

```
}
```

catch(InvalidArgumentException e){}

System.out.println("Error: " + e.getMessage());

```
scanner.close();
```

if(k > n) {
 System.out.println("k is greater than n");
}

i("is increased")

if(i > n) {
 System.out.println("i is greater than n");
}

Word frequency Counter

```
import java.util.Map;  
import java.util.Scanner;  
import java.util.TreeMap;  
public class wordfrequencyCounter {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Enter your text:");  
        String text = scanner.nextLine();  
        String[] words = text.toLowerCase().split("\\w+");  
        TreeMap<String, Integer> wordFreq = new TreeMap<  
            String>();  
        for (String word : words) {  
            if (word.isEmpty()) {  
                continue;  
            }  
            wordFreq.put(word, wordFreq.getOrDefault(w  
                ord, 0) + 1);  
        }  
    }  
}
```

```
System.out.println("In Word Frequencies:");
for (Map.Entry<String, Integer> entry : wordFrequency.entrySet()) {
    System.out.println(entry.getKey() + ":" + entry.getValue());
}
Scanner.close();
```

```
class Node {
    String data;
    int freq;
    Node left;
    Node right;
}
```

Map ID name

```
import java.util.Map;
import java.util.Scanner;
import java.util.TreeMap;
public class mapIDName {
    public static void main (String [] args) {
        TreeMap<Integer, String> studentMap = new TreeMap();
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the number of students");
        int n = sc.nextInt();
        sc.nextLine();
        for (int i=0; i < n; i++) {
            System.out.println ("Enter studentID (integer): ");
            int id = sc.nextInt();
            sc.nextLine();
            System.out.println ("Enter details (e.g., name, department): ");
            String details = sc.nextLine();
            studentMap.put (id, details);
        }
    }
}
```

```
System.out.println("n .. student details (sorted) ..")
for (Map.Entry<Integer, String> entry : studentMap
    .entrySet())
```

```
System.out.println("ID: " + entry.getKey() + " " +
```

Details: " + entry.getValue());

Output: {point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

{point = 90, name = "John Doe", age = 20, marks = 90, address = "Mumbai", fees = 10000}

Q) Linklist equality checker

```
import java.util.LinkedList;
import java.util.Iterator;
public class LinkedListEqualityChecker {
    public static <T> boolean areEqual(LinkedList<T> list1, LinkedList<T> list2) {
        if (list1.size() != list2.size())
            return false;
        Iterator<T> it1 = list1.iterator();
        Iterator<T> it2 = list2.iterator();
        while (it1.hasNext() && it2.hasNext()) {
            T val1 = it1.next();
            T val2 = it2.next();
            if (!val1.equals(val2))
                return false;
        }
    }
}
```

returns true, it means utilises a function to

```
public static void main (String [] args) {
    LinkedList<Integer> listA = new LinkedList<()>
    LinkedList<Integer> listB = new LinkedList<()>
    listA.add(10);
    listA.add(20);
    listA.add(30);
    listB.add(10);
    listB.add(20);
    listB.add(30);
    if (areEqual(listA, listB))
        System.out.println ("The linked lists are equal");
    else
        System.out.println ("The linked lists are not equal");
}
```

```
④ for EmployeeDepartmentMap) rettiring.two.mst2p2
import java.util.HashMap<String, Integer> employeeMap) rof
import java.util.Map<() to Lenes. qMoryplqac2
import java.util.Scanner; rettiring.two.mst2p2
public class EmployeeDepartmentMap) f( fextsp) . ptoz
public static void main (String [] args) {
    HashMap<Integer, String> employeeMap = new Hashm
    Scanner scanner = new Scanner (System.in);
    System.out.print("Enter number of employees:");
    int n = scanner.nextInt();
    for (int i=0; i<n; i++) {
        System.out.print("Enter Employee ID: ");
        int empId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Department: ");
        String department = scanner.nextLine();
        employeeMap.put(empId, department);
    }
}
```

```
System.out.println("Enter Employee I.D to Department");
for (Map.Entry<Integer, String> entry : employeeMap.entrySet()) {
    System.out.println("Employee ID: " + entry.getKey() + ", Department: " + entry.getValue());
}
```

```
System.out.print("Enter ID to search: ");
int searchId = scanner.nextInt();
String dept = employeeMap.get(searchId);
if (dept != null) {
    System.out.println("Employee ID " + searchId +
        " belongs to Department: " + dept);
```

```
} else {
    System.out.println("Employee ID not found")
```

```
} scanner.close();
```