1.public class Emp1 {

private int id;

private String name;

private String dept;

private double sal;

public Emp1(int id, String name, String dept, double sal) {

super();

this.id = id;

this.name = name;

this.dept = dept;

this.sal = sal;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDept() {

return dept;

}

public void setDept(String dept) {

this.dept = dept;

}

public double getSal() {

return sal;

}

public void setSal(double sal) {

this.sal = sal;

}

}

import java.util.Comparator;

public class IdCompare implements Comparator<Emp1>{

public int compare(Emp1 o1, Emp1 o2)

{

return o1.getId() - o2.getId();

}

}

import java.util.Comparator;

public class NCompare implements Comparator<Emp1>{

public int compare(Emp1 o1, Emp1 o2)

{

return o1.getName().compareTo(o2.getName());

}

}

import java.util.Comparator;

public class DCompare implements Comparator<Emp1>{

public int compare(Emp1 o1, Emp1 o2)

{

return o1.getDept().compareTo(o2.getDept());

}

}

import java.util.Comparator;

public class SCompare implements Comparator<Emp1>{

public int compare(Emp1 o1, Emp1 o2)

{

return o1.getDept().compareTo(o2.getDept());

}

}

import java.util.\*;

import java.util.TreeSet;

public class Compare {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc = new Scanner(System.in);

System.out.println("You want to sort in order of \n\n1.ID\n2.Department\n3.Name\n4.Salary\n\nEnter your option: ");

int option = sc.nextInt();

switch(option)

{

case 1:

TreeSet<Emp1> tset = new TreeSet<Emp1>(new IdCompare());

tset.add(new Emp1(1,"Trainee","Ram",18000.0));

tset.add(new Emp1(2,"Manager","Ravi",32000.0));

tset.add(new Emp1(3,"Analyst","Sam",21000.0));

System.out.println(" Increasing Order with the Id : ");

for(Emp1 o : tset)

{

System.out.print(o.getId()+","+o.getDept()+","+o.getName()+","+o.getSal());

System.out.println();

}

break;

case 2:

TreeSet<Emp1> tset1 = new TreeSet<Emp1>(new NCompare());

tset1.add(new Emp1(1,"Trainee","Ram",18000.0));

tset1.add(new Emp1(2,"Manager","Ravi",32000.0));

tset1.add(new Emp1(3,"Analyst","Sam",21000.0));

System.out.println(" Increasing Order with the Name : ");

for(Emp1 o : tset1)

{

System.out.print(o.getId()+","+o.getDept()+","+o.getName()+","+o.getSal());

System.out.println();

}

break;

case 3:

TreeSet<Emp1> tset2 = new TreeSet<Emp1>(new DCompare());

tset2.add(new Emp1(1,"Trainee","Ram",18000.0));

tset2.add(new Emp1(2,"Manager","Ravi",32000.0));

tset2.add(new Emp1(3,"Analyst","Sam",21000.0));

System.out.println(" Increasing Order with the Department : ");

for(Emp1 o : tset2)

{

System.out.print(o.getId()+","+o.getDept()+","+o.getName()+","+o.getSal());

System.out.println();

}

break;

case 4:

TreeSet<Emp1> tset3 = new TreeSet<Emp1>(new SCompare());

tset3.add(new Emp1(1,"Trainee","Sam",18000.0));

tset3.add(new Emp1(2,"Manager","Rob",32000.0));

tset3.add(new Emp1(3,"Analyst","Tom",21000.0));

System.out.println(" Increasing Order with the Salary : ");

for(Emp1 o : tset3)

{

System.out.print(o.getId()+","+o.getDept()+","+o.getName()+","+o.getSal());

System.out.println();

}

break;

}

}

}

2. import java.util.\*;

public class Duplicate {

public static void main(String [] args) {

int arr[] = {2,7,0,5,2,6,1,4};

ArrayList<Integer> a = new ArrayList<>();

HashSet<Integer> hs = new HashSet<>();

for(int i : arr) {

if(!hs.contains(i)) {

a.add(i);

hs.add(i);

}

}

for (int i : a) {

System.out.print(i + " ");

}

}

}

3. package com.java.collections;

import java.util.\*;

import java.util.Set;

import java.util.TreeMap;

class contact{

int phnum;

String Name;

String Email;

String Gender;

public contact(int phnum,String name,String email,String gender) {

this.phnum=phnum;

this.Name=name;

this.Email=email;

this.Gender=gender;

}

public String toString() {

return this.phnum +" "+this.Name +" "+this.Email+" "+this.Gender;

}

}

class Sortbyphnum implements Comparator<contact>{

public int compare(contact a,contact b) {

return b.phnum-a.phnum;

}

}

public class TreeMapDemo {

static void main() {

//public static void main(String[] args) {

// TODO Auto-generated method stub

TreeMap<contact,Integer> treeMap= new TreeMap<contact,Integer>(new Sortbyphnum());

treeMap.put(new contact(1234567, "swapna",swapna@gmail.com,"female"),3); treeMap.put(new contact(4567890, "kittu",kittu@gmail.com,"male"),4); treeMap.put(new contact(1112131, "nari",nari@gmail.com,"male"),2); treeMap.put(new contact(1516171, "ravi",ravi@gmail.com,"male"),1);

System.out.println("treeMap : " + treeMap + "\n");

//a.fetch all keys and print them

Set<contact> set= treeMap.keySet();

for(contact key:set) {

System.out.println(key);

}

//b.fetch all values and print them

System.out.println("fetching all values:"+treeMap.values());

}

public static void main(String[] args) {

// TODO Auto-generated method stub

//c.print all key-value pairs

System.out.println("treeMap using");

main();

}

}

4. Import java.time.LocalDate;

Public class Leapyear {

Public static void main (String[] args) {

// TODO Auto- generated method stub

LocalDate cal1 = LocalDate.of(1998, 11, 10);

LocalDate cal2 = LocalDate.of (2000, 11, 25);

LinkedList< LocalDate> cellist = new LinkedList<LocalDate>();

callist.add(cal1);

cellist.add(cal2);

for(LocalDate c: cellist)

{

String fDate = c.format(DateTimeFormatter.ofpattern(“dd-mm-yyyy));

If(c.isLeapyear())

{

System.out.println(“ Your Date of Birth “+ fDate + “ and it was a leapyear);

}

Else

{

System.out.println(“ Your Date of Birth “+ fDate + “ and it was not a leapyear);

}

}

}