1. **Given a tree map <long,contact> which has phone number and contact details and also sorted in descending order.**

package tree;

import java.util.Collections;

import java.util.Map;

import java.util.Set;

import java.util.TreeMap;

import tree.contact.gender;

public class tree1 {

public static void main(String[] args)

{

Map<Long,contact> map=new TreeMap<Long,contact>();

contact c1=new contact((long)89034567,"OM ","OM @.com",gender.m);

contact c2=new contact((long)12345679,"RAJ","RAJ@.com",gender.m);

contact c3=new contact((long)67864747,"JOE ","JOE @.com",gender.Fe);

contact c4=new contact((long)45754757,"TOM","TOM @.com",gender.m);

map.put((long)89034567, c1);

map.put((long)12345679, c2);

map.put((long)67864747, c3);

map.put((long)45754757, c4);

for(Map.Entry<Long, contact> entry:map.entrySet()){

Long key=entry.getKey();

contact c=entry.getValue();

System.out.println(key+" Details:");

System.out.println(c.phoneno+" "+c.name+" "+c.email+" "+c.g);

}

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

System.out.println("After Sorted:");

Map<Long,contact> sortedMapDesc = new TreeMap<>(

Collections.reverseOrder());

sortedMapDesc.putAll(map);

for(Map.Entry<Long,contact> entry1 : sortedMapDesc.entrySet())

{

Long key=entry1.getKey();

contact c8=entry1.getValue();

System.out.println(key+" Details:");

System.out.println(c8.phoneno+" "+c8.name+" "+c8.email+" "+c8.g);

}

}

}

package tree;

import java.util.EnumSet;

public class contact {

long phoneno;

String name,email;

public enum gender {Fe,m}

gender g;

/\*\*

\* @paOM phoneno

\* @paOM name

\* @paOM email

\* @paOM g

\*/

public contact(long phoneno, String name, String email, gender g) {

super();

this.phoneno = phoneno;

this.name = name;

this.email = email;

this.g = g;

}

public long getPhoneno() {

return phoneno;

}

public void setPhoneno(long phoneno) {

this.phoneno = phoneno;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public gender getG() {

return g;

}

public void setG(gender g) {

this.g = g;

}

}

Output:

12345679 Details:

12345679 RAJ RAJ@.com m

45754757 Details:

45754757 TOM TOM@.com m

67864747 Details:

67864747 JOE JOE@.com Fe

89034567 Details:

89034567 OM OM@.com m

............

After Sorted:

89034567 Details:

89034567 OM OM@.com m

67864747 Details:

67864747 JOE JOE@.com Fe

45754757 Details:

45754757 TOM TOM@.com m

12345679 Details:

12345679 RAJ RAJ@.com m

**2.Write an application to store 10 unique objects. In case there is an attempt to add duplicate product, it should be silently rejected.**

**package** collections;

**import** java.util.TreeSet;

**public** **class** Addproduct

{

**public** **static** **void** main(String[] args)

{

TreeSet<String> t=**new** TreeSet<String>();

t.add("IPhone");

t.add("OnePlus");

t.add("JOEsung");

t.add("Nokia");

t.add("Redme");

t.add("Realme");

t.add("Vivo");

t.add("Oppo");

t.add("Micromax");

t.add("Lava");

//duplicate

t.add("IPhone");

System.***out***.println("TreeSet for MOBILE PHONE Involves:");

**for**(String temp:t)

{

System.***out***.println(temp);

}

}

}

Output:

TreeSet for MOBILE PHONE Involves:

IPhone

Lava

Micromax

Nokia

OnePlus

Oppo

Realme

Redme

JOEsung

Vivo

**3. Store the employee objects in an TreeSet**

import java.util.TreeSet;

import java.util.Comparator;

import java.util.Scanner;

public class emp1 {

public static void main(String[] args)

{

TreeSet<Emp> name1=new TreeSet<Emp>(new nameComp());

TreeSet<Emp> id1=new TreeSet<Emp>(new idComp());

TreeSet<Emp> dep1=new TreeSet<Emp>(new depComp());

TreeSet<Emp> sal1=new TreeSet<Emp>(new salComp());

name1.add(new Emp(1,"OM","seadmin",2500.23));

name1.add(new Emp(3,"ANJU","SE",4399.23));

name1.add(new Emp(4,"SAM","HR",3422.78));

name1.add(new Emp(8,"RAJ","gmadmin",2500.23));

name1.add(new Emp(5,"JOE","admin",3400.23));

id1.add(new Emp(1,"OM","seadmin",2500.23));

id1.add(new Emp(3,"ANJU","SE",4399.23));

id1.add(new Emp(4,"SAM","HR",3422.78));

id1.add(new Emp(8,"RAJ","gmadmin",2500.23));

id1.add(new Emp(5,"JOE","admin",3400.23));

dep1.add(new Emp(1,"OM","seadmin",2500.23));

dep1.add(new Emp(3,"ANJU","SE",4399.23));

dep1.add(new Emp(4,"SAM","HR",3422.78));

dep1.add(new Emp(8,"RAJ","gmadmin",2500.23));

dep1.add(new Emp(5,"JOE","admin",3400.23));

sal1.add(new Emp(1,"OM","seadmin",2500.23));

sal1.add(new Emp(3,"ANJU","SE",4399.23));

sal1.add(new Emp(4,"SAM","HR",3422.78));

sal1.add(new Emp(8,"RAJ","gmadmin",2500.23));

sal1.add(new Emp(5,"JOE","admin",3400.23));

int s;

Scanner sc=new Scanner(System.in);

System.out.println("Enter:");

s=sc.nextInt();

switch(s)

{

case 1: for(Emp e:name1)

{

System.out.println(e);

}

break;

case 2: for(Emp e1:id1)

{

System.out.println(e1);

}

break;

case 3:for(Emp e2:dep1)

{

System.out.println(e2);

}

break;

case 4:for(Emp e3:sal1)

{

System.out.println(e3);

}

break;

}

}

}

import java.util.Comparator;

class nameComp implements Comparator<Emp>{

public int compare(Emp e1, Emp e2)

{

return e1.getName().compareTo(e2.getName());

}

}

import java.util.Comparator;

public class depComp implements Comparator<Emp>{

public int compare(Emp e1, Emp e2)

{

return e1.getDept().compareTo(e2.getDept());

}

}

import java.util.Comparator;

public class idComp implements Comparator<Emp> {

public int compare(Emp e1, Emp e2)

{

if(e1.getId()>e2.getId())

return 1;

else

{

return -1;

}

}

}

**import** java.util.Comparator;

**public** **class** idComp **implements** Comparator<Emp> {

**public** **int** compare(Emp e1, Emp e2)

{

**if**(e1.getId()>e2.getId())

**return** 1;

**else**

{

**return** -1;

}

}

}

**public** **class** Emp {

**private** **int** id;

**private** String name,dept;

**private** **double** salary;

/\*\*

\* **@paOM** id

\* **@paOM** name

\* **@paOM** dept

\* **@paOM** salary

\*/

**public** Emp(**int** id, String name, String dept, **double** salary) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.dept = dept;

**this**.salary = salary;

}

**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** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** String toString()

{

**return** "Id:"+**this**.id +" " +"name:"+**this**.name+" " +"dept:"+**this**.dept+" " +"sal:"+**this**.salary ;

}

}

Output:

Enter:

2

Id:1 name:OM dept:seadmin sal:2500.23

Id:3 name:ANJU dept:SE sal:4399.23

Id:4 name:SAM dept:HR sal:3422.78

Id:5 name:JOE dept:admin sal:3400.23

Id:8 name:RAJ dept:gmadmin sal:2500.23

**4. Give LinkedList of Objects representing date of birth’s. Print the date along with message: Your date of Birth is DD-MM\_YYY, and it(was or was not) a leap year.**

**package** collections;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeFormatter;

**import** java.util.LinkedList;

**public** **class** Leapyear

{

**public** **static** **void** main(String[] args)

{

LocalDate date1=LocalDate.*of*(2004,02,10);

LocalDate date2=LocalDate.*of*(2020,10,20);

LocalDate date3=LocalDate.*of*(1998,06,14);

LinkedList<LocalDate> list=**new** LinkedList<LocalDate>();

list.add(date1);

list.add(date2);

list.add(date3);

**for**(LocalDate l: list)

{

String printDate= l.format(DateTimeFormatter.*ofPattern*("dd-MM-YYYY"));

**if**(l.isLeapYear())

{

System.***out***.println("Your Date of birth is:"+printDate+"and it was a Leap Year");

}

**else**

{

System.***out***.println("Your Date of Birth is:"+printDate+ "and it was not a Leap Year");

}

}

}

}

Output:

Your Date of birth is:10-02-2004and it was a Leap Year

Your Date of birth is:20-10-2020and it was a Leap Year

Your Date of Birth is:14-06-1998and it was not a Leap Year