1. **import** java.util.Set;

**import** java.util.Collection;

**import** java.util.Collections;

**import** java.util.Iterator;

**import** java.util.Map;

**import** java.util.TreeMap;

**public** **class** Contact {

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

TreeMap<Long, contacts> treeMap = **new** TreeMap<>(Collections.*reverseOrder*());

contacts c1 = **new** contacts("Giri", "naldjqiwj@jlsdk.com", contacts.Gender.***MASCULINE***);

contacts c2 = **new** contacts("Malli", "kheuhrqoup@jlsdk.com", contacts.Gender.***FEMININE***);

contacts c3 = **new** contacts("Nandu", "ioueyiu2uo@jlsdk.com", contacts.Gender.***FEMININE***);

contacts c4 = **new** contacts("Sai", "oiiheioheoi@jlsdk.com", contacts.Gender.***MASCULINE***);

contacts c5 = **new** contacts("Kate", "ksdjuehiue@jlsdk.com", contacts.Gender.***NEUTRAL***);

contacts c6 = **new** contacts("Raghu", "uyrqwhfbrb@jlsdk.com", contacts.Gender.***MASCULINE***);

treeMap.put(9908704519L, c1);

treeMap.put(9999402029L, c2);

treeMap.put(8983849888L, c3);

treeMap.put(7880388337L, c4);

treeMap.put(6784019849L, c5);

treeMap.put(9874262782L, c6);

System.***out***.println("Printing Keys in the TreeMap");

Set<Long> keys = treeMap.keySet();

Iterator<Long> itr = keys.iterator();

**while**(itr.hasNext()) {

System.***out***.println(itr.next());

}

System.***out***.println("/n/nPrinting Values in the TreeMap");

Set values = treeMap.entrySet();

Iterator itr2 = values.iterator();

**while**(itr2.hasNext()) {

Map.Entry e = (Map.Entry)itr2.next();

System.***out***.println(e.getValue());

}

System.***out***.println("/n/nPrinting the Key-Value pairs in the TreeMap");

**for**(Map.Entry<Long, contacts> entry: treeMap.entrySet())

{

System.***out***.println(entry.getKey()+ " - " +entry.getValue());

}

// System.out.println(treeMap);

}

}

**public** **class** contacts {

**public** **enum** Gender {***MASCULINE***,***FEMININE***, ***NEUTRAL***};

**private** String name;

**private** String email;

**private** Gender gender;

**public** contacts(String name,String email, Gender gender) {

**super**();

**this**.name = name;

**this**.email = email;

**this**.gender = gender;

}

**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 getGender() {

**return** gender;

}

**public** **void** setGender(Gender gender) {

**this**.gender = gender;

}

**public** String toString() {

**return** " { name: " +name+ " , email: " +email+ " , Gender: " +gender+ " }";

}

}

1. **public** **class** Products {

**private** **int** product\_Id;

**private** String product\_name;

**public** Products(**int** product\_Id, String product\_name) {

**super**();

**this**.product\_Id = product\_Id;

**this**.product\_name = product\_name;

}

**public** **int** getProduct\_Id() {

**return** product\_Id;

}

**public** **void** setProduct\_Id(**int** product\_Id) {

**this**.product\_Id = product\_Id;

}

**public** String getProduct\_name() {

**return** product\_name;

}

**public** **void** setProduct\_name(String product\_name) {

**this**.product\_name = product\_name;

}

**public** String toString() {

**return** "Product: { ID: " +product\_Id+ ", name: " +product\_name+ " }";

}

}

**import** java.util.HashSet;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**public** **class** Product {

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

List<Products> al = **new** ArrayList<>();

al.addAll(Arrays.*asList*(

**new** Products(1001, "Cookie"),

**new** Products(1002, "Candy"),

**new** Products(1003, "IceCream"),

**new** Products(1004, "Chocolate"),

**new** Products(1005, "Pastries")));

HashSet<Products> set = **new** HashSet<Products>(al);

set.addAll(Arrays.*asList*(**new** Products(1006, "Cake")));

Iterator<Products> itr = set.iterator();

**while**(itr.hasNext()) {

System.***out***.println(itr.next());

}

}

}

1. **import** java.util.Comparator;

**import** java.util.Scanner;

**import** java.util.TreeSet;

**class** employee {

**int** id,salary;

String name,department;

**public** employee(**int** id,**int** salary,String name,String department)

{

**this**.id=id;

**this**.salary=salary;

**this**.name=name;

**this**.department=department;

}

**public** **int** getId() {

**return** id;

}

**public** **int** getSalary() {

**return** salary;

}

**public** String getName() {

**return** name;

}

**public** String getDepartment() {

**return** department;

}

}

**class** Idcompare **implements** Comparator<employee>{

//@Override

**public** **int** compare(employee o1, employee o2) {

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

}

}

**class** Salcompare **implements** Comparator<employee>{

@Override

**public** **int** compare(employee o1, employee o2) {

**return** o1.getSalary()-o2.getSalary();

}

}

**class** Namecompare **implements** Comparator<employee>{

@Override

**public** **int** compare(employee o1, employee o2) {

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

}

}

**class** Deptcompare **implements** Comparator<employee>{

@Override

**public** **int** compare(employee o1, employee o2) {

**return** o1.getDepartment().compareTo(o2.getDepartment());

}

}

**public** **class** Employee1{

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

{

@SuppressWarnings("resource")

Scanner e = **new** Scanner(System.***in***);

System.***out***.println("Select one option to sort the list:\n\n1. Id\n2.Salary\n3.Name\n4.Department\n\n");

**int** option = e.nextInt();

**if**(option == 1)

{

TreeSet<employee> set1 = **new** TreeSet<employee>(**new** Idcompare());

employee e1 = **new** employee(1, 20000 , "Joy", "marketing");

employee e2=**new** employee(2, 10000, "Arjun", "IT");

employee e3=**new** employee(3,5000,"Ram", "HR");

set1.add(e1);

set1.add(e2);

set1.add(e3);

System.***out***.println("Sorted by ID:");

**for**(employee o :set1) {

System.***out***.println("[ Id: "+o.id+ ", salary: "+o.salary+" ,name: "+o.name+", department: "+o.department+" ]");

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

}

}

**else** **if**(option == 2)

{

TreeSet<employee> set1 = **new** TreeSet<employee>(**new** Salcompare());

employee e1 = **new** employee(1, 20000 , "Joy", "marketing");

employee e2=**new** employee(2, 10000, "Arjun", "IT");

employee e3=**new** employee(3,5000,"Ram", "HR");

set1.add(e1);

set1.add(e2);

set1.add(e3);

System.***out***.println("Sorted by Salary:");

**for**(employee o :set1) {

System.***out***.println("[ Id: "+o.id+ ", salary: "+o.salary+" ,name: "+o.name+", department: "+o.department+" ]");

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

}

}

**else** **if**(option == 3)

{

TreeSet<employee> set1 = **new** TreeSet<employee>(**new** Namecompare());

employee e1 = **new** employee(1, 20000 , "Joy", "marketing");

employee e2=**new** employee(2, 10000, "Arjun", "IT");

employee e3=**new** employee(3,5000,"Ram", "HR");

set1.add(e1);

set1.add(e2);

set1.add(e3);

System.***out***.println("Sorted by Name:");

**for**(employee o :set1) {

System.***out***.println("[ Id: "+o.id+ ", salary: "+o.salary+" ,name: "+o.name+", department: "+o.department+" ]");

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

}

}

**else** **if**(option == 4)

{

TreeSet<employee> set1 = **new** TreeSet<employee>(**new** Deptcompare());

employee e1 = **new** employee(1, 20000 , "Joy", "marketing");

employee e2=**new** employee(2, 10000, "Arjun", "IT");

employee e3=**new** employee(3,5000,"Ram", "HR");

set1.add(e1);

set1.add(e2);

set1.add(e3);

System.***out***.println("Sorted by Department:");

**for**(employee o :set1) {

System.***out***.println("[ Id: "+o.id+ ", salary: "+o.salary+" ,name: "+o.name+", department: "+o.department+" ]");

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

}

}

**else** {

System.***out***.println("Invalid Input");

}

}

}

1. **import** java.time.LocalDate;

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

**import** java.util.LinkedList;

**import** java.util.List;

**public** **class** Leap\_Date {

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

LocalDate l1 = LocalDate.*of*(2012, 12, 12);

LocalDate l2 = LocalDate.*of*(1997, 02, 03);

LocalDate l3 = LocalDate.*of*(1998, 02, 27);

LocalDate l4 = LocalDate.*of*(1972, 05, 02);

LinkedList<LocalDate> list = **new** LinkedList<LocalDate>(List.*of*(l1, l2, l3, l4));

**for**(LocalDate a:list) {

String printDate = a.format(DateTimeFormatter.*ofPattern*("dd-MM-yyyy"));

**if**(a.isLeapYear()) {

System.***out***.println(printDate + " is a Leap Year");

}

**else**

{

System.***out***.println(printDate+ " is Not a Leap Year");

}

}

}

}