1.

package Generics;

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

public class Program1 {

public static void main(String[] args) {

Map<Long,Contact> m=new HashMap<>();

m.put(10012002l,new Contact(99887766,"Unkown","anonymous@abc.com",Contact.Gender.Male));

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

System.out.println("Keys:-");

for(long i:keys)

{

System.out.println(i);

}

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

System.out.println("Values:-");

for(long i:keys)

{

System.out.println(m.get(i));

}

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

System.out.println("Key and Value-");

System.out.println(m);

}

}

class Contact{

long phoneNo;

String name;

String email;

private Contact.Gender gen;

enum Gender{

Male,Female,Others

}

public Contact(long phoneNo, String name, String email ,Gender gen) {

super();

this.phoneNo = phoneNo;

this.name = name;

this.email = email;

this.gen=gen;

}

@Override

public String toString() {

return "Contact [phoneNo=" + phoneNo + ", name=" + name + ", email=" + email + ", gen=" + gen + "]";

}

}

2.

package Generics;

import java.util.Iterator;

import java.util.HashSet;

public class Program2 {

public static void main(String[] args) {

HashSet<Object> rj=new HashSet<>();

rj.add(-5);

rj.add(100);

rj.add(null);

rj.add("Hi");

rj.add(0);

rj.add(75);

rj.add(-80);

rj.add(-5); //will be negleted for being duplicate

rj.add("Bye");

rj.add("3.14");

System.out.println(rj);

}

}

3.

package Generics;

import java.util.Comparator;

import java.util.Scanner;

import java.util.TreeSet;

public class Prgm3 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Choose between following options:");

System.out.println("1.ID|2.Name|3.Department|4.Salary");

int choose=sc.nextInt();

TreeSet<Employee> emp=null;

switch(choose)

{

case 1: emp=new TreeSet<>(new IdComp());

break;

case 2: emp=new TreeSet<>(new NameComp());

break;

case 3: emp=new TreeSet<>(new DeptComp());

break;

case 4: emp=new TreeSet<>(new SalComp());

break;

default:System.out.println("Invalid Option");

}

emp.add(new Employee(10,"Naruto","IT",2000));

emp.add(new Employee(11,"Luffy","Management",2200));

emp.add(new Employee(12,"Goku","HR",1800));

emp.add(new Employee(13,"Madara","Management",1900));

emp.add(new Employee(14,"Itachi","HR",2100));

emp.add(new Employee(16,"Vegeta","Training",2400));

emp.add(new Employee(15,"Karma","IT",2000));

emp.add(new Employee(19,"Leleouch","IT",2300));

emp.add(new Employee(17,"Minato","HR",1900));

emp.add(new Employee(18,"Gohan","Management",1700));

for(Employee e:emp)

{

System.out.println(e);

}

}

}

class IdComp implements Comparator<Employee> {

@Override

public int compare(Employee o1, Employee o2) {

if(o1.getId()>o2.getId())

{

return 1;

}

else

{

return -1;

}

}

}

class NameComp implements Comparator<Employee>{

@Override

public int compare(Employee o1, Employee o2) {

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

}

}

class DeptComp implements Comparator<Employee>{

@Override

public int compare(Employee o1, Employee o2) {

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

}

}

class SalComp implements Comparator<Employee>{

@Override

public int compare(Employee o1, Employee o2) {

if(o1.getSalary()>o2.getSalary())

{

return 1;

}

else

{

return -1;

}

}

}

class Employee{

int id;

String name;

String department;

int 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 getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public int getSalary() {

return salary;

}

public void setSalary(int salary) {

this.salary = salary;

}

public Employee(int id, String name, String department, int salary) {

super();

this.id = id;

this.name = name;

this.department = department;

this.salary = salary;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", department=" + department + ", salary=" + salary + "]";

}

}

4.

package Generics;

import java.time.LocalDate;

import java.time.Month;

import java.util.LinkedList;

import java.util.List;

public class Program4{

public static void main(String[] args) {

List<LocalDate> ll=new LinkedList<>();

ll.add(LocalDate.of(2000,Month.APRIL,23));

ll.add(LocalDate.of(2004,Month.FEBRUARY,29));

ll.add(LocalDate.of(2001,Month.JANUARY,10));

ll.add(LocalDate.of(2000,Month.JULY,28));

ll.add(LocalDate.of(2003,Month.JUNE,12));

ll.add(LocalDate.of(2005,Month.DECEMBER,21));

ll.add(LocalDate.of(2006,Month.OCTOBER,30));

ll.add(LocalDate.of(2008,Month.MARCH,31));

ll.add(LocalDate.of(2009,Month.SEPTEMBER,20));

ll.add(LocalDate.of(2004,Month.NOVEMBER,8));

for(LocalDate ld: ll)

{

if(ld.isLeapYear())

{

System.out.println("Your date of birth is "+ld+" and it was leap year");

}

else

{

System.out.println("Your date of birth is "+ld+" and it was not leap year");

}

}

}

}