**Collections Assignment**

1. **import** java.util.\*;

**public** **class** QTree

{

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

{

TreeMap<Long, String> treeMap = **new** TreeMap<>();

treeMap.put(9893284442L, "Priyanshi");

treeMap.put(9893284402L, "Sakshi");

treeMap.put(9893284403L, "Ishika");

treeMap.put(9893284405L, "Anamika");

treeMap.put(9893284406L, "Aditya");

Set keys = treeMap.keySet();

Iterator i = keys.iterator();

**while** (i.hasNext())

{

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

}

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

Collection getValues = treeMap.values();

i = getValues.iterator();

**while** (i.hasNext())

{

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

}

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

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

}

}

1. **import** java.util. TreeSet;

**public** **class** CollectionsQuestion2

{

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

{

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

func.add(**new** String("Cheese"));

func.add(**new** String("Pasta"));

func.add(**new** String("Chips"));

func.add(**new** String("Maggi"));

func.add(**new** String("Chocolates"));

func.add(**new** String("Maggi"));

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

}

}

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** CollectionsQuestion3{

**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(3, 20000 , "Priyanshi", "marketing");

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

employee e3=**new** employee(1,50000,"Ishika", "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 , "Priyanshi", "marketing");

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

employee e3=**new** employee(3,5000,"Ishika", "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 , "Priyanshi", "marketing");

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

employee e3=**new** employee(3,5000,"Ishika", "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 , "Priyanshi", "marketing");

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

employee e3=**new** employee(3,5000,"Ishika", "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;

**public** **class** FourthQuestionCollections {

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

LocalDate date1 = LocalDate.*of*(2000, 12, 23);

LocalDate date2 = LocalDate.*of*(1998, 2, 27);

LocalDate date3 = LocalDate.*of*(2001, 12, 23);

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");

}

}

}

}