JAVA TRAINING CENTER

Java Training Center

(No.1 in Training & placement)

java.util WorkBook

Master the Content…

GROWTH UNBOUND

W.B-1

Author

Som Prakash Rai

JAVA TRAINING CENTER

Topic

Jtc 1: Example using Collection interface.

```
import java.util.*;
public class Jtc1 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[])    {
ArrayList al=new ArrayList();
System.out.println(al);
System.out.println(al.size());
System.out.println(al.isEmpty());

al.add(new Integer(99));
al.add("som");                    al.add("123");
al.add("som@jtc");                al.add(new Double(999.99));
al.add(new Long(11111));
System.out.println(al);
System.out.println(al.size());
System.out.println(al.isEmpty());
Object[] obj=al.toArray();
System.out.println(obj.length);
System.out.println("printing directly");
for(int i=0;i<obj.length;i++){
System.out.println(obj[i]);
}

System.out.println("checking and casting");
for(int i=0;i<obj.length;i++){
Object o=obj[i];
if(o instanceof String){
String str=(String)o;
System.out.println(str);
}else if(o instanceof Integer){
Integer in=(Integer)o;
System.out.println(in);
}else if(o instanceof Double){
Double d=(Double)o;
System.out.println(d);
}
```

```
}
Iterator it=al.iterator();
while(it.hasNext()){
Object o=it.next();
System.out.println(o);
}
System.out.println(al.contains("som"));
System.out.println(al.contains("jtc"));
System.out.println(al);
al.add("123");
System.out.println(al);
al.remove("123");
System.out.println(al);

ArrayList al1=new ArrayList();
al1.add("aa");
al1.add("bb");
al1.add("cc");
al1.add("dd");
System.out.println(al1);
System.out.println(al);
System.out.println(al.size());
//al.add(al1);
al.addAll(al1);
System.out.println(al);
System.out.println(al.size());
System.out.println(al.containsAll(al1));
System.out.println(al);
/*
al.removeAll(al1);
System.out.println(al);
*/
al.retainAll(al1);
System.out.println(al);
al.clear();
System.out.println(al);
}
}
```

Jtc 2: Example using List interface.

```
import java.util.*;
/*
```

```
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc2 {
public static void main(String as[]){
ArrayList al=new ArrayList();
al.add(new Integer(99));
al.add("som");
al.add("123");
al.add("som@jtc");
al.add("som");
System.out.println(al);
al.add(0,"aaaa");
al.add(2,"bbbb");
System.out.println(al);
al.remove(4);
System.out.println(al);
System.out.println(al.get(2));
System.out.println(al.indexOf("som"));
System.out.println(al.lastIndexOf("som"));
al.set(0,"jtc");
System.out.println(al);
List list=al.subList(1,4);
System.out.println(list);
System.out.println("forward order");
ListIterator li=al.listIterator();
while(li.hasNext()){
System.out.println(li.next());
}
System.out.println("reverse order");
while(li.hasPrevious()){
System.out.println(li.previous());
}
}
}
```

Jtc 3: Example using Vector and LinkedList .

```
import java.util.*;
public class Jtc3 {
/*
```

```
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[]){
Vector v=new Vector();
v.add("99");
v.add("som");
v.addElement("som@jtc");
v.addElement("som");
System.out.println(v);
Enumeration e=v.elements();
System.out.println("forward order");
while(e.hasMoreElements()){
System.out.println(e.nextElement());
}
Iterator it=v.iterator();
System.out.println("forward order");
while(it.hasNext()){
System.out.println(it.next());
}
LinkedList ll=new LinkedList(v);
ll.addFirst("11");
ll.addLast("22");
System.out.println(ll);
System.out.println(ll.getFirst());
System.out.println(ll.getLast());
}
}
```

Jtc 4: Example using HashSet,TreeSet,LinkedHashSet .

```
import java.util.*;
public class Jtc4 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[]){
```

```
HashSet hs=new HashSet();
hs.add(new Integer(99));
System.out.println(hs.add("som"));
hs.add("som@jtc");
System.out.println(hs.add("som"));
System.out.println(hs);

TreeSet ts=new TreeSet();
//ts.add(new Integer(99));
System.out.println(ts.add("som"));
ts.add("som@jtc");
System.out.println(ts.add("som"));
ts.add("aaaa");
ts.add("cccc");
ts.add("bbbb");
System.out.println(ts);

LinkedHashSet lhs=new LinkedHashSet();
lhs.add(new Integer(99));
System.out.println(lhs.add("som"));
lhs.add("som@jtc");
System.out.println(lhs.add("som"));
System.out.println(lhs);
}
}
```

Jtc 5: Example using HashMap .

```
import java.util.*;
public class Jtc5 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[])   {
HashMap hm=new HashMap();
System.out.println(hm);
System.out.println(hm.size());
System.out.println(hm.isEmpty());
hm.put("sid",new Integer(99));
hm.put("sname","som");
```

```
hm.put("email","abc");
System.out.println(hm);
System.out.println(hm.size());
System.out.println(hm.isEmpty());
System.out.println(hm.containsKey("sid"));
System.out.println(hm.containsKey("sid1"));
System.out.println(hm.containsValue("som"));
System.out.println(hm.containsValue("som1"));
System.out.println(hm.get("sname"));
System.out.println(hm);
hm.put("sname","rai");
System.out.println(hm);
hm.put("sname1","rai");
System.out.println(hm);
hm.put(null,"rai");
hm.put(new Double(999.99),null);
System.out.println(hm);
//hm.remove("xx");
System.out.println(hm);
Collection col=hm.values();
System.out.println(col);
System.out.println("using keySet()");
Set s=hm.keySet();
System.out.println(s);
Iterator it=s.iterator();
while(it.hasNext()){
Object o1=it.next();
String key="";
if(o1!=null){
key=o1.toString();
}
else{
key=null;
}
Object o2=hm.get(key);
String val="";
if(o2!=null){
val=o2.toString();
}
else{
val=null;
}
```

```
System.out.println(key+"..."+val);
}
System.out.println("using entrySet()");
Set es=hm.entrySet();
Iterator it1=es.iterator();
while(it1.hasNext()){
Object o=it1.next();
Map.Entry me=(Map.Entry)o;
Object o1=me.getKey();
String key="";
if(o1!=null){
key=o1.toString();
}
else{
key=null;
}
Object o2=me.getValue();
String val="";
if(o2!=null){
val=o2.toString();
}
else{
val=null;
}
System.out.println(key+"..."+val);
}
}
}
```

Jtc 6: Example using LinkedHashMap,TreeMap,Hashtable .

```
import java.util.*;
public class Jtc6 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[])
{
LinkedHashMap hm=new LinkedHashMap();
hm.put("sid",new Integer(99));
```

```
hm.put("sname","Som");
hm.put("fee",new Double(9000.99));
System.out.println(hm);
hm.put("x","10");
hm.put(new Integer(99),"10");
System.out.println(hm);
hm.put("x","20");
System.out.println(hm);
hm.put(null,null);
System.out.println(hm);

Hashtable ht=new Hashtable();
ht.put("sid",new Integer(99));
ht.put("sname","Som");
ht.put("fee",new Double(9000.99));
System.out.println(ht);
ht.put("x","10");
ht.put(new Integer(99),"10");
System.out.println(ht);
ht.put("x","20");
System.out.println(ht);
/*
ht.put("z",null);
System.out.println(ht);
*/
/*
ht.put(null,"s");
System.out.println(ht);
*/
TreeMap tm=new TreeMap();
tm.put("sid",new Integer(99));
tm.put("sname","Som");
tm.put("fee",new Double(9000.99));
System.out.println(tm);
tm.put("x","10");
//tm.put(new Integer(99),"10");
System.out.println(tm);
tm.put("x","20");
System.out.println(tm);
tm.put("z",null);
System.out.println(tm);
/*
```

```
tm.put(null,"s");
System.out.println(tm);
*/
}
}
```

Jtc 7: Example using Comparator and Comparable .

```
import java.util.*;
class Student implements Comparable{
int sid;
String sname;
String email;
Student(int sid,String sname,String email){
this.sid=sid;
this.sname=sname;
this.email=email;
}
public String toString(){
return ""+sid+"\t"+sname+"\t"+email;
}
public boolean equals(Object o){
Student s=(Student)o;
if(this.sid==s.sid)
return true;
return false;
}
public int compareTo(Object o){
Student s=(Student)o;
return this.sid-s.sid;
}
}
class SnameComparator implements Comparator{
public int compare(Object o1,Object o2){
Student s1=(Student)o1;
Student s2=(Student)o2;
return s1.sname.compareTo(s2.sname);
}
}
class EmailComparator implements Comparator{
public int compare(Object o1,Object o2){
Student s1=(Student)o1;
```

```
Student s2=(Student)o2;
return s1.email.compareTo(s2.email);
}
}
class Jtc7{
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String as[]){
ArrayList al=new ArrayList();
Student s1=new Student(22,"dd","cc@jtc");
Student s2=new Student(44,"aa","bb@jtc");
Student s3=new Student(11,"cc","dd@jtc");
Student s4=new Student(33,"bb","aa@jtc");
al.add(s1);     al.add(s2);     al.add(s3);     al.add(s4);
System.out.println("No Sorting");
Iterator it=al.iterator();
while(it.hasNext()){
Student s=(Student)it.next();
System.out.println(s);
}
System.out.println("Sorting by Sid");
Collections.sort(al);
it=al.iterator();
while(it.hasNext()){
Student s=(Student)it.next();
System.out.println(s);
}
System.out.println("Sorting by Sname");
Collections.sort(al,new SnameComparator());
it=al.iterator();
while(it.hasNext()){
Student s=(Student)it.next();                System.out.println(s);
}
System.out.println("Sorting by Email");
Collections.sort(al,new EmailComparator());
it=al.iterator();
while(it.hasNext()){
Student s=(Student)it.next();                System.out.println(s);
```

```
}
}
}
```

Jtc 8: Example using Collections .

```
import java.util.*;
public class Jtc8 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */

public static void main(String[] args) {
ArrayList al=new ArrayList();
al.add("bb");          al.add("cc");       al.add("dd");       al.add("aa");
//al.add(new Integer(99));
System.out.println(al);
Collections.sort(al);
System.out.println(al);
Collections.reverse(al);
System.out.println(al);
Collections.shuffle(al);
System.out.println(al);
Collections.rotate(al,1);
System.out.println(al);
Collections.swap(al,1,3);
System.out.println(al);
System.out.println(Collections.max(al));
System.out.println(Collections.min(al));
Collections.sort(al);
System.out.println(Collections.binarySearch(al,"cc"));
Collections.fill(al,"jtc");
System.out.println(al);
Vector v=new Vector();
v.add("99");     v.add("som");     v.addElement("abc");
System.out.println(v);
Enumeration e=v.elements();
List al1=Collections.list(e);
System.out.println(v);
System.out.println(al1);
```

```
al1=Collections.unmodifiableList(al1);
al1.add("11");
}
}
```

Jtc 9: Example using Arrays .

```
import java.util.*;
public class Jtc9 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
int arr[] = { 10, 45, 25, 6, 78, 12, 21 };
int arr1[] = { 10, 45, 25, 6, 78, 12, 21 };
int arr2[] = { 10, 45, 85, 45, 12, 78, 896 };
for (int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + "\t");
}
for (int i = 0; i < arr1.length; i++) {
System.out.print(arr1[i] + "\t");
}
for (int i = 0; i < arr2.length; i++) {
System.out.print(arr2[i] + "\t");
}
System.out.println("\n******* arr After sorting ******");
Arrays.sort(arr);
for (int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + "\t");
}
System.out.println();
System.out.println(Arrays.binarySearch(arr, 6));
System.out.println(Arrays.binarySearch(arr, 9));
System.out.println(Arrays.equals(arr, arr1));
System.out.println(Arrays.equals(arr, arr2));
Arrays.fill(arr, 32);
for (int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + "\t");
}
Object ob[] = { "jtc", "india", "som", "rai", "white", "red" };
```

```
for (int i = 0; i < ob.length; i++) {
System.out.print(ob[i] + "\t");
}
System.out.println();
List list = Arrays.asList(ob);
Iterator it = list.iterator();
while (it.hasNext()) {
System.out.print(it.next()+ "\t");
}
System.out.println();
Arrays.sort(ob);
for (int i = 0; i < ob.length; i++) {
System.out.print(ob[i] + "\t");
}
System.out.println();
}
}
```

Jtc 10: Example using Date and Calendar .

```
import java.util.*;
public class Jtc10 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
Calendar cal = Calendar.getInstance();
Date d = cal.getTime();
System.out.println(d);
System.out.println(cal.get(Calendar.DATE));
System.out.println(cal.get(Calendar.MONTH));
System.out.println(cal.get(Calendar.YEAR));
System.out.println(cal.get(Calendar.HOUR));
System.out.println(cal.get(Calendar.MINUTE));
System.out.println( cal.get(Calendar.SECOND));
System.out.println( cal.get(Calendar.AM_PM));
Calendar cal1 = Calendar.getInstance();
cal.set(Calendar.DATE, 20);
cal.set(Calendar.MONTH, 1);
cal.set(Calendar.YEAR, 2010);
```

```
System.out.println( cal.after(cal1));
System.out.println( cal.before(cal1));
System.out.println(cal.getFirstDayOfWeek());
System.out.println(cal.getTimeInMillis());

Date dt = new Date();
System.out.println(dt);
System.out.println(dt.getDate());
System.out.println(dt.getMonth());
System.out.println((dt.getMonth() + 1));
System.out.println(dt.getYear());
System.out.println((dt.getYear() + 1900));
System.out.println(dt.getHours());
System.out.println(dt.getMinutes());
System.out.println(dt.getSeconds());
System.out.println(dt.getTime());
int day = dt.getDay();
String str = "";
switch (day) {
case 0:        str = "Sunday";          break;
case 1:        str = "Monday";          break;
case 2:        str = "Tuesday";         break;
case 3:        str = "Wednesday";       break;
case 4:        str = "Thrusday";        break;
case 5:        str = "Friday";          break;
case 6:        str = "Saturday";        break;
}
System.out.println(day + "\t" + str);
Date dt1 = new Date(110, 1, 12);
System.out.println(dt1);
System.out.println(dt.after(dt1));
System.out.println(dt.before(dt1));
System.out.println(dt.after(dt));
System.out.println(dt.before(dt));
dt.setDate(10);          dt.setMonth(0);
dt.setYear(110);
dt.setHours(12);
dt.setMinutes(53);
dt.setSeconds(55);
System.out.println("****  After Setting new Date ***\n" + dt);
}
}
```

Jtc 11: Example using Locale .

```
import java.util.*;
public class Jtc11 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
System.out.println(Locale.getDefault());
Locale locales[] = Locale.getAvaiJtcleLocales();
for (int i = 0; i < locales.length; i++) {
System.out.print(locales[i] + ", ");
}
String countries[] = Locale.getISOCountries();
for (int i = 0; i < countries.length; i++) {
System.out.print(countries[i] + ", ");
}
String languagess[] = Locale.getISOLanguages();
for (int i = 0; i < languagess.length; i++) {
System.out.print(languagess[i] + ", ");
}
System.out.println();
Locale loc = new Locale("EN");
System.out.println(loc.getCountry());
System.out.println(loc.getDisplayCountry());
System.out.println(loc.getDisplayLanguage());
System.out.println(loc.getDisplayName());
System.out.println(loc.getVariant());
System.out.println(loc.getDisplayVariant());
Locale loc1 = new Locale("EN", "US");
System.out.println( loc1.getCountry());
System.out.println(loc1.getDisplayCountry());
System.out.println( loc1.getDisplayLanguage());
System.out.println(loc1.getDisplayName());
}
}
```

Jtc 12: Example using StringTokenizer interface .

```
import java.util.*;
public class Jtc12 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
String str = "Welcome to Java Training Center to learn java  and advanve java.
Thankyou";
StringTokenizer token = new StringTokenizer(str);
System.out.println(token.hasMoreTokens());
System.out.println(token.countTokens());

while (token.hasMoreTokens()) {
String str1 = token.nextToken();
System.out.println(str1);
}
System.out.println(token.hasMoreTokens());
System.out.println(token.countTokens());

StringTokenizer token1 = new StringTokenizer(str, "ja");
System.out.println(token1.hasMoreTokens());
System.out.println(token1.countTokens());

while (token1.hasMoreElements()) {
Object obj = token1.nextElement();
System.out.println(obj);
}
}
}
```

Jtc 13: Example using Timer and TimerTask .

```
import java.util.*;
class Reminder {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
```

```
* */
Timer timer;
public Reminder(int seconds) {
timer = new Timer();
timer.schedule(new RemindTask(), seconds * 1000);
}
class RemindTask extends TimerTask {
public void run() {
System.out.println("Time's up!");
timer.cancel();
}
}
}
public class Jtc13 {
public static void main(String args[]) {
System.out.println("About to schedule task.");
new Reminder(5);
System.out.println("Task scheduled.");
}
}
```

Jtc 14: Example using ResourceBundle and properties file .

```
import java.io.*;
import java.util.*;

public class Jtc14 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* *
public static void main(String[] args) throws Exception, IOException {
Properties p = new Properties();
FileInputStream fis = new FileInputStream("data.properties");
p.load(fis);

Enumeration en = p.propertyNames();
while (en.hasMoreElements()) {
String key = (String) en.nextElement();
```

```
System.out.println(key + "\t" + p.getProperty(key));
}
fis.close();

System.out.println("H\tHindi");
System.out.println("E\tEnglish");
System.out.println("Enter Language Code");
char ch = (char) System.in.read();
ch = Character.toUpperCase(ch);
Locale loc = new Locale("en", "IN");
if (ch == 'H')
loc = new Locale("hi", "IN");
else if (ch == 'E')
loc = new Locale("en", "IN");
else
System.out.println("Not valid Option, Reading English Language");

System.out.println("\n---- Resource Bundle ----");
System.out.println(loc);
ResourceBundle rb = ResourceBundle.getBundle("data", loc);
Enumeration en1 = rb.getKeys();
while (en1.hasMoreElements()) {
String ky = (String) en1.nextElement();
System.out.println(ky + "\t" + rb.getString(ky));
}
System.out.println("\n---- Reading specific key value --");
String val = rb.getString("sid");
System.out.println("Value of\t:" + val);
}
}
```

data.properties

```
sid=9876 [ENG]
name=SomPrakash [ENG]
email=som@jtcindia.org [ENG]
phone=45133258 [ENG]
```

data_hi.properties

```
sid=9876 [HND]
name=SomPrakash [HND]
```

email=som@jtcindia.org [HND]
phone=45133258 [HND]

java.util API

```
interface Collection{
   public abstract int hashCode();
   public abstract int size();
   public abstract void clear();
   public abstract boolean isEmpty();
   public abstract Object[] toArray();
   public abstract boolean add(Object);
   public abstract boolean contains(Object);
   public abstract boolean equals(Object);
   public abstract boolean remove(Object);
   public abstract boolean addAll(Collection);
   public abstract boolean containsAll(Collection);
   public abstract boolean removeAll(Collection);
   public abstract boolean retainAll(Collection);
   public abstract Iterator iterator();
}
interface Iterator{
   public abstract void remove();
   public abstract boolean hasNext();
   public abstract Object next();
}
interface ListIterator extends Iterator{
   public abstract int nextIndex();
   public abstract int previousIndex();
   public abstract void remove();
   public abstract boolean hasNext();
   public abstract boolean hasPrevious();
   public abstract Object next();
   public abstract Object previous();
   public abstract void add(Object);
   public abstract void set(Object);
}
interface Enumeration{
   public abstract boolean hasMoreElements();
   public abstract Object nextElement();
}
```

```
interface List extends Collection{
   public abstract Object get(int);
   public abstract Object remove(int);
   public abstract void add(int,Object);
   public abstract int indexOf(Object);
   public abstract int lastIndexOf(Object);
   public abstract boolean addAll(int,Collection);
   public abstract List subList(int,int);
   public abstract ListIterator listIterator();
   public abstract ListIterator listIterator(int);
   public abstract Object set(int,Object);
}
public class ArrayList extends AbstractList implements List,RandomAccess,Cloneable,
Serializable{
   public ArrayList();
   public ArrayList(int);
   public ArrayList(Collection);
   protected void removeRange(int,int);
 }
public class Vector extends AbstractList implements List,RandomAccess,Cloneable,
Serializable{
   public Vector();
   public Vector(int);
   public Vector(Collection);
   public synchronized void removeAllElements();
   public synchronized boolean isEmpty();
   public synchronized void removeElementAt(int);
   protected void removeRange(int,int);
    public synchronized Object firstElement();
   public synchronized Object lastElement();
   public synchronized Object elementAt(int);
   public synchronized void addElement(Object);
   public synchronized boolean removeElement(Object);
   public synchronized void insertElementAt(Object,int);
   public synchronized void setElementAt(Object,int);
   public Enumeration elements();
}
public class LinkedList extends AbstractSequentialList implements List,Cloneable,
Serializable{
   public LinkedList();
   public LinkedList(Collection);
```

```
    public Object getFirst();
    public Object getLast();
    public Object removeFirst();
    public Object removeLast();
    public void addFirst(Object);
    public void addLast(Object);
}
interface Set extends Collection{
    // no new Methods ..all are same as in Collection
}
public class HashSet extends AbstractSet implements Set,Cloneable, Serializable{
    public HashSet();
    public HashSet(int);
    public HashSet(Collection);
    public Iterator iterator();
}
public class TreeSet extends AbstractSet implements SortedSet,Cloneable, Serializable{
    public TreeSet();
    public Object first();
    public Object last();
    public TreeSet(Collection);
    public Comparator comparator();
    public TreeSet(Comparator);
}
public class LinkedHashSet extends HashSet implements Set,Cloneable, Serializable{
    public LinkedHashSet();
    public LinkedHashSet(int);
    public LinkedHashSet(Collection);
}
interface Map{
    public abstract int hashCode();
    public abstract int size();
    public abstract void clear();
    public abstract boolean isEmpty();
    public abstract boolean containsKey(Object);
    public abstract boolean containsValue(Object);
    public abstract boolean equals(Object);
    public abstract Collection values();
    public abstract void putAll(Map);
    public abstract Set entrySet();
    public abstract Set keySet();
    public abstract Object get(Object);
```

```
   public abstract Object remove(Object);
   public abstract Object put(Object,Object);
}
public class HashMap extends AbstractMap implements Map,Cloneable, Serializable{
   public HashMap();
   public HashMap(int);
   public HashMap(Map);
}
public class Hashtable extends Dictionary implements Map,Cloneable,Serializable{
   public Hashtable();
   public Hashtable(int);
   public Hashtable(Map);
}
public class LinkedHashMap extends HashMap{
   public LinkedHashMap();
   public LinkedHashMap(int);
   public LinkedHashMap(Map);
}
public class TreeMap extends AbstractMap implements SortedMap,Cloneable,
Serializable{
   public TreeMap();
   public TreeMap(Map);
   public Object firstKey();
   public Object lastKey();
}
public class Collections extends Object{
   public static Comparator reverseOrder();
   public static void reverse(List);
   public static void shuffle(List);
   public static void sort(List);
   public static void rotate(List,int);
   public static void swap(List,int,int);
   public static Object max(Collection);
   public static Object min(Collection);
   public static int binarySearch(List,Object);
   public static void fill(List,Object);
   public static ArrayList list(Enumeration);
   public static Collection synchronizedCollection(Collection);
   public static Collection unmodifiableCollection(Collection);
   public static void sort(List,Comparator);
   public static Enumeration enumeration(Collection);
   public static List singletonList(Object);
```

```
    public static List synchronizedList(List);
    public static List unmodifiableList(List);
    public static Map synchronizedMap(Map);
    public static Map unmodifiableMap(Map);
    public static Set singleton(Object);
    public static Set synchronizedSet(Set);
    public static Set unmodifiableSet(Set);
    public static boolean replaceAll(List,Object,Object);
    public static Object max(Collection,Comparator);
    public static Object min(Collection,Comparator);
}
interface Comparator{
    public abstract boolean equals(Object);
    public abstract int compare(Object,Object);
}
public class Date extends Object implements Serializable,Cloneable,Comparable{
    public Date();
    public int getDate();
    public int getDay();
    public int getHours();
    public int getMinutes();
    public int getMonth();
    public int getSeconds();
    public int getYear();
    public long getTime();
    public void setDate(int);
    public void setHours(int);
    public void setMinutes(int);
    public void setMonth(int);
    public void setSeconds(int);
    public void setYear(int);
    public int compareTo(Object);
    public boolean equals(Object);
    public int compareTo(Date);
    public boolean after(Date);
    public boolean before(Date);
}
public class StringTokenizer extends Object implements Enumeration{
    public StringTokenizer(String,String);
    public StringTokenizer(String,String,boolean);
    public int countTokens();
    public boolean hasMoreElements();
```

```
   public boolean hasMoreTokens();
   public Object nextElement();
   public String nextToken();
   public StringTokenizer(String);
   public String nextToken(String);
}
public final class Locale extends Object implements Cloneable,Serializable{
   public Locale(String);
   public Locale(String,String);
   public Locale(String,String,String);
   public static Locale getDefault();
   public boolean equals(Object);
   public String getCountry();
   public final String getDisplayCountry();
   public String getLanguage();
   public final String getDisplayLanguage();
   public final String getDisplayName();
   public String getVariant();
   public final String getDisplayVariant();
   public final String toString();
   public static String[] getISOCountries();
   public static String[] getISOLanguages();
}
public class Timer extends Object{
        public Timer();
        public void cancel();
        public void schedule(TimerTask task, Date time);
        public void schedule(TimerTask task, Date firstTime ,long period);
        public void schedule(TimerTask task, long delay);
        public void schedule(TimerTask ,long delay ,long period);
        public void scheduleAtFixedRate(TimerTask task,long delay ,long period);
        public void scheduleAtFixedRate(TimerTask, Date firstTime ,long period);
}
public abstract class TimerTask extends Object implements Runnable{
protected TimerTask()
        boolean cancel()
        public void run();
}
public abstract class Calendar extends Object implements Serializable, Cloneable{
   public static final int YEAR;
   public static final int MONTH;
   public static final int WEEK_OF_YEAR;
```

```
public static final int WEEK_OF_MONTH;
public static final int DATE;
public static final int DAY_OF_MONTH;
public static final int DAY_OF_YEAR;
public static final int DAY_OF_WEEK;
public static final int DAY_OF_WEEK_IN_MONTH;
public static final int AM_PM;
public static final int HOUR;
public static final int HOUR_OF_DAY;
public static final int MINUTE;
public static final int SECOND;
public static final int MILLISECOND;
public static final int SUNDAY;
public static final int MONDAY;
public static final int TUESDAY;
public static final int WEDNESDAY;
public static final int THURSDAY;
public static final int FRIDAY;
public static final int SATURDAY;
public static final int JANUARY;
public static final int FEBRUARY;
public static final int MARCH;
public static final int APRIL;
public static final int MAY;
public static final int JUNE;
public static final int JULY;
public static final int AUGUST;
public static final int SEPTEMBER;
public static final int OCTOBER;
public static final int NOVEMBER;
public static final int DECEMBER;
public static final int AM;
public static final int PM;
public int getFirstDayOfWeek();
public int hashCode();
public long getTimeInMillis();
public final void clear();
public boolean isLenient();
public int get(int field);
public void setFirstDayOfWeek(int);
public void set(int field, int value);
public final void set(int year, int month, int date);
```

```
    public final void set(int year, int month, int date, int hour, int minute);
    public final void set(int year, int month, int date, int hour, int minute, int second);
    public void setTimeInMillis(long);
    public Object clone();
    public boolean after(Object);
    public boolean before(Object);
    public boolean equals(Object);
    public String toString();
    public static Calendar getInstance();
    public final Date getTime();
    public final void setTime(Date);
    public TimeZone getTimeZone();
    public void setTimeZone(TimeZone);
    public static Calendar getInstance(Locale);

}

public class Arrays extends Object{
    public static void sort(xxx[]);
    public static int binarySearch(xxx[],xxx);
    public static void fill(xxx[],xxx);
    public static boolean equals(xxx[]);
    public static void sort(Object[],Comparator);
    public static List asList(Object[]);
    public static int binarySearch(Object[],Object,Comparator);
}
```