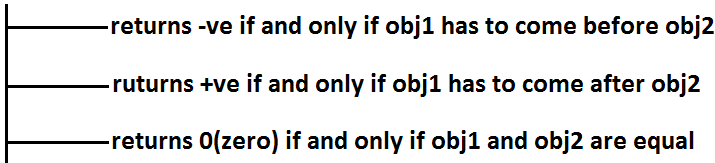
**Comparable interface:**

Comparable interface present in java.lang package and contains only one method compareTo() method.   
public int compareTo(Object obj);   
Example:   
**obj1.compareTo(obj2);**  
Diagram:  
  


class Test

{

public static void main(String[] args)

{

System.out.println("A".compareTo("Z"));

System.out.println("Z".compareTo("K"));

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

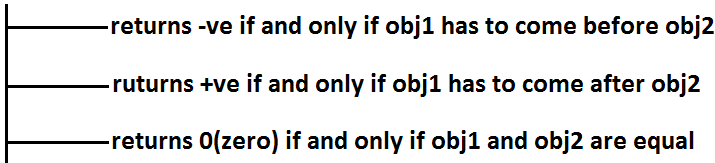
}

}

### Comparator interface:

Comparator interface present in java.util package this interface defines the following 2 methods.

#### 1) public int compare(Object obj1,Object Obj2);

Diagram:  
  


#### 2) public boolean equals(Object obj);

* Whenever we are implementing Comparator interface we have to provide implementation only for compare() method.
* Implementing equals() method is optional because it is already available from Object class through inheritance.

Requirement: **Write a program to insert integer objects into the TreeSet where the sorting order is descending order.**

import java.util.\*;

class Test

{

public static void main(String[] args)

{

TreeSet t=new TreeSet(new MyComparator()); //---->(1)

t.add(10);

t.add(0);

t.add(15);

t.add(5);

t.add(20);

System.out.println(t);//[20, 15, 10, 5, 0]

}

}

class MyComparator implements Comparator

{

public int compare(Object obj1,Object obj2)

{

Integer i1=(Integer)obj1;

Integer i2=(Integer)obj2;

if(i1<i2)

return +1;

else if(i1 > i2)

return -100;

else return 0;

}

}

import java.util.\*;

class Employee implements Comparable

{

String name;

int eid;

Employee(String name,int eid)

{

this.name=name;

this.eid=eid;

}

public String toString()

{

return name+"----"+eid;

}

public int compareTo(Object o)

{

int eid1=this.eid;

int eid2=((Employee)o).eid;

if(eid1 < eid2)

{

return -1;

}

else if(eid1 > eid2)

{

return 1;

}

else return 0;

}

}

class CompComp

{

public static void main(String[] args)

{

Employee e1=new Employee("raju",100);

Employee e2=new Employee("ravi",200);

Employee e3=new Employee("sai",50);

Employee e4=new Employee("pavan",150);

Employee e5=new Employee("raju",100);

TreeSet t1=new TreeSet();

t1.add(e1);

t1.add(e2);

t1.add(e3);

t1.add(e4);

t1.add(e5);

System.out.println(t1);

TreeSet t2=new TreeSet(new MyComparator());

t2.add(e1);

t2.add(e2);

t2.add(e3);

t2.add(e4);

t2.add(e5);

System.out.println(t2);

}

}

class MyComparator implements Comparator

{

public int compare(Object obj1,Object obj2)

{

Employee e1=(Employee)obj1;

Employee e2=(Employee)obj2;

String s1=e1.name;

String s2=e2.name;

return s1.compareTo(s2);

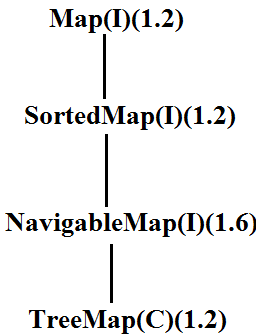
}

}

**Compression of Comparable and Comparator ?**

|  |  |
| --- | --- |
| **Comparable** | **Comparator** |
| 1) Comparable meant for default natural sorting order. | 1) Comparator meant for customized sorting order. |
| 2) Present in java.lang package. | 2) Present in java.util package. |
| 3) Contains only one method.  compareTo() method. | 3) Contains 2 methods. Compare() method. Equals() method. |
| 4) String class and all wrapper Classes implements Comparable interface. | 4) used in GUI) |

## NavigableMap:

It is the child interface of SortedMap and it defines several methods for navigation purpose.   
Diagram:  
  
  
  
  
**NavigableMap interface defines the following methods.**

1. ceilingKey(e);
2. higherKey(e);
3. floorKey(e);
4. lowerKey(e);
5. pollFirstEntry();
6. pollLastEntry();
7. descendingMap();

Example:

import java.util.\*;

class NavigableMapDemo

{

public static void main(String[] args)

{

TreeMap<String,String> t=new TreeMap<String,String>();

t.put("b","banana");

t.put("c","cat");

t.put("a","apple");

t.put("d","dog");

t.put("g","gun");

System.out.println(t);//{a=apple, b=banana, c=cat, d=dog, g=gun}

System.out.println(t.ceilingKey("c"));//c

System.out.println(t.higherKey("e"));//g

System.out.println(t.floorKey("e"));//d

System.out.println(t.lowerKey("e"));//d

System.out.println(t.pollFirstEntry());//a=apple

System.out.println(t.pollLastEntry());//g=gun

System.out.println(t.descendingMap());//{d=dog, c=cat, b=banana}

System.out.println(t);//{b=banana, c=cat, d=dog}

}

}

import java.io.\*;

class Account implements Serializable

{

String userName="Bhaskar";

transient String pwd="kajal";

}

class CustomizedSerializeDemo

{

public static void main(String[] args)throws Exception{

Account a1=new Account();

System.out.println(a1.userName+"........."+a1.pwd);

FileOutputStream fos=new FileOutputStream("abc.ser");

ObjectOutputStream oos=new ObjectOutputStream(fos);

oos.writeObject(a1);

FileInputStream fis=new FileInputStream("abc.ser");

ObjectInputStream ois=new ObjectInputStream(fis);

Account a2=(Account)ois.readObject();

System.out.println(a2.userName+"........."+a2.pwd);

}

}