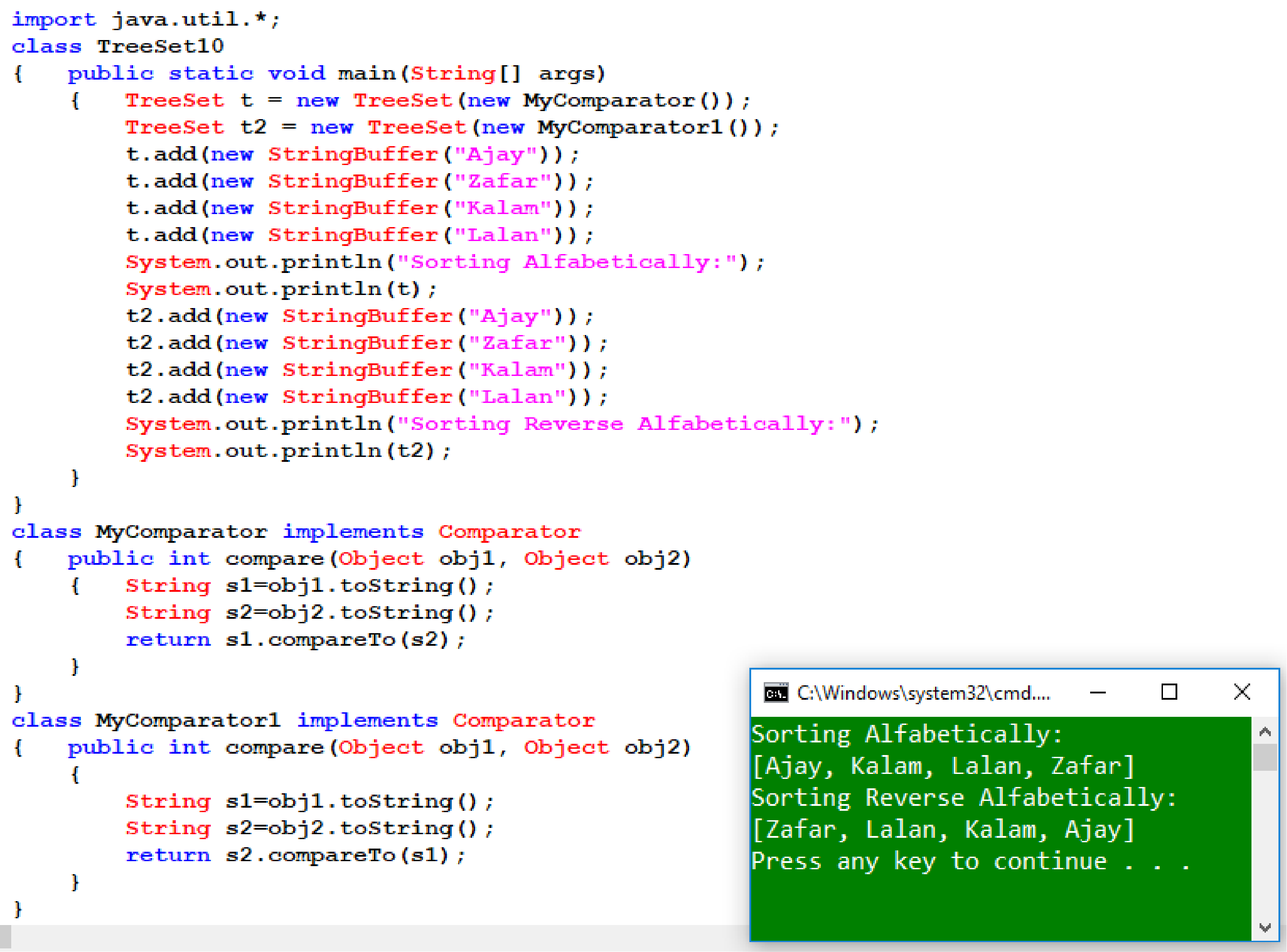
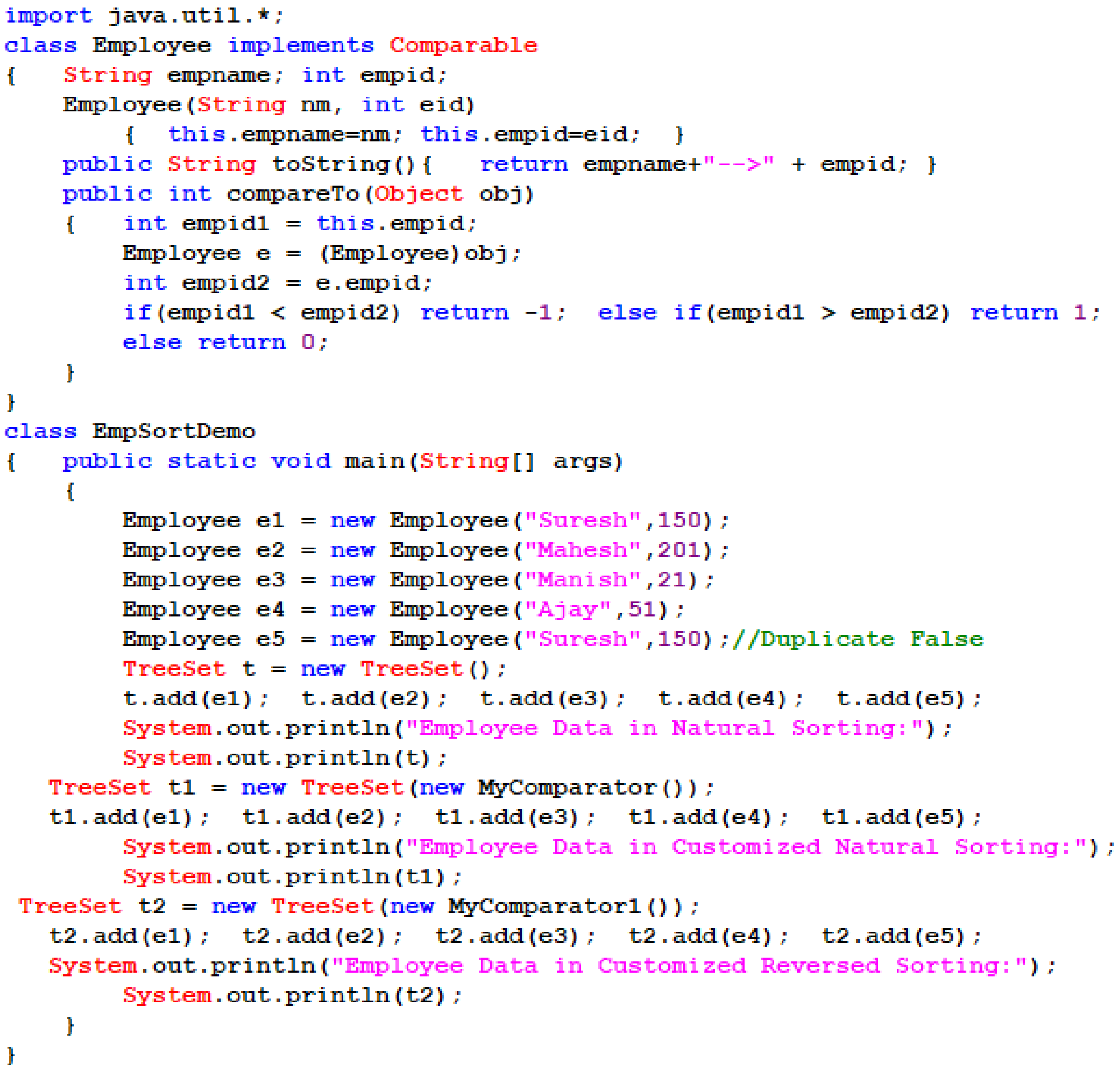
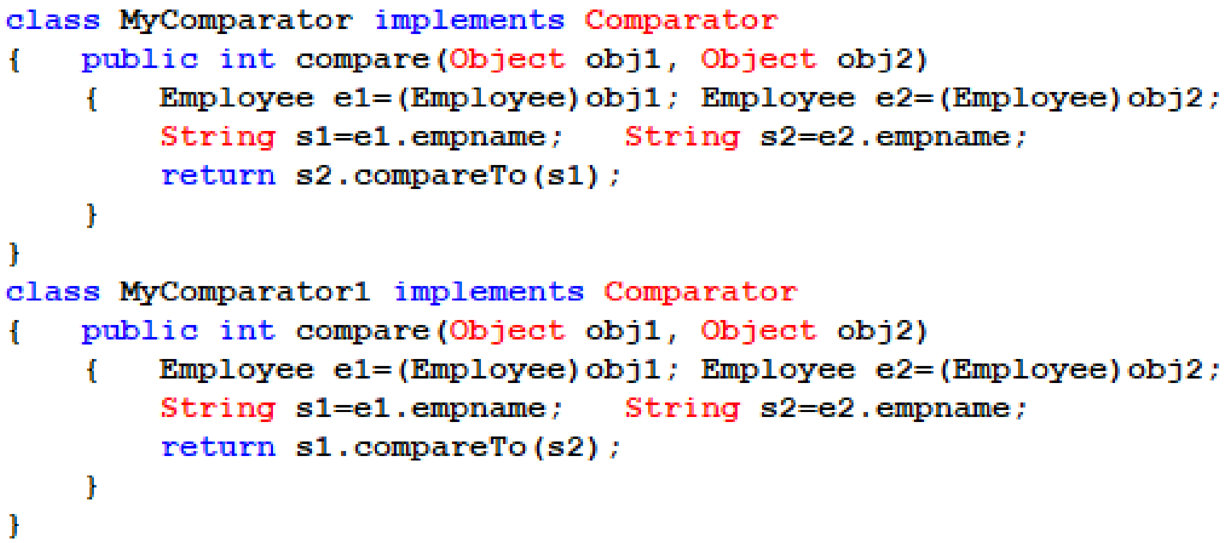


```java
import java.util.*;
class TreeSet10
{    public static void main(String[] args)
     {    TreeSet t = new TreeSet(new MyComparator());
          TreeSet t2 = new TreeSet(new MyComparator1());
          t.add(new StringBuffer("Ajay"));
          t.add(new StringBuffer("Zafar"));
          t.add(new StringBuffer("Kalam"));
          t.add(new StringBuffer("Lalan"));
          System.out.println("Sorting Alfabetically:");
          System.out.println(t);
          t2.add(new StringBuffer("Ajay"));
          t2.add(new StringBuffer("Zafar"));
          t2.add(new StringBuffer("Kalam"));
          t2.add(new StringBuffer("Lalan"));
          System.out.println("Sorting Reverse Alfabetically:");
          System.out.println(t2);
     }
}
class MyComparator implements Comparator
{    public int compare(Object obj1, Object obj2)
     {    String s1=obj1.toString();
          String s2=obj2.toString();
          return s1.compareTo(s2);
     }
}
class MyComparator1 implements Comparator
{    public int compare(Object obj1, Object obj2)
     {
          String s1=obj1.toString();
          String s2=obj2.toString();
          return s2.compareTo(s1);
     }
}
```

```
Sorting Alfabetically:
[Ajay, Kalam, Lalan, Zafar]
Sorting Reverse Alfabetically:
[Zafar, Lalan, Kalam, Ajay]
Press any key to continue . . .
```

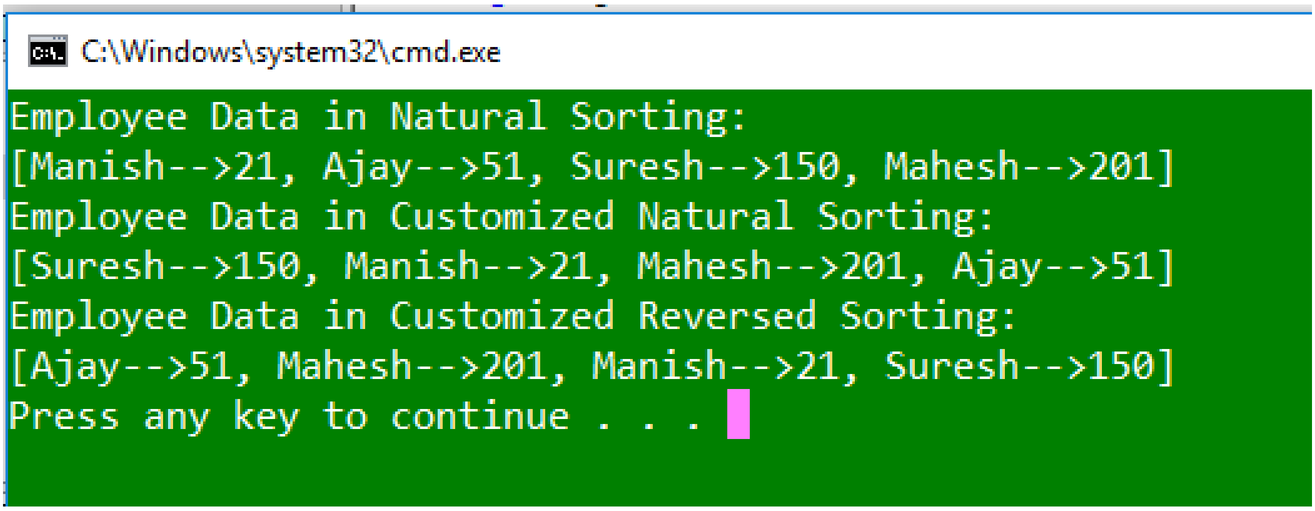


```
import java.util.*;
class Employee implements Comparable
{    String empname; int empid;
     Employee(String nm, int eid)
         {  this.empname=nm; this.empid=eid;  }
     public String toString(){    return empname+"-->" + empid; }
     public int compareTo(Object obj)
     {    int empid1 = this.empid;
          Employee e = (Employee)obj;
          int empid2 = e.empid;
          if(empid1 < empid2) return -1;  else if(empid1 > empid2) return 1;
          else return 0;
     }
}
class EmpSortDemo
{    public static void main(String[] args)
     {
          Employee e1 = new Employee("Suresh",150);
          Employee e2 = new Employee("Mahesh",201);
          Employee e3 = new Employee("Manish",21);
          Employee e4 = new Employee("Ajay",51);
          Employee e5 = new Employee("Suresh",150);//Duplicate False
          TreeSet t = new TreeSet();
          t.add(e1);  t.add(e2);  t.add(e3);  t.add(e4);  t.add(e5);
          System.out.println("Employee Data in Natural Sorting:");
          System.out.println(t);
   TreeSet t1 = new TreeSet(new MyComparator());
   t1.add(e1);  t1.add(e2);  t1.add(e3);  t1.add(e4);  t1.add(e5);
          System.out.println("Employee Data in Customized Natural Sorting:");
          System.out.println(t1);
 TreeSet t2 = new TreeSet(new MyComparator1());
   t2.add(e1);  t2.add(e2);  t2.add(e3);  t2.add(e4);  t2.add(e5);
   System.out.println("Employee Data in Customized Reversed Sorting:");
          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.empname;   String s2=e2.empname;
          return s2.compareTo(s1);
     }
}
class MyComparator1 implements Comparator
{    public int compare(Object obj1, Object obj2)
     {    Employee e1=(Employee)obj1; Employee e2=(Employee)obj2;
          String s1=e1.empname;   String s2=e2.empname;
          return s1.compareTo(s2);
     }
}
```

Output:-



```
Employee Data in Natural Sorting:
[Manish-->21, Ajay-->51, Suresh-->150, Mahesh-->201]
Employee Data in Customized Natural Sorting:
[Suresh-->150, Manish-->21, Mahesh-->201, Ajay-->51]
Employee Data in Customized Reversed Sorting:
[Ajay-->51, Mahesh-->201, Manish-->21, Suresh-->150]
Press any key to continue . . .
```