

```

public class TreeSetExample {
    public static void main(String[] args) {
        //TreeSet 컬렉션 생성
        TreeSet<Integer> scores = new TreeSet<>();

        //Integer 객체 저장
        scores.add(87);
        scores.add(98);
        scores.add(75);
        scores.add(95);
        scores.add(80);

        //정렬된 Integer 객체를 하나씩 가져오기
        for(Integer s : scores) {
            System.out.print(s + " ");
        }
        System.out.println("\n");
    }
}

```

#### 실행 결과

75 80 87 95 98

가장 낮은 점수: 75

가장 높은 점수: 98

95점 아래 점수: 87

95점 위의 점수: 98

95점이거나 바로 아래 점수: 95

85점이거나 바로 위의 점수: 87

98 95 87 80 75

80 87 95 98

80 87

```

//특정 Integer 객체를 가져오기
System.out.println("가장 낮은 점수: " + scores.first());
System.out.println("가장 높은 점수: " + scores.last());
System.out.println("95점 아래 점수: " + scores.lower(95));
System.out.println("95점 위의 점수: " + scores.higher(95));
System.out.println("95점이거나 바로 아래 점수: " + scores.floor(95));
System.out.println("85점이거나 바로 위의 점수: " + scores.ceiling(85) + "\n");

//내림차순으로 정렬하기
NavigableSet<Integer> descendingScores = scores.descendingSet();
for(Integer s : descendingScores) {
    System.out.print(s + " ");
}
System.out.println("\n");

//범위 검색( 80 <= )
NavigableSet<Integer> rangeSet = scores.tailSet(80, true);
for(Integer s : rangeSet) {
    System.out.print(s + " ");
}
System.out.println("\n");

//범위 검색( 80 <= score < 90 )
rangeSet = scores.subSet(80, true, 90, false);
for(Integer s : rangeSet) {
    System.out.print(s + " ");
}
}
}

```