

BoardDao 객체의 getBoardList() 메소드를 호출하면 List<Board> 타입의 컬렉션을 리턴합니다. ListExample 클래스의 실행 결과를 보고, BoardDao 클래스와 getBoardList() 메소드를 작성해보세요.

```
public class Board {
    private String title;
    private String content;

    public Board(String title, String content) {
        this.title = title;
        this.content = content;
    }

    public String getTitle() { return title; }
    public String getContent() { return content; }
}
```

```
import java.util.List;

public class ListExample {
    public static void main(String[] args) {
        BoardDao dao = new BoardDao();
        List<Board> list = dao.getBoardList();
        for(Board board : list) {
            System.out.println(board.getTitle() + "-" + board.getContent());
        }
    }
}
```

실행 결과

제목1-내용1

제목2-내용2

제목3-내용3

HashSet에 Student 객체를 저장하려고 합니다. 학번이 같으면 동일한 Student라고 가정하고 중복 저장이 되지 않도록 하고 싶습니다. Student 객체의 해시코드는 학번이라고 가정하고 Student 클래스를 작성해보세요.

```
public class Student {
    public int studentNum;
    public String name;

    public Student (int studentNum, String name) {
        this.studentNum = studentNum;
        this.name = name;
    }

    //여기에 코드를 작성하세요.
}
```

```
import java.util.HashSet;
import java.util.Set;

public class HashSetExample {
    public static void main(String[] args) {
        Set<Student> set = new HashSet<Student>();

        set.add(new Student(1, "홍길동"));
        set.add(new Student(2, "신용권"));
        set.add(new Student(1, "조민우"));

        System.out.println("저장된 객체 수: " + set.size());
        for(Student s : set) {
            System.out.println(s.studentNum + ":" + s.name);
        }
    }
}
```

실행 결과

저장된 객체 수: 2
1:홍길동
2:신용권

HashMap에 아이디(String)와 점수(Integer)가 저장되어 있습니다. 실행 결과와 같이 평균 점수, 최고 점수, 최고 점수를 받은 아이디를 출력하도록 코드를 작성해보세요.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class MapExample {
    public static void main(String[] args) {
        Map<String,Integer> map = new HashMap<String,Integer>();
        map.put("blue", 96);
        map.put("hong", 86);
        map.put("white", 92);

        String name = null;    //최고 점수를 받은 아이디를 저장하는 변수
        int maxScore = 0;      //최고 점수를 저장하는 변수
        int totalScore = 0;    //점수 합계를 저장하는 변수

        //여기에 코드를 작성하세요.
    }
}
```

실행 결과

평균 점수: 91
최고 점수: 96
최고 점수를 받은 아이디: blue

. TreeSet에 Student 객체를 저장할 때, Student의 score 필드값을 기준으로 자동 정렬되도록 구현하고 싶습니다. TreeSet의 last() 메소드를 호출했을 때 가장 높은 score의 Student 객체가 리턴되도록 Student 클래스에서 밑줄 친 곳과 비어있는 곳에 코드를 작성해보세요.

```
public class Student _____ {  
    public String id;  
    public int score;  
  
    public Student (String id, int score) {  
        this.id = id;  
        this.score = score;  
    }  
    _____  
}
```

```
import java.util.TreeSet;  
  
public class TreeSetExample {  
    public static void main(String[] args) {  
        TreeSet<Student> treeSet = new TreeSet<Student>();  
        treeSet.add(new Student("blue", 96));  
        treeSet.add(new Student("hong", 86));  
        treeSet.add(new Student("white", 92));  
  
        Student student = treeSet.last();  
        System.out.println("최고 점수: " + student.score);  
        System.out.println("최고 점수를 받은 아이디: " + student.id);  
    }  
}
```

실행 결과

```
최고 점수: 96  
최고 점수를 받은 아이디: blue
```