Assignment 2: Sorted Set

1. Requirement

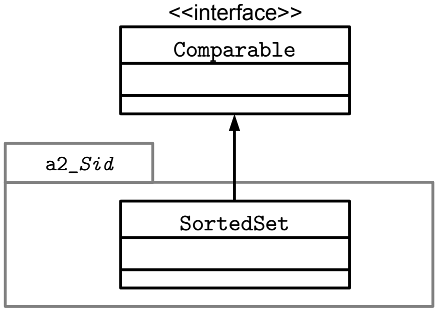

Figure 1: UML class diagram.

Program FSIS now needs to support a collection data type named `SortedSet`. This represents unbounded, sorted sets of `Comparable` objects; the objects are sorted in the ascending order based on the result of comparing them using the method `Comparable.compareTo`. Specifically, for all s: `SortedSet`, o_1, o_2: `Comparable`, y = `min`(o_1, o_2), x = `max`(o_1, o_2): $o_1, o_2 \in s \leftrightarrow x \notin$ `subset`(s, `first`(s), y).

Here, `first`(s) returns the first (*i.e.* smallest) element of s; `subset`(s,l,u) returns a `SortedSet` containing a sub-set of the elements of s starting (in the sorting order) from an element l up to and including an element u.

Figure 1 shows a concept class diagram of `SortedSet`. The arrowed line means that `SortedSet` uses `Comparable`, which in this context means that it operates on `Comparable` objects. Table 1 lists the attribute design details of `SortedSet`. It is expected that `SortedSet` will contain a number of essential operations (which do *not* need to include the two operations `subset` and `first` explained above).

Please note that your program will be marked automatically by a program, which expects you to strictly follow all the relevant the design rules. An error in one part may affect other parts.

Table 1: The attribute design details of `SortedSet`.

Attribute(s)	**formal type**	**mutable**	**optional**	**min**	**max**	**length**
`elements`	`SORTEDSET<Comparable>`	T	F	-	-	-

Usage example

Below is a code example that uses class `Customer` (which is a `Comparable`) to help you understand how your `SortedSet` is used in a program:

```
Comparable c1 = new Customer(1, "Tran Mai Linh", "0249871234", "Hoan Kiem, Hanoi");
Comparable c2 = new Customer(2, "Nguyen Van Anh", "0243561789", "Dong Da, Hanoi");
SortedSet s = new SortedSet();
s.insert(c1);
s.insert(c2);
// this should display c2 before c1
System.out.println("A sorted Customer set: \n" + s);
```

2. Tasks

1. Create a package named `a2_`*`Sid`* as shown in Figure 1, where *`Sid`* is your student id. For example, if your student id is 123456789 then the package name is `a2_123456789`.

   You will need to use this package to hold all the Java classes that you create for the program.

   IMPORTANT: Failure to name the package as described above will result in an invalid program.

2. Specify and implement class `SortedSet` with all the ***essential*** attribute(s) and operations.

   Note that:

   - the operations must be annotated appropriately
   - there must be an iteration abstraction for iterating through its elements
   - the elements are sorted as they are inserted into the set.

   IMPORTANT: In order for `SortedSet` to be complete, all the essential operations must work together as a whole. An error in one operation may affect the validity of other operations. For example, an error in the add operation can cause errors in the remove and observer operations.

3. Submission

Create a **zip-compressed** file containing **just the folder of the package** specified in Task 1. You must name the file as follows: **a2_*Sid*.zip**, where ***Sid*** is your student id.

Submit your file to the designated submission box for this assignment.

IMPORTANT: Failure to prepare the file as described above will result in an invalid program. In particular, ONLY the **ZIP** format is accepted. Other formats (e.g. RAR) are NOT accepted.