Exercise 4 - Containers & Generics

- *Modify the DiscreteAttribute class as follows:*
  - *modify the values member declaration to use a generics container of type TreeSet<String>.*
  - *the class must now implement* interface generics *Iterable<String> and then provide the implementation for the public Iterator<T> iterator() method*
  - *elimination of String getValue(int i) method*

- *Change the Date class as follows:*
  - *define the inner class Example (inner in Data, visibility friendly) to model each transaction. The class implements the generics interface Comparable<Example> e include i following members* **(decide the visibility of each Example member appropriately):**
    - *List<Object> example=new ArrayList<Object>(); // array of Objects representing the individual transaction (or row of a table)*
    - *void add(Object o) // adds o to the queue of example*
    - *Object get(int i) // returns the i-th reference collected in example*
    - *int compareTo(Example ex) // returns 0, -1, 1 based on the result of the comparison. 0 if the two examples include the same values. Otherwise the result of the compareTo(...) invoked on the first disagreement value pair.*
    - *public String toString()// restores a string representing the state of example (make use of for-each)*

- *modify the attributeSet member declaration to use a generics container of type List<Attribute> :*

List<Attribute> attributeSet = new LinkedList<Attribute>();

- *modify the date member declaration to use a List<Example> generics container:*

  List<Example> data =new ArrayList<Example>();

- Modify the constructor of Date so as to populate date without duplicate examples. *To do this, use a TreeSet as shown below*

```
public Data(){
        //date
        TreeSet<Example> tempData = new TreeSet<Example>();
        Example ex0=new Example();
        Example ex1=new Example();
        ... // COMPLETE
        ex0.add(new String ('sunny'));
        ex1.add(new String ('sunny'));
        ... // COMPLETE
        ex0.add(new String ('hot'));
        ex1.add(new String ('hot'));
        ... // COMPLETE
        ex0.add(new String ('high'));
        ex1.add(new String ('high'));
        ... // COMPLETE
        ex0.add(new String ('weak'));
        ex1.add(new String ('strong'));
        ... // COMPLETE
```

```
        ex0.add(new String ('no'));
        ex1.add(new String ('no'));
        ... // COMPLETE
        tempData.add(ex0);
        tempData.add(ex1);
        ... // COMPLETE
        data=new ArrayList<Example>(tempData);
        ... // COMPLETE
        // initialise numberOfExamples
        ... // COMPLETE
        //initialise explanatory Set
    }
```

- *remove the countDistinctTuples() method: this method is no longer necessary since now date definitely does not contain duplicate transactions*

Modify the methods using members and methods introduced so far accordingly.

- *Remove the ArraySet class from the project and replace it with the HashSet container. Then modify the declaration of clusteredData as follows*

Set<Integer> clusteredData=new HashSet<Integer>().

And modify where and if necessary the classes in the project that use clusteredData.