

ATLANTIC AIR ROUTES

The **sorted multimap** abstract data type is a container of key / value mappings with a not necessarily unique key of a parametric type limited to Comparable. Is defined by the interfaces:

```
public interface SortedMultimap<K extends Comparable,V> extends Container
{
    /**
     * Returns an array containing all the values mapped to the specified key in
     * this multimap if this multimap contains mappings for the key, otherwise
     * an empty array
     * @param key the specified key to return mapped values for
     * @return an array containing all the values mapped to the specified key in
     *         this multimap if this multimap contains mappings for the key,
     *         otherwise an empty array
     */
    Object[] findAll(K key);

    /**
     * Maps the specified value to the specified key in this multimap
     * @param key the specified key
     * @param value the specified value
     * @throws java.lang.IllegalArgumentException if the specified key or value
     *         is null
     */
    void insert(K key, V value);

    /**
     * Returns the keys in this multimap
     * @return a sorted array containing the keys of this map if this multimap
     *         is not empty, otherwise an empty array
     */
    Comparable[] keys();

    /**
     * Removes from this multimap all the mappings of the specified key and
     * returns the mapped values in an array
     * @param key the specified key
     * @return an array containing the values mapped to the specified key,
     *         if any, otherwise an empty array
     */
    Object[] removeAll(K key);
} // end of the SortedMultimap interface

public interface Container
{
    /**
     * Checks if this container is empty
     * @return true if this container is empty, otherwise false
     */
    boolean isEmpty();

    /**
     * Makes the container empty
     */
    void makeEmpty();

    /**
     * Provides the number of data items in this container
     * @return the number of data items in this container
     */
    int size();
} // end of Container interface
```

Write the Java code of the sorted multimap below:

```
public class D<K extends Comparable, V> implements SortedMultimap<K, V>
```

whose constructor initializes an empty multimap.

Next, write the private section of the extended sorted multimap below:

```
public class ESD<K extends Comparable, V extends Comparable> extends D<K, V>  
{ // private section
```

```
    ...
```

```
    public ESD() {...} // constructor
```

```
    /** @return true if this multimap contains a mapping for the specified key  
    */ public boolean contains(K key){...}
```

```
    /** @return a sorted array containing the keys without duplicates of this  
        multimap or an empty array if this multimap is empty  
    */ public Comparable[] keySet() {...}
```

```
    /**@return a sorted array containing the values without duplicates in this  
        multimap or an empty array if this multimap is empty  
    */ public Comparable[] valueSet() {...}  
}
```

in which the constructor initializes an empty extended sorted multimap.

During the code correction, the code will be tested with the commands below:

```
$rm *.class
```

```
$javac ESDTester.java
```

```
$java ESDTester flights.txt
```

where flights.txt is the attached file and the ESDTester test class is to be written (see instructions in the exercise called classroom programming 10).

The attached OK.txt text file shows the printout on standard output when the ESDTester class runs and reads data from the flights text file.

Generally, the code that reports errors at compile time is considered insufficient.

During the correction of code, the following maximum scores will be assigned to the methods:

D: findAll 6, keys 6, insert 6, removeAll 6 (Tot = 24)

ESD: contains 1, keySet 2, valueSet 2 (Tot = 5)

A score can be assigned for the programming style and up to two scores for the time complexity of the findAll method.

In writing the D and ESD classes, it is not allowed:

- to add non-private variables and/or methods;
- to use classes from the Java Platform API except for those of the java.lang package

At the end of the programming test, the student will leave all code files in the working directory. The files submitted by the student for evaluation need to contain the first comment including first and family names, roll number, date, number of work station.

FOUNDATION OF COMPUTER SCIENCE

Name _____ Roll no.. _____ Station ADT _____

I submit for evaluation the following files: ☐ D.java ☐ ESD.java ☐

Student's Signature _____

I do not submit files for evaluation and I retire from the exam

Student's Signature _____