

CIS 2168: Assignment #2

Due on Tuesday, September 9, 2014

11:59pm

Anwar Mamat

Problem 1

Fixed Size Bag LoopBag (100%) Do you know how the Flight Recorders work? They always store the last 2 hours of sound. They use a continuous loop of tape that completes a cycle every 2 hours. As new material is recorded, the oldest material is replaced. We talked about the resizable bag class. In this assignment you will implement a variant of the Bag class: LoopBag. LoopBag is a fixed size bag and works like the flight recorder. The size of the bag is given the when the LoopBag object is created. After the LoopBag is full, if a new item is inserted, the oldest item in the LoopBag is replaced. You will implement the following interface:

```
import java.util.Iterator;
public interface LoopBag<E> extends Iterable {
    /**
     * Adds the given item to bad LoopBag.
     * @param item the item to add
     */
    void insert(E item);

    /**
     * Returns the number of items in this LoopBag.
     * @return the number of items in this LoopBag
     */
    int size();

    /**
     * Returns true if this LoopBag is empty and false otherwise.
     * @return true if this LoopBag is empty; false otherwise.
     */
    boolean isEmpty();

    /**
     * if the bag contains a given item?
     * @return true if bag contains the item. false otherwise
     */
    boolean contains(E item);

    /**
     * creates the union with the given LoopBag
     */
    void union(LoopBag lb);

    /**
     * Returns an iterator for this LoopBag. Iterator iterates from oldest to newest.
     * @return an iterator for this LoopBag
     */
    Iterator<E> iterator();
}
```

Listing 1: Test Example

```
LoopBag<Integer> bag = new LoopBag(5);
bag.insert(1);
bag.insert(2);
bag.insert(3);
```

```
5 bag.insert(4);
  bag.insert(5);
  bag.insert(6);
  bag.insert(7);
  for (Integer i: bag) {
10     System.out.print(i+", ");
  }

Output:
3, 4, 5, 6, 7
```

1. You are not allowed to use Java ArrayList. You have to implement the LoopBag using basic arrays.
2. Iterator iterates from the oldest to the newest.

What will you learn from this assignment

1. Implementing collections
2. interface
3. iterator
4. comparing objects

Grading

Homework is 100 points. 80 will reflect functionality and correctness. 20 points on your program will reflect your programming style, documentation. If your code does not compile, you will not receive any credit.

Commenting and Documenting Code

Code must be properly commented. The main idea is that the grader should be able to understand your code easily, not have to tear his or her hair out wondering what some statement is doing. The first time you have to deal with poorly commented code (if you haven't already), you will understand how annoying it is. In particular, the top of each code file should contain your name, the course and assignment numbers, and a brief summary of what's in the file. Line-by-line comments should be included as necessary to make the code easy to read. A clear coding style, together with informative variable and function names, will reduce the number of comments required. Obscure code or cryptic function names will cause loss of points (for bad style) and also require more extensive comments.

What to submit

A single zip file called Assignment2_firstname_lastname.zip, where firstname is your first name, and lastname is your last name. In this zip file, put:

1. Java source
2. A README file with:

- Instructions to compile and run of your code (include a description of command line options).
 - If your solution is not perfect, explain what parts you did and what part you did not do.
 - List of files submitted
 - All your data and results (in plain text files).
 - Anything else you want TA know
3. Submit this zip file to Blackboard