

# CS367 Announcements

## Thursday, June 20, 2013

- H1, P1 handin dirs created
- H1 due Monday, June 24th 6pm

### Last Time

- ListADT review
- implementing the List ADT using arrays
- Lists and the Java API
- implementing the List ADT contains method
- iterators
  - what are they
  - using

### Today

- Iterators
  - review
  - implementing
- ArrayBagIterator
- using the command line

## Iterator Review - Breadcrumbs

```
List<String> myList = new ArrayList<String>();  
  
... // populate list  
  
for( String item : myList) {  
    System.out.println(item);  
}
```

---

```
public class ArrayList<E> implements ListADT<E> {  
    ... // other required methods  
  
    public          {  
  
    }  
}
```

---

```
public interface ListADT<E> extends          {  
    ... // other unimplemented methods  
  
    ;  
}
```

---

```
public class          implements          {  
    ...  
    public          { ... }  
    public          { ... }  
    public          { ... }  
}
```

## Making the ArrayBox Iterable

```
import java.util.*;
public class ArrayBox<E> implements BoxADT<E> {

    // *** Data members (fields) ***
    private E[] items;
    private int numItems;
    private static final int INIT_SIZE = 100;

    // *** Constructor ***
    public ArrayBox() {
        items = (E[])(new Object[INIT_SIZE]);
        numItems = 0;
    }

    // *** Box ADT methods ***
    public void add(E item) {
        if (item == null) {
            throw new NullPointerException("Box does not allow null elements");
        }
        if (numItems == items.length) { expandArray(); }
        items[numItems] = item;
        numItems++;
    }

    public E remove() throws NoSuchElementException {
        if (numItems < 1) throw new NoSuchElementException();
        numItems--;
        return items[numItems];
    }

    public boolean isEmpty() { return numItems == 0; }

    public Iterator<E> iterator() {

    }

    // *** Private methods ***
    private void expandArray() { ... }
}
}
```

## Implementing an Iterator Option 1 (done like SimpleArrayList in readings)

```
import java.util.*;
public class ArrayBoxIterator<E> implements Iterator<E> {

    public ArrayBoxIterator(                ) {

    }

    public boolean hasNext() {

    }

    public E next() {

    }

    public void remove() {

    }
}
```

## Options for Coding Iterator Classes

Option 1

Option 2

## Coding a Direct-Access Iterator Class

### BoxADT interface

```
import java.util.*;
public interface BoxADT<E> {
    void add(E item);
    E remove() throws NoSuchElementException;
    boolean isEmpty();
    Iterator<E> iterator();
}
```

### ArrayBox class

```
import java.util.*;
public class ArrayBox<E> implements BoxADT<E> {

    // *** Data members (fields) ***
    private E[] items;
    private int numItems;
    private static final int INIT_SIZE = 100;

    ...

    public Iterator<E> iterator() {

    }

    ...

}
```

## Coding a Direct-Access Iterator Class

```
import java.util.*;
public class ArrayBoxIterator<E> implements Iterator<E> {

    public ArrayBoxIterator(                ) {

    }

    public boolean hasNext() {

    }

    public E next() {

    }

    public void remove() {

    }
}
```

## Command Line Java Development

### Edit

```
mumble-16% ls  
bin src  
mumble-16% nano src/BoxIteratorTester.java
```

### Compile

```
mumble-16% javac -d bin -sourcepath src src/BoxIteratorTester.java
```

### Run

```
mumble-16% java -cp bin BoxIteratorTester
```



## Command Line Arguments

**Write a java program (making use of iterators) that takes several command line arguments, puts them in a box, and counts the number of times the first argument appears in the bag.**

```
mumble-16% java BoxIteratorTester a b r a c a d a b r a
mumble-16% java BoxIteratorTester red blue green red yellow orange
2
```

```
public class BagIteratorTester {
    public static void main(String[] args) {
```