CS203 Programming with Data Structures

Lists and Iterators

Chengyu Sun
California State University, Los Angeles

Data Structures

- Abstract Data Type (ADT)
- A collection of data and a set of operations that can be performed on the data
- *Abstract* – operations are defined, but how to implements these operations are not

Some Simple ADTs

- List
- Queue
- Stack
- Tree
- …
- Just look under `java.util` package

Why Do We Study Data Structures?

- Common to all languages, not just Java
- Building blocks for more complex algorithms
- Programming techniques
- Complexity analysis and performance tradeoffs

List

beef
beer
biscuits
broccoli
… …

- A *ordered* collection of *objects*

List Operations

- Insert
- Remove
- Get
- Search
- Clear
- Size
- Output

The `List` Interface

- http://sun.calstatela.edu/~cysun/www/teaching/cs203/extras/List.java

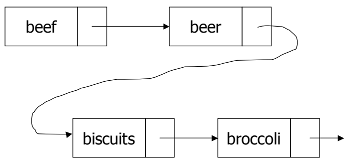
Array Implementation

```
public class ArrayList implements List {

    private Object elements[];
    …
}
```

Performance Concerns of ArrayList

- Which operations are efficient??
- Which are not??
- How much space are we wasting??
- What can we do about it??

Linked List



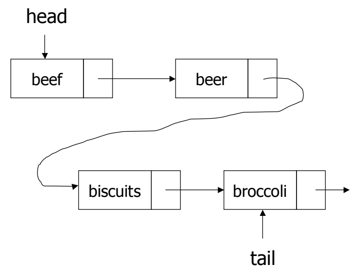
ListNode

```
class ListNode {

        Object          data;
        ListNode        next;

  …
}
```

Implementation Considerations

- How do we insert at the front of the list??
- How do we remove the first element of the list??
- How do we remove the last element of the list??

Head and Tail



Performance Concerns of LinkedList

```
// assume list is a linked list of Integers

int sum = 0
for( int i=0 ; i < list.size() ; ++i )
    sum += (Integer) list.get(i);
```

◈What wrong with this code??

Iterator

◈Iterates though each element of a collection …

◈… *without* exposing the internal of the collection class

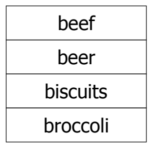
The `Iterator` Interface

```
public interface Iterator {

    public boolean hasNext();

    public Object next();

}
```

The `ListIterator` Interface

```
public interface ListIterator extends Iterator {

    public boolean hasPrevious();

    public Object previous();

}
```

ListIterator Operations

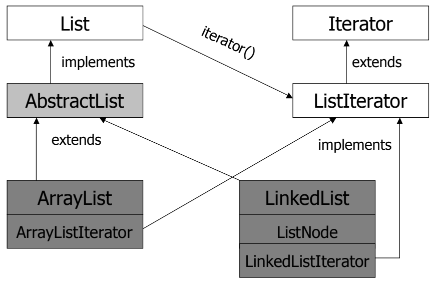


```
ListIterator it = list.iterator();

it.hasPrevious();     // false
it.hasNext();         // true
it.next();            // "beef"
it.next();            // "beer"
it.previous();        // "beer"
it.next();            // "beer"
```

LinkedListIterator

◈An iterator class for LinkedList

Refactoring List Classes



List
iterator()
Iterator
implements
extends
AbstractList
ListIterator
extends
implements
ArrayList
ArrayListIterator
LinkedList
ListNode
LinkedListIterator