Inheritance: Applications

Lecture 23
Sections 14.5 - 14.6

Robb T. Koether

Hampden-Sydney College

Wed, Mar 17, 2010

Outline

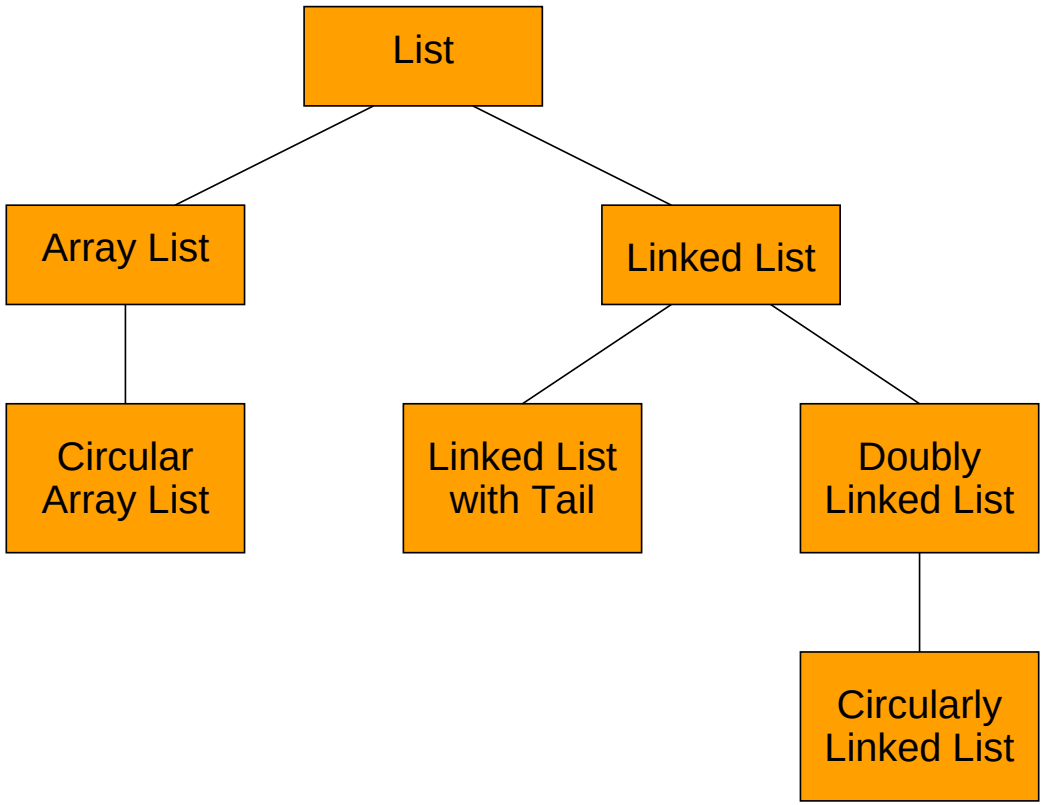
Outline

List Classes

- We have developed several different implementations of an abstract list.
  - `ArrayList`
  - `CircArrayList`
  - `LinkedList`
  - `LinkedListwTail`
  - `DoublyLinkedList`
  - `CircLinkedList`
- They fall into two basic categories.
  - Array-based.
  - Linked.

Hierarchy of Lists

- We can easily imagine a hierarchy of list classes.



The List Interface

```
class List
{
    List();
    List(int sz);
    List(int sz, const T& value);
    List(const List<T>& lst);
    ~List();
    T getElement(int pos) const;
    T& getElement(int pos);
    int size() const;
    bool isEmpty() const;
    void setElement(int pos, const T& value);
    void insert(int pos, const T& value);
    void remove(int pos);
```

The List Interface

```
    void pushFront(const T& value);
    void pushBack(const T& value);
    T popFront();
    T popBack();
    void makeEmpty();
    void input(istream& in);
    void output(ostream& out) const;
    List<T>& operator=(const List<T>& lst);
    T operator[](int pos) const;
    T& operator[](int pos);
    void swap(List<T>& lst);
    int search(const T& value) const;
    void sort();
};
```

The `List` Class

- Which data members, if any, should be in the `List` class?

The `List` Class

- Which data members, if any, should be in the `List` class?
- Which member functions' *prototypes* should appear in the `List` class?

The `List` Class

- Which data members, if any, should be in the `List` class?
- Which member functions' *prototypes* should appear in the `List` class?
- Which member functions should be *implemented* in the `List` class?

The `List` Class

- Which data members, if any, should be in the `List` class?
- Which member functions' *prototypes* should appear in the `List` class?
- Which member functions should be *implemented* in the `List` class?
- Which member functions should be virtual?

The `List` Class

- Which data members, if any, should be in the `List` class?
- Which member functions' *prototypes* should appear in the `List` class?
- Which member functions should be *implemented* in the `List` class?
- Which member functions should be virtual?
- Which member functions should be pure virtual?

The `ArrayList` Class

- Which data members, if any, should be in the `ArrayList` class?

The `ArrayList` Class

- Which data members, if any, should be in the `ArrayList` class?
- Which member functions should be *implemented* in the `ArrayList` class?

The `ArrayList` Class

- Which data members, if any, should be in the `ArrayList` class?
- Which member functions should be *implemented* in the `ArrayList` class?
- Which member functions should be *re-implemented* in the `ArrayList` class?

The `CircArrayList` Class

- Which data members, if any, should be in the `CircArrayList` class?

The `CircArrayList` Class

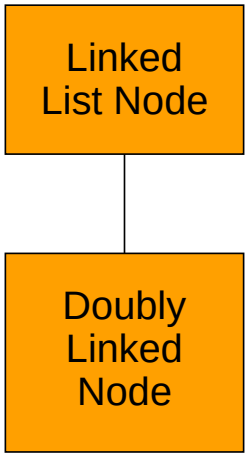
- Which data members, if any, should be in the `CircArrayList` class?
- Which member functions should be *implemented* in the `CircArrayList` class?

The `CircArrayList` Class

- Which data members, if any, should be in the `CircArrayList` class?
- Which member functions should be *implemented* in the `CircArrayList` class?
- Which member functions should be *re-implemented* in the `CircArrayList` class?

Hierarchy of Node Classes

- We could also have a hierarchy of node classes.



The `DoublyLinkedList` Class

- The `LinkedList` class has a `next` pointer to a `LinkedListNode`.
- Could the `DoublyLinkedList` class use this as its `next` pointer to a `DoublyLinkedNode`?

The `DoublyLinkedList` Class

- The `LinkedList` class has a `next` pointer to a `LinkedListNode`.
- Could the `DoublyLinkedList` class use this as its `next` pointer to a `DoublyLinkedNode`?
- If not, then what should we do about it?

Outline

Assignment

Homework

- Read Section 14.5, pages 810 - 822.
- Read Section 14.6, pages 823 - 827.