Bees

Data Structures Spring 2011
The Final Project

Xiao Jia

April 2nd, 2011

The Final Project

- BEES: **B**eginners' **E**ducational **E**mulation of **S**TL
- Deadline: June 20th, 2011 (18th week)
- Grading
  - **Basic functionalities (80%)**
  - **Performance**, reports, etc. (20%)
- Contributors
  - Tianxing He
  - Jiejun Zhang

Containers

		Interface			
		Hash Table	Resizable Array	Balanced Tree	Linked List
Implementation	Set	**HashSet**		**TreeSet**	
	List		**ArrayList**		**LinkedList**
	Map	**HashMap**		**TreeMap**	

There are iterators for these containers.

ArrayList

- `ensureCapacity(int minCapacity)`
- `clear()`
- `size()`
- `isEmpty()`
- `add(const E &e)`
- `add(int index, const E &element)`
- `contains(const E &e)`
- `get(int index)`
- `set(int index, const E &element)`

ArrayList (cont')

- `indexOf(const E &e)`
- `lastIndexOf(const E &e)`
- `removeIndex(int index)`
- `remove(const E &e)`
- `removeRange(int from, int to)`
- `subList(int from, int to)`

ArrayList::Iterator

- `bool hasNext()`
- `E& next()`
  - `retrieve the current element`
  - `advance the iterator itself`
- `void remove()`
  - `throws ElementNotExist`

- `(sorted)`

ArrayList::ConstIterator

- `bool hasNext()`
- `const E& next()`

- `(sorted)`

LinkedList

- `add(int index, const T &elem)`
- `add(const T &elem)`
- `addFirst(const T &elem)`
- `clear()`
- `contains(const T &elem)`
- `get(int index)`
- `getFirst()`
- `getLast()`

LinkedList (cont')

- `indexOf(const T &elem)`
- `isEmpty()`
- `removeIndex(int index)`
- `remove(const T &elem)`
- `removeFirst()`
- `removeLast()`
- `set(int index, const T &elem)`
- `size()`
- `subList(int from, int to)`

LinkedList’s Iterators

- Sorted
- All elements should be iterated.

HashSet / HashMap

```
class Hashint {
public:
  static int hashcode(int obj) {
    return obj;// hash it here
  }
};

HashSet<int, Hashint> hash;
```

HashSet’s Iterators

- Unsorted
- All elements should be iterated.

TreeSet / TreeMap

- Implemented using a **balanced tree**
- Comparison: `operator <`
- The iterator must iterate in the order determined by `operator <` (from the smallest to the largest)

Questions?

- Find out more details on our website.
- Make progress every day.