

## CSCE 146 Lecture 10 - Java Objects and The JCL

- Overview of Today's lecture
  - What was missing from the Sample Test? Chapter 4
  - Java Object Class
  - A Bag of Objects
- Linked List Implementation of Bag
  - section 4.4
  - good test questions
  - countOccurrences
- Comparison Arrays, Linked Lists and Doubly Linked Lists
  - arrays are better at random access
  - linked list are more efficient at insertions/deletions at cursor
  - doubly linked lists are better if you have to move both directions
  - resizing is inefficient for arrays
- Java Objects (ON TEST 2!)
  - 8 primitive types
    1. byte
    2. short
    3. int
    4. long
    5. float
    6. double
    7. char
    8. boolean
  - Everything else is a reference to an Object
  - e.g., String is an object
  - Java provides the superclass Object for any object
  - [javasoft.com](http://javasoft.com) link on Web
- Widening conversions
  - For example

```
String s = new String("Gamecocks beat Nebraska");
Object obj;
obj = s; // this is a widening conversion
```
  - widening conversions are always allowed in Java
  - What does widening really mean?

- Narrowing Conversions

```
String s;
```

```
Object obj = new String("Gamecocks beat Nebraska");
```

```
s = (String) obj;
```

- ClassCastException exception

- If a method returns an Object then typically a narrowing conversion is needed

- Wrapper Classes around primitives

- Why Needed?

- class Integer ... accessor intValue()

- 

⋮

- class Boolean ... accessor booleanValue()

- The Java Object Class

- <http://java.sun.com/j2se/1.4/docs/api/index.html>

Constructor Summary

Object()

Method Summary

Object clone() Creates and returns a copy of this object.

boolean equals(Object obj) Indicates whether some other object is "equal to" this one. protected

void finalize() Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

Class getClass() Returns the runtime class of an object.

int hashCode() Returns a hash code value for the object.

void notify() Wakes up a single thread that is waiting on this object's monitor.

void notifyAll() Wakes up all threads that are waiting on this object's monitor.

String toString() Returns a string representation of the object.

void wait() Causes current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.

void wait(long timeout) Causes current thread to wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.

void wait(long timeout, int nanos) Causes current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object, or some other thread interrupts

the current thread, or a certain amount of real time has elapsed.

- A Bag of Objects
  - allows a bag to contain several different types of Objects
  - IntArrayBag  $\Rightarrow$  ArrayBag, should be just Bag
  - instanceof operator allows to distinguish which type a returned element is
- Add
  - public void add(Object element)
  - **Parameters:** element to be added to the bag
  - **Precondition:** none
  - **Postcondition:** a reference to the specified object has been added to the bag
  - **Throws:** OutOfMemory

```
ArrayBag nouns = new ArrayBag();
nouns.add("animal");
nouns.add(Double(24.7));
System.out.println(nouns.countOccurrences("animal");
```
- equals
  - in general collections should use equals instead of ==
  - there are exceptions current == null should not be changed to equals(null)
- 5 Steps to converting a collection class to hold Java Objects
  - rename the class
  - change underlying type
  - Equality tests
  - Decide on how to treat null references
  - Update documentation
- Deep cloning
  - ArrayBag has object references as elements
  - cloning creates copy, but they are copies of references
  - changing the underlying Object will change both the clone and the original
  - Deep cloning
- ArrayBag Implementation
  - figure 5.1
- Java Application

– figure 5.2

- The Java Class Library

– <http://java.sun.com/j2se/1.4/docs/api/index.html>

– collections of predefined objects

- Collections of the JCL

– vector

– sets

– lists

– maps

- IntNode  $\Rightarrow$  Node

```
public class Node
{
    private Object Data;
    Node link;
}
// the methods use Object Data
```

- Iterators

–

- Java Interface

– usually a list of methods that a class may implement

– Class xbar implements Cloneable;

– For example: AudioClip has methods

```
public void loop()
```

```
public void play()
```

```
public void stop()
```

– any class that want to implement AudioClip needs these

– throws UnsupportedOperationException if not implemented and used

– public class Example implements Cloneable, AudioClip

- instanceof Operator

– with Object type to distinguish if this is say “AudioClip”

– Format: variable-name instanceof interface-or-class-name

```
public static show (Object info)
```

```
if( info instanceof AudioClip)
```

```
    info.play();
```

- Iterator Interface
  - public void hasNext()
  - public void next()
  - public void remove()
- Lab05 - Linked Lists
  - merge sorted lists
  - reverse list
  - split list
- Review for Test
  -
- –
- –
- –
- Questions about Program