

# CSE 8B—Intro to CS: Java

Winter, 2006

March 14 (Day 19)  
Maps (cont.)  
Applets

Instructor: Neil Rhodes

## Map

Used to map from one object (key) to another object (value)

- For example, keep count of how many times a word appeared in a file (mapping from word to number)

### Methods

- `put(K key, V value)`
- `boolean containsKey(K key)`
- `boolean containsValue(V value)`
- `V get(K key)`
- `V remove(K key)`
- `Set<T> keySet()`
- `Collection<T> values()`
- `Entry<K, V> entrySet()`

Inner class  
`Entry<K, V>` stores a key/  
value pair  
Methods:  
`K getKey()`  
`V getValue()`

2

## Map Example

Count occurrences of words in a file

```
Scanner in;  
...  
  
Map<String, Integer> m = new HashMap<String, Integer>();  
while (in.hasNext()) {  
    String s = in.next();  
    Integer num = m.get(s);  
    if (num == null)  
        m.put(s, 1);  
    else  
        m.put(s, num + 1);  
}  
for (Map.Entry<String, Integer> e: m.entrySet())  
    System.out.println(e.getKey() + ": " + e.getValue());
```

3

## Interface vs. Implementation

|            | Hash table | Resizable array | Balanced Tree | Linked list |
|------------|------------|-----------------|---------------|-------------|
| Collection | HashSet    | ArrayList       | TreeSet       | LinkedList  |
| Set        | HashSet    |                 | TreeSet       |             |
| SortedSet  |            |                 | TreeSet       |             |
| List       |            | ArrayList       |               | LinkedList  |
| Map        | HashMap    |                 | TreeMap       |             |
| SortedMap  |            |                 | TreeMap       |             |

4

## Combined Map/List example

### Finds all anagrams

- For each word, adds to a list keyed by alphabetized version of word

```
Map<String, List<String>> m =
    new HashMap<String, List<String>>();
while (in.hasNext()) {
    String s = in.next();
    String alphabetic = alphabetize(s);
    List<String> list = m.get(alphabetic);
    if (list == null)
        m.put(alphabetic, list = new ArrayList<String>());
    list.add(s);
}
for (List<String> l: m.values())
    if (l.size() > 5)
        System.out.println(l.size() + ": " + l);
```

5

## Applets

An applet is a java application embedded inside an HTML page

### Overview

- HTML
- The JApplet container
  - Converting a Swing application to an Applet
- Using the appletviewer tool
- Limitations on applets

6

## HTML

### Quick intro to HTML (HyperText Markup Language)

- Text with embedded *markup tags*
- Markup tags look like:

<tag>

Stuff that is tagged

</tag>

- Example tags:

- HTML: tags the entire HTML contents
- TITLE: tags the title for the window
- BODY: tags the contents within the window
- B: bold font

<HTML>

<TITLE>

This is the title

</TITLE>

<BODY>

This is the <B>Body</B>.

</BODY>

</HTML>

7

## Web browser/Web server interaction

### A web browser requests a URL:

- Web server sends back the HTML page
- Web browser reads the HTML, interprets it, and displays it
  - May require browser making additional requests from the server
    - Images
    - Applets
    - Style sheets
- If user clicks on a link (specifying a URL), browser requests that URL

8

## JApplet container

Another container, like JFrame

Differences:

- No constructor (init method instead)
- No call to super
- No WindowListener or default close operation
- No pack() or specifying the size

Simple applet

```
import javax.swing.*;

public class HelloWorld extends JApplet
{
    public void init() {
        add(new JLabel("Hello, world"));
    }
}
```

9

## Embedding an applet in an HTML file

Use the <applet> tag

- Tag takes parameters:
  - height
  - width
    - These specify the size of the applet
  - code
    - specifies the class file

```
<applet code="HelloWorld.class"
width=300 height=300>
</applet>
```

- In order to display, web browser will download HelloWorld.class from web server. Then, it will run the class within the JVM on the client machine.
  - For our testing, when we open the file from our web browser, it reads the file from the OS. It will also read the .class file directly

10

## Converting a Swing application to an applet

1. Derive class from JApplet instead of JFrame
2. Remove the main method
3. Replace the constructor with an init method

- a) Delete any call to super
- b) Delete any calls to deal with the close window
- c) Delete any calls to add a WindowListener
- d) Delete any calls to setTitle
- e) Delete any calls to setSize
- f) Delete any calls to pack

11

## Example

Converting Tic-Tac-Toe from Swing application to applet.

12

## Using the appletviewer tool

### appletviewer

- Command-line application reads an HTML file, ignores everything but tags specifying an applet, and loads those applets

### Example

- <http://www.faqs.org/docs/javap/c6/s1.html>
- Load in web browser: displays all text, and embedded applets
- Load in appletviewer: opens each applet in a separate window

13

## Limitations on applets

### Applets must be limited in what they can do

- Imagine an applet that plays Lunar Lander, but also reads all your documents and sends them over the internet to some remote server.
- Normal java application has unlimited access to all of your system's resources

### Applets are run in a *sandbox*

- Limited environment for untrusted code. Can't (by default):
  - Read/write files
  - Delete/rename/create files/directories
  - List files in a directory
  - Create a network connection to any computer (other than the host it came from)
  - Create a window (unless it is marked with "Java Applet Window")
  - Listen for network connections
  - Exit the interpreter (System.exit)
  - ...

14