CompSci 6
Programming Design and Analysis

April 20, 2010
Prof. Rodger
Announcements

- Next time: More maps
- Assignment 9 out – Due Thursday
Last Time – Recursive Art

- Two ways to draw art recursively
  - One object
    - Repeatedly draw the same object smaller
  - Multiple objects
    - Each object is “linked” to a smaller object
    - Each object draws itself
    - See the myNext variable
Maps

- Maps are another way of organizing data
- Keys and Values
  - Each key maps to a value
  - Some keys can map to the same value
  - Can change the value a key maps to
Example

- Each student could be mapped to their favorite ice cream flavor

Students

- Astrachan
- Sun
- Rodger
- Forbes

Ice Cream Flavors

- Chocolate
- Chocolate Chip
- Strawberry
Implementing a Map

• We will use TreeMap in Java

• Example:

```java
Map<String, String> fav = new TreeMap<>();
```

• Keys map to values
To use a Map

• Put in a key and its value
  fav.put("Forbes", "Strawberry");
• Get a value for a key
  val = fav.get("Forbes");
• Change value for key
  fav.put("Astrachan", "Coffee Mocha");
Change Astrachan’s value

Students

- Astrachan
- Sun
- Rodger
- Forbes

Ice Cream Flavors

- Coffee Mocha
- Chocolate Chip
- Strawberry
Value could be a set

Students

- Astrachan
- Sun
- Rodger
- Forbes

Ice Cream Flavors

- Coffee Mocha
- Chocolate Mocha
- Vanilla
- Blueberry
- Chocolate Chip
- Blueberry
- Banana
- Strawberry
- Coffee Mocha
Classwork today

- File of words
  - Determine number times each word appears
  - For each word, determine all line numbers it appears on
  - For each alphabetical letter, determine all the words that start with that letter.
First look at methods given

- main
- getWordcounts
  - Given a Scanner bound to a file
  - Return a Map of words to counts
- printResults
  - Given a map print key followed by value
public Map<String, Integer> getWordCounts (Scanner input) {
    Map<String, Integer> results = new TreeMap<String, Integer>();
    
    while (input.hasNext()) {
        String word = input.next();
        
        Integer count = results.get(word);
        if (count == null) {
            results.put(word, 1);
        } else {
            results.put(word, count + 1);
        }
    }
    
    return results;
}
public void printResults (Map<String, ?> results)
{
    for (String key : results.keySet())
    {
        System.out.println(key + "\t" + results.get(key).toString());
    }
}
Output

aid  1
aided, 1
air  1
alarming  1
all  2
all,  1
aloud- 1
am  1
among 2
an  6
and  54
another 2
answer 1
answer. 1
answered 1
any  1
aperture 1
Todo: getLineNumbers

- Map each word to a set of line numbers it occurs on

<table>
<thead>
<tr>
<th>Word</th>
<th>Line Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>offering</td>
<td>[213]</td>
</tr>
<tr>
<td>old</td>
<td>[25, 329]</td>
</tr>
<tr>
<td>on</td>
<td>[33, 58, 86]</td>
</tr>
<tr>
<td>on,</td>
<td>[215]</td>
</tr>
<tr>
<td>on,'</td>
<td>[108]</td>
</tr>
<tr>
<td>on.</td>
<td>[178]</td>
</tr>
<tr>
<td>once</td>
<td>[127]</td>
</tr>
<tr>
<td>one</td>
<td>[30, 58, 93, 96, 224, 230, 245]</td>
</tr>
<tr>
<td>one.</td>
<td>[186]</td>
</tr>
<tr>
<td>only</td>
<td>[9, 325]</td>
</tr>
<tr>
<td>opportunity</td>
<td>[23]</td>
</tr>
<tr>
<td>or</td>
<td>[226, 227]</td>
</tr>
<tr>
<td>orbs</td>
<td>[111]</td>
</tr>
<tr>
<td>orders</td>
<td>[91, 92]</td>
</tr>
<tr>
<td>ornamented</td>
<td>[222]</td>
</tr>
<tr>
<td>other</td>
<td>[19, 244, 245]</td>
</tr>
<tr>
<td>our</td>
<td>[214, 216, 301]</td>
</tr>
</tbody>
</table>
Todo: getFrequencies

• Map each letter of alphabet to words

[The, Then, There, These, They, Three, Throwing, Unsheathing]
[We, When, Will, With, Withdrawing, Within]
[You, You, Your]
["A, "Amontillado!", "Amontillado?, "And,
[a, about, above, absconded, accosted, admired,
[back, back:, back:, backed, bargain.'', be, be,
called, came, cannot, cap, carnival, carnival!
d["or,, damp, damp., dampness, damps.'', dear,
e[each, earliest, earth,, eighth,, ejaculated, etc.
f[fabric, face,, fails, fall, familiarly,, family
[gait, gave, gemmary,, gesticulation, getting,
h["a!, ha!--he!, had, hairs, half, hand, hand,'',
i[idea, ignoramus,'', ill,, imbedded, immediate,
j[jest,'', jest., jingled, jingled., jingling, jo
k[key, kill, knew,, knocked, know]
l[labour., labours, laces, lacist.'', laid, largely,
m[made, madness, make, man, man., manner., many,