CompSci 6
Programming Design and Analysis

December 1, 2009

Prof. Rodger and Forbes
Announcements

• Assignment out today
• Review next time
• Final Exam
  – Forbes class: Tuesday, Dec 8, 9am-noon
  – Rodger’s class: Wednesday, Dec 9, 7-10pm
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
Implementing a Map

• We will use TreeMap in Java

• Example:
  
  Map<String, String> fav =
  new TreeMap<String, String>();

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

• File of words
  – Determine number times each words 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

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>aid</td>
<td>1</td>
</tr>
<tr>
<td>aided,</td>
<td>1</td>
</tr>
<tr>
<td>air</td>
<td>1</td>
</tr>
<tr>
<td>alarming</td>
<td>1</td>
</tr>
<tr>
<td>all</td>
<td>2</td>
</tr>
<tr>
<td>all,</td>
<td>1</td>
</tr>
<tr>
<td>aloud-</td>
<td>1</td>
</tr>
<tr>
<td>am</td>
<td>1</td>
</tr>
<tr>
<td>among</td>
<td>2</td>
</tr>
<tr>
<td>an</td>
<td>6</td>
</tr>
<tr>
<td>and</td>
<td>54</td>
</tr>
<tr>
<td>another</td>
<td>2</td>
</tr>
<tr>
<td>answer</td>
<td>1</td>
</tr>
<tr>
<td>answer.</td>
<td>1</td>
</tr>
<tr>
<td>answered</td>
<td>1</td>
</tr>
<tr>
<td>any</td>
<td>1</td>
</tr>
<tr>
<td>aperture</td>
<td>1</td>
</tr>
</tbody>
</table>
Todo: getLineNumbers

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

• Map each letter of alphabet to words