CS 121 - Intro to Programming: Lecture 25

Announcements

Program 7 Up today
Ch 11 OWL exercises due tomorrow
Ch 12 OWL exercises up

Final Exam: Wednesday, Dec 17, 8 AM
New ideas:

Graphics elements that deal with files and text

ArrayList - a new data structure (A trapdoor at the end of 7.3 in text)
Next programming assignment - due 12/5
Official posting later today
The skyline project
A demo
import java.util.ArrayList;
public class ArrayListTester{
    public static void main(String[] args){
        ArrayList<Infant> kids = new ArrayList<Infant>();
        kids.add( new Infant("Jill", 11));
        kids.add( new Infant("Jake", 7));
        kids.add( new Infant("Justin", 4));
        kids.add( new Infant("Zeke", 19));
        kids.add( new Infant("Beth", 5));
        for(Infant kid : kids)
            System.out.println(kid.getName() + " " + kid.getAge());

        Infant i = kids.get(1);
        System.out.println(i.getName());
    }
}
ArrayList generalizes array (no size limit)
Must be ArrayList of some kind of object
You lose the [] notation though

Bracket notation:
ArrayList<Infant> kids = new

    ArrayList<Infant>();

You identify the type (class) of the objects on the list. This angle-bracket notation - generics - a big deal in the next class (187)

Discussed in a trapdoor in section 7.3
import java.util.*;

public class ArrayListTester2{
    public static void main(String[] args){
        ArrayList<String> words = new ArrayList<String>();
        words.add("max");    words.add("tax");
        words.add("lax");     words.add("lipid");
        words.add("munch");
        for(String w : words)
            System.out.print(w + " ");
        Collections.sort(words);
        System.out.println();
        for(String w : words)
            System.out.print(w + " ");
    }
}
Program’s output:

- java ArrayListTester2

max tax lax lipid munch
lax lipid max munch tax
Some **ArrayList** methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>add(e)</td>
<td>appends new element e as last cell</td>
</tr>
<tr>
<td>remove(i)</td>
<td>removes element at location i</td>
</tr>
<tr>
<td>set(i,e)</td>
<td>sets element at position i to e</td>
</tr>
<tr>
<td>get(i)</td>
<td>returns element at position i in the ArrayList</td>
</tr>
<tr>
<td>size()</td>
<td>returns number of elements in ArrayList</td>
</tr>
</tbody>
</table>
A complex GUI application

I want to examine an arbitrarily large text file - a novel, say - and report word frequencies in alphabetical order.

I want a set of GUI controls on this process
What is the Central feature/object/(data) structure of the application?
An ArrayList of `WordInfo` objects...

<table>
<thead>
<tr>
<th></th>
<th>word: dog</th>
<th>count: 7</th>
</tr>
</thead>
</table>

A typical entry on the ArrayList...
What behaviors does this structure require?
Build it

It’s got to live somewhere, in some kind of object

I’ve got to be able to make the object

I’ve got to be able to sort it alphabetically

I need to display what I’ve gotten

If it’s big (long), I’ve got to be able to scroll through it

File comes in as lines - I’ve got to chop these up into words
I’ll go with WordVault class - it’s principal attribute will be an ArrayList of WordInfo objects..

------------------------------------------
import java.util.*;

public class WordVault{

    private ArrayList<WordInfo> words = new
    ArrayList<WordInfo>();

    .....
Two broader mechanisms surround the WordVault object

• The file reading piece

• The controls / display piece
The file piece

Extend LineReader (abstract class)

LineReader looks a little different - a second constructor

We want to use JFileChooser - a graphical file selection mechanism from the swing library (javax.swing) [discussed in ch 12!]

public abstract class LineReader{
    String fileName; // external file name
    Scanner scan; // obj for reading from external file

    public LineReader(String f) throws IOException
    {
        fileName = f;
        scan = new Scanner(new FileReader(fileName));
    }

    public LineReader(File aFile) throws IOException
    {
        scan = new Scanner(aFile);
    }

    ...
}
The controls / display piece

The usual DisplayWindow / JPanel extension (FreqPanel)

One very important point:

Commands in FreqPanel (which is both the app controller and the app display mechanism) need direct access to the principal object of the application - the WordVault object

How do we do this??
public class FreqPanel extends JPanel implements ActionListener {

    JTextArea textarea = new JTextArea(20,50);
    JScrollPane scroller = new JScrollPane(textarea);
    JFileChooser chooser;

    WordVault vault; // references main app obj
    WordFetch fetcher; // reads the file
    JButton processWords = new JButton("Report Words");
    JButton file = new JButton("Choose File");
    JButton quit = new JButton("Quit");

    public FreqPanel(WordVault v){
        vault = v;
        ...........
    }
}
The heart of the matter...

```java
if (e.getSource() == quit) System.exit(0);
else
    if(e.getSource() == file){
        chooser = new JFileChooser();
        int returnVal = chooser.showOpenDialog(null);
        fetcher = new WordFetch(chooser.getSelectedFile(),vault);}
else
    if (e.getSource() == processWords){
        processFile();
        writeToPanel();
        vault.wordReport(); // stats to console
    }
```
public void processFile(){
    fetcher.readLines();  // builds vault
    (fetcher.getVault()).sortWords(); // sorts vault
}

public void writeToPanel(){
    for(WordInfo w : vault.getWords())
        textarea.append(w.toString() + 'n');
}
How would you add a search feature?

(How many times does “donkey” appear in the text?)
TextField for word submission
Button for “Search”
A “clear” mechanism for the textarea
Where would the actual search take place? (which method (new or old) in which class (new or old))