CS 121 - Intro to Java - Lecture 26

Announcements

Program 7 due Friday. OWL 13 due Friday

Final Exam: Friday 12/18, 130 -330

Special arrangements? Let me hear from you!
What’s on the final? - Obvious emphasis on material from last half of class:

Arrays
Inheritance
Files
Interfaces
Graphics
The event Model

Also: randomness, exceptions, layouts, methods
Other possible topics:

• Recursion
• ArrayLists
• Menus
Some sample problems..
int value = 0;

for (int j = big; j > 0; j = j/2)
    value++;

System.out.println(value);

1. What does this loop do when big is 20? 0?
2. Rewrite as a while loop
3. Are there values of big for which this will run forever?
int j = big;

while(j > 0)
{
    j = j/2;
    value++;
}
System.out.println(value);
public class Bowl{
    private double weight;
    private boolean empty;
    private String origin;

    public Bowl(double w, boolean e, String origin){
        weight = w;
        empty = e;
        this.origin = origin;
    }

    public double getWeight(){return weight;}
    public boolean getEmpty(){return empty;}
    public String getOrigin(){return origin;}

    public String toString(){
        return("from "+origin+" weight: " + weight);
    }
}
Write a `setEmpty` method that allows you to set the empty attribute to true or false.

Explain how `toString` could possibly work properly, since the return type is `String`, but the weight instance variable is part of the return expression.

Write a statement that would appear in a driver class and would create a `Bowl` object called `myBowl` that weighs .65 Kg, is empty, and comes from Italy.

Add a method to the `Bowl` class called `majorityEmpty`, which is passed an array of `Bowl` objects, and returns true if strictly more than half of the bowls in the array are empty.
public void setEmpty(boolean newVal)
{
    empty = newVal;
}
public static boolean majorityEmpty(Bowl[] bowls) {
    int ctr = 0;
    for (Bowl b : bowls)
        if (b.empty) ctr++;
        else ctr--;
    return (ctr > 0);
}
Write a one class application that reads a single integer from the keyboard using a Scanner object. If that value is negative, the program should print “negative”; otherwise the program should print the square root of the number.

-------------------

What does the import statement do?
Use inheritance to extend the Bowl class to a new class called OvenProofBowl. This class should add one attribute to the base class, the boolean attribute “ovenproof”. Be sure to include in your class declaration

1) a constructor that takes four parameters;
2) get and set methods for the ovenproof attribute; and
3) a version of toString that includes an embedded call to toString from the base class.
public class OvenPfBowl extends Bowl{
    private boolean ovenproof;
    public OvenPfBowl(double w; boolean e; String s;
                        boolean pf){
        super(w,e,s);
        ovenproof = pf;
    }
    public String toString(){
        String s = super.toString();
        return(s + " ovenproof? " + ovenproof);
    }
}
Which of these is legal?

a. int a = 5;  double r = 4.3;  a = a/r;
b. int a = 5;  double r = 4.3;  a = (int)a/r;
c. int a = 5;  double r = 4.3;  a = a/(int)r;
d. int a = 5;  double r = 4.3;  a = (int)(a/r);
public class BankDriver{
    public static void main(String[] args){
        int pennies = 143; int nickels = 94;
        int dimes = 11; int quarters = 44;
        PiggyBank b = new PiggyBank(pennies,nickels,dimes,quarters);
        System.out.println("decimal fraction of coins that are quarters");
        System.out.println(b.quarterFraction());
        System.out.println("total dollar value,in dollars,cents");
        System.out.println(b.totalValue());
    }
}

// give PiggyBank class definition
Suppose mystery is defined in class SomeClass

```java
public static int mystery(int[] a, int j){
    if (j < 0) return 1;
    else return(a[j]*mystery(a,j-1));
}
```

Given this array definition:
```java
    int[] b = {2,4,6};
```
What does this method call return:
```java
    SomeClass.mystery(b,2);
```

Then explain in a sentence or two what mystery does in general.
SomeClass.mystery(b,2)   [b = {2, 4, 6} ]

-> 6*mystery(b,1)

-> 6*(4*mystery(b,0))

-> 6*(4*(2*(mystery(b,-1)))

non-recursive, = to 1
import java.awt.*;
import javax.swing.*;
import java.io.*;

public class FileFreqDriver{
    public static void main(String[] args{
        try{
            try{
                DisplayWindow display = new DisplayWindow();
                WordVault vault = new WordVault();
                FreqPanel p = new FreqPanel(vault);
                display.add(p);
                display.showFrame();
            }
            catch(Exception e){e.printStackTrace();}
        }
    }
}
public class FreqPanel extends JPanel implements ActionListener {

    JTextArea textarea = new JTextArea(40, 50);
    JScrollPane scroller = new JScrollPane(textarea);
    JFileChooser chooser;
    WordVault vault;
    WordFetch fetcher;
    JButton file = new JButton("Choose File");
    JButton quit = new JButton("Quit");
    JButton search = new JButton("Search");
    JButton clear = new JButton("Clear");
    JButton all = new JButton("Show All Words");
    JTextField theWord = new JTextField(20);

    public FreqPanel(WordVault v){

    }
public FreqPanel(WordVault v) {
    vault = v;
    textarea.setEditable(false); // can't edit text area
    this.add(file);
    this.add(all);
    this.add(clear);
    this.add(theWord);
    this.add(search);
    this.add(quit);
    this.add(scroller); // scrollable text area to panel
    file.addActionListener(this);
    clear.addActionListener(this);
    search.addActionListener(this);
    all.addActionListener(this);
    quit.addActionListener(this);
}
... else if (e.getSource() == search) {
    clearArea();
    String s = theWord.getText();
    WordInfo i = vault.findWord(s);
    if (i == null) textarea.append(s + "  " + 0);
    else {textarea.append(i.toString());}
}
public class WordVault{

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

public void updateWords(String w){
    WordInfo which = findWord(w);
    if (which != null) which.incCount(); else addWord(w);
}

public void clearVault(){words = new 
            ArrayList<WordInfo>();
}

public WordInfo findWord(String w){
    for(WordInfo i : words){
        if(i.getWord().equals(w))return i;}
    return null;}
}