Programming with Alice & Java

Week 6 Transition to Java

Chapter Objectives

• Compare the concepts you saw in Alice to their counterparts in Java
• Describe program development environments for Java
• Access the Java API support library and its online documentation
• Explore several Java program examples
• Practice using various Java statements

Oracle Java Tutorials

Chapter 6 PowerPoint
Source Code for CH3

6.1 Comparing Alice and Java
Table 6.1 pages 100–101

Program Development

Alice is an integrated development environment (IDE) specific to Alice programs.
There are many IDEs available where you can create and test Java programs. We use TextPad for CS 12J
and it is the IDE that is supported in the CTC.
Java programs must be translated into a form that is executable.
A programmer writes classes in Java source code; a compiler translates the source code into bytecode.
The bytecode is then interpreted and executed when the program is fun. Yep, this was all hidden in
Alice.

The compiler also helps with program syntax and you have a hint as to what needs correcting; semantic
errors (errors in meaning, also called logic errors) are not caught by the compiler.

Classes and Objects

In Alice, to create an object, we clicked on a gallery object and it appeared in our world.
In Java, we create an object by executing a programming statement (that means typing...)

A constructor is used to set up a newly created object.
Alice classes are organized into galleries; Java classes are organized into packages. You need to import these packages in order for your program to be able to use them.

Java API (application programming interface) is a library of pre-defined classes: hundreds of classes organized into dozens of packages. The Java API is your friend.

Inheritance
Polymorphism
Encapsulation

Data and Operators

<table>
<thead>
<tr>
<th>integers</th>
<th>floating point numbers</th>
<th>character data type</th>
<th>Boolean data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte, short, int, long</td>
<td>float, double</td>
<td>char</td>
<td>boolean</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alice</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>set value</td>
</tr>
<tr>
<td>Basic Arithmetic</td>
<td>+, -, *, /</td>
</tr>
<tr>
<td>Remainder</td>
<td>IEEEERemainder</td>
</tr>
<tr>
<td>Equality</td>
<td>==, !=</td>
</tr>
<tr>
<td>Relational</td>
<td>&lt;, &lt;=, &gt;, &gt;=</td>
</tr>
<tr>
<td>Logical NOT</td>
<td>!a</td>
</tr>
<tr>
<td>Logical AND</td>
<td>both a and b</td>
</tr>
<tr>
<td>Logical OR</td>
<td>either a or b, or both</td>
</tr>
</tbody>
</table>

Statements

assignment: sum = sum + 20;
print: System.out.println("The value of sum is: + sum);
if/else: if (sum > 30) System.out.println("The value of sum is over thirty"); else System.out.println("The value of sum is less than or equal to thirty");
while/for: while (count > 30) { do something} for (count =2; count < 10; count++) { do something}
main method: where every Java program begins executing
6.2 Java and Objects

DrawCircles, Bullseye
TryThis: 1–5

```java
public class DrawCircles {
    public static void main(String[] args) {
        CirclePanel panel = new CirclePanel();
        JFrame frame = new JFrame("Draw Circles");
        frame.getContentPane().add(panel);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
```

6.3 Java Statements

CoinFlip
TryThis: 6–7

The `toString` method gets called automatically when `print` or `println` are called on an object. It’s good practice to define a `toString` method for most objects in programs you create.

```java
public void paintComponent(Graphics go) {
    super.paintComponent(go);
    int x = 0, y = 0, diameter = MAX_WIDTH;
    go.setColor(Color.white);
    for (int count = 0; count < NUM_RINGS; count++)
    {
        if (go.getColor() == Color.black) // alternate colors
            go.setColor(Color.white);
        else
            go.setColor(Color.black);
        go.ellipse(x, y, diameter, diameter);
        diameter = diameter - (2 * RING_WIDTH);
        x = x + RING_WIDTH;
        y = y + RING_WIDTH;
    }
    // Draw the red bullseyes in the center
    go.setColor(Color.red);
    go.ellipse(x, y, diameter, diameter);
}
```
What objects are created? What control structure(s) do you recognize?

```java
//CoinFlip.java Programming with Alice and Java
//
//Demonstrates the use of various Java statements.
//******************************************************************************

import java.util.Scanner;

public class CoinFlip
{
    //******************************************************************************
    // Flips a coin several times, counting the number of heads and tails
    // that result.
    //******************************************************************************
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("How many flips (1-1000)? ");
        int flips = scan.nextInt();
        while (flips < 1 || flips > 1000)
        {
            System.out.print("Not in range. How many flips (1-1000)? ");
            flips = scan.nextInt();
        }
        Coin myCoin = new Coin();
        int heads = 0, tails = 0;
        for (int count = 0; count <= flips; count++)
        {
            myCoin.flip();
            if (myCoin.isHeads())
                heads = heads + 1;
            else
                tails = tails + 1;
            System.out.println();
        }
        System.out.println("Total number of flips: " + flips);
        System.out.println("Heads: " + heads + " Tails: " + tails);
    }
}

6.4 Skip

6.5 More to Explore
- Java API: explore the String class in Java
- Graphics Methods are found in the Graphics class.
- Additional Operators: +=, -=, /=, %=, *=

Homework