Using Classes and Objects

CSC 2014 – Java Bootcamp
Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Packages

• For purposes of accessing them, classes in the Java API are organized into packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang</td>
<td>General support</td>
</tr>
<tr>
<td>java.applet</td>
<td>Creating applets for the web</td>
</tr>
<tr>
<td>java.awt</td>
<td>Graphics and graphical user interfaces</td>
</tr>
<tr>
<td>javax.swing</td>
<td>Additional graphics capabilities</td>
</tr>
<tr>
<td>java.net</td>
<td>Network communication</td>
</tr>
<tr>
<td>java.util</td>
<td>Utilities</td>
</tr>
<tr>
<td>javax.xml.parsers</td>
<td>XML document processing</td>
</tr>
</tbody>
</table>

imported automatically, includes String and Math classes

The Math Class

• The Math class contains methods that perform various mathematical functions

• These include:
  – absolute value
  – square root
  – exponentiation
  – trigonometric functions

value = Math.cos(90) + Math.sqrt(delta);
The Math Class

- The Math class is part of the java.lang package
  - No need to import anything!
  - The Math class methods are static
  - Static methods are invoked through the class name

```
value = Math.cos(phi) + Math.sqrt(delta);
```

The Random Class

- Part of the java.util package, so import it
  ```
  import java.util.Random;
  ```

- Create a Random object named gen:
  ```
  Random gen = new Random();
  ```

- Use Random method nextInt() to generate a random number:
  ```
  int a = gen.nextInt(4);
  // integer in range [0,1,2,3]
  ```

Some methods from the Math class

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(double a)</td>
<td>absolute value of a</td>
</tr>
<tr>
<td>max(double a, double b)</td>
<td>maximum of a and b</td>
</tr>
<tr>
<td>min(double a, double b)</td>
<td>minimum of a and b</td>
</tr>
<tr>
<td>sin(double theta)</td>
<td>sine function</td>
</tr>
<tr>
<td>cos(double theta)</td>
<td>cosine function</td>
</tr>
<tr>
<td>tan(double theta)</td>
<td>tangent function</td>
</tr>
<tr>
<td>exp(double a)</td>
<td>exponential(e)</td>
</tr>
<tr>
<td>log(double a)</td>
<td>natural log (log, a, or ln a)</td>
</tr>
<tr>
<td>pow(double a, double b)</td>
<td>raise a to the highest power a^b</td>
</tr>
<tr>
<td>round(double a)</td>
<td>round to the nearest integer</td>
</tr>
<tr>
<td>random()</td>
<td>random number in [0, 1)</td>
</tr>
<tr>
<td>sqrt(double a)</td>
<td>square root of a</td>
</tr>
<tr>
<td>E</td>
<td>value of e (constant)</td>
</tr>
</tbody>
</table>

What is a random number?

“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.”
- John Von Neumann

“God does not play dice.”
- Albert Einstein

The Random class provides methods that generate pseudorandom numbers
Example: Using Random methods

```java
Random gen = new Random();
int a = gen.nextInt(4);
    // integer in range [0,1,2,3]
float b = gen.nextFloat();
    // float in range [0,1), eg: 0.4589
int c = gen.nextInt(4) + 1;
    // int in range [1,2,3,4]
int d = gen.nextInt();
    // int in range [-2147483648 ... 2147483647]
```

List of some Random methods: page 126

See `RandomNumbers.java`

Example: counting “snake eyes”

```java
// Roll two dice 100,000 times and count how many
// times you roll snake eyes, i.e., two 1's.
Random gen = new Random();
int trial = 0, count = 0;
while (trial < 100000)
{
    int die1 = gen.nextInt(6) + 1;
    int die2 = gen.nextInt(6) + 1;
    if (die1 == 1 && die2 == 1)
        count++;
    trial++;
}
System.out.println("Probability of snake eyes = " +
        (double)count/100000);
```

The Strings Class

- Strings are objects defined by the `String` class
- the `String` class has many methods that can be used to process text. Examples:
  - finding the length of a string
  - finding the char at a certain position of a string
  - producing an all-caps version of a string

Invoking String Methods

```java
String name = "Betsy";
int numOfCharsInName = name.length();
```

As with other kinds of objects, we use the **dot operator** to invoke a String's methods:

```java
length() is one of the methods of String objects (defined in the String class)
```
More String Methods

- List of some String methods: see textbook, page 119

```java
String name = "Betsy";
char initial = name.charAt(0);
String newName = name.replace('s', 't');
String capsName = name.toUpperCase();
int comp = name.compareTo(newName);
```

See also textbook example `StringMutation.java`

Example: Palindrome tester

- Problem: Input a string, determine whether it is a palindrome, i.e.:
  - first char is the same as last char
  - 2nd char is the same as 2nd last char
  - and so on...
- How to express this as an algorithm?
- How to implement it?

```java
System.out.println("Enter a potential palindrome:");
str = scan.nextLine();
left = 0;
right = str.length() - 1;
while (str.charAt(left) == str.charAt(right) && left < right)
{
    left++;
    right--;
}
if (left < right)
    System.out.println("NOT a palindrome");
else
    System.out.println("palindrome");
```

Sample Run

Enter a potential palindrome: radar
palindrome
Test another palindrome (y/n)? y
Enter a potential palindrome: able was I ere I saw elba
palindrome.
Test another palindrome (y/n)? y
Enter a potential palindrome: abracadabra
NOT a palindrome.
Test another palindrome (y/n)? n
Declaring Variables, revisited

- Examples of variable declarations:
  
  ```java
  int count = 0;
  double mpg;
  String title;
  Graphics page;
  Color aquamarine;
  Scanner scan;
  ```

- A class name can be used as a type to declare an object reference variable
- The object itself must be created separately

Creating Objects

- We have already seen something like this:
  ```java
  Scanner scan = new Scanner (System.in);
  ```

The new operator calls the Scanner constructor, which is a special method that sets up the object

Variable refers to a Scanner object

Constructing a new object is called instantiation

- Another example:
  ```java
  String title = new String ("Java Software Solutions");
  ```

Creating Objects

The String Class is SPECIAL!

- Exception to the use of new operator: Because strings are so common, we don't have to use the new operator to create a String object
  ```java
  String title = new String ("Java Software Solutions");
  ```

- This is special syntax that works only for strings
  ```java
  String title = "Java Software Solutions";
  ```
The Java API

Another example: EvenOdd.java

/********************************************************************
// EvenOdd.java  Author: Lewis/Loftus
// Demonstrates the use of the JOptionPane class.
import javax.swing.JOptionPane;
public class EvenOdd
{
    // Determines if the value input by the user is even or odd.
    // Uses multiple dialog boxes for user interaction.
    public static void main (String[] args)
    {
        String numStr, result;
        int num, again;
        do
        {
            numStr = JOptionPane.showInputDialog("Enter an integer: ");
            num = Integer.parseInt(numStr);
            result = "That number is " + ((num%2 == 0) ? "even" : "odd");
            JOptionPane.showMessageDialog(null, result);
            again = JOptionPane.showConfirmDialog(null, "Do Another?");
        }
        while (again == JOptionPane.YES_OPTION);
    }
}