Introduction to Computers and Programming

Lecture 16

Chris Conway
New York University
Road Map

- More array examples.
- Passing arrays to methods

- Reading: Chapter 6, Section 6.4
review

• What is an array?
• What is a subscript?
• What is another term for a subscript?
• Given an array of size n, what is the valid range of subscripts?
• How can you obtain the size of an array in Java?
• What happens if you try to access an element outside the valid range of subscripts?
• What are the default values given to array elements in Java?
7.4 Examples Using Arrays (Cont.)

• Histogram

• Using the elements of an array as counters
  – Use a series of counter variables to summarize data
// Fig. 7.7: RollDie.java
// Roll a six-sided die 6000 times.
import javax.swing.*;

public class RollDie {
    public static void main( String args[] )
    {
        int frequency[] = new int[7];

        // roll die 6000 times; use die value as frequency index
        for ( int roll = 1; roll <= 6000; roll++ )
            ++frequency[1 + ( int ) ( Math.random() * 6 )];

        String output = "Face\tFrequency"
        // append frequencies to String output
        for ( int face = 1; face < frequency.length; face++ )
            output += "\n" + face + "\t" + frequency[ face ];

        System.out.println("Rolling a Die 6,000 times");
        System.out.println ( output );
        System.exit( 0 );
    } // end main
} // end class RollDie
7.4 Examples Using Arrays (Cont.)

• Using arrays to analyze survey results
  – 40 students rate the quality of food
    • 1 - 10 Rating scale: 1 mean awful, 10 means excellent
  – Place 40 responses in array of integers
  – Summarize results
// Fig. 7.8: StudentPoll.java
// Student poll program.
import javax.swing.*;

public class StudentPoll {

    public static void main( String args[] ) {
        int responses[] = {
            1, 2, 6, 4, 8, 5, 9, 7, 8, 10, 1, 6, 3, 8, 6,
            10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7, 5, 6, 6, 5, 6, 7, 5, 6,
            4, 8, 6, 8, 10
        };

        int frequency[] = new int[11];

        // for each answer, select responses element and use that value
        // as frequency index to determine element to increment
        for ( int answer = 0; answer < responses.length; answer++ )
            ++frequency[ responses[ answer ] ];

        String output = "Rating\tFrequency\n";

        // append frequencies to String output
        for ( int rating = 1; rating < frequency.length; rating++ )
            output += rating + "\t" + frequency[ rating ] + "\n";

        JOptionPane.showMessageDialog( null, output );

        System.exit( 0 );
    }
}

©2003 Prentice Hall, Inc.
All rights reserved.
<table>
<thead>
<tr>
<th>Rating</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>
7.4 Examples Using Arrays (Cont.)

• Some additional points
  – When looping through an array
    • Index should never go below 0
    • Index should be less than total number of array elements
  – When invalid array reference occurs
    • Java generates ArrayIndexOutOfBoundsException
      – Chapter 15 discusses exception handling
7.6 Passing Arrays to Methods

• To pass array argument to a method
  – Specify array name without brackets
    • Array hourlyTemperatures is declared as
      int[] hourlyTemperatures = new int[24];

• The method call
  modifyArray( hourlyTemperatures );

• Passes array hourlyTemperatures to method
  modifyArray
• In the method header, use similar syntax as that for array declaration:

    public static void modifyArray(int[] array) 
    or

    public static void modifyArray(int array[])
7.5 References and Reference Parameters

- Two ways to pass arguments to methods
  - Pass-by-value
    - Copy of argument’s value is passed to called method
    - In Java, every primitive is pass-by-value
  - Pass-by-reference
    - Caller gives called method direct access to caller’s data
    - Called method can manipulate this data
    - Improved performance over pass-by-value
    - In Java, every object is pass-by-reference
      - In Java, arrays are objects
        - Therefore, arrays are passed to methods by reference
        - Technically, we are using pass by value but the value we are passing is a reference to the array.