Building Java Programs

Chapter 7
Lecture 7-1: Arrays

**reading: 7.1**
self-checks: #1-9
videos: Ch. 7 #4

Can we solve this problem?

- Consider the following program (input underlined):

  How many days' temperatures? 7  
  Day 1's high temp: 45  
  Day 2's high temp: 44  
  Day 3's high temp: 39  
  Day 4's high temp: 48  
  Day 5's high temp: 37  
  Day 6's high temp: 46  
  Day 7's high temp: 53  
  Average temp = 44.6  
  4 days were above average.
Why the problem is hard

- We need each input value twice:
  - to compute the average (a cumulative sum)
  - to count how many were above average

- We could read each value into a variable... but we:
  - don't know how many days are needed until the program runs
  - don't know how many variables to declare

- We need a way to declare many variables in one step.

Arrays

- **array**: object that stores many values of the same type.
- **element**: One value in an array.
- **index**: A 0-based integer to access an element from an array.

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>12</td>
<td>49</td>
<td>-2</td>
<td>26</td>
<td>5</td>
<td>17</td>
<td>-6</td>
<td>84</td>
<td>72</td>
<td>3</td>
</tr>
</tbody>
</table>

- element 0
- element 4
- element 9
Array declaration

```java
    type[] name = new type[length];
```

- Example:
  ```java
    int[] numbers = new int[10];
  ```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Array declaration, cont.

- The length can be any integer expression.
  ```java
  int x = 2 * 3 + 1;
  int[] data = new int[x % 5 + 2];
  ```

- Each element initially gets a "zero-equivalent" value.

<table>
<thead>
<tr>
<th>Type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>0</td>
</tr>
<tr>
<td>double</td>
<td>0.0</td>
</tr>
<tr>
<td>boolean</td>
<td>false</td>
</tr>
<tr>
<td>String</td>
<td>null (means, &quot;no object&quot;)</td>
</tr>
</tbody>
</table>
Accessing elements

name[index] // access
name[index] = value; // modify

- Example:
  numbers[0] = 27;
  numbers[3] = -6;

  System.out.println(numbers[0]);
  if (numbers[3] < 0) {
      System.out.println("Element 3 is negative.");
  }

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>-6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Arrays of other types

double[] results = new double[5];
results[2] = 3.4;
results[4] = -0.5;

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0.0</td>
<td>0.0</td>
<td>3.4</td>
<td>0.0</td>
<td>-0.5</td>
</tr>
</tbody>
</table>

boolean[] tests = new boolean[6];
tests[3] = true;

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
Out-of-bounds

- Legal indexes: between 0 and the array's length - 1.
  - Reading or writing any index outside this range will throw an ArrayIndexOutOfBoundsException.

Example:
```java
int[] data = new int[10];
System.out.println(data[0]);       // okay
System.out.println(data[9]);       // okay
System.out.println(data[-1]);      // exception
System.out.println(data[10]);      // exception
```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Accessing array elements

```java
int[] numbers = new int[8];
numbers[1] = 3;
numbers[4] = 99;
numbers[6] = 2;
int x = numbers[1];
numbers[x] = 42;
numbers[numbers[6]] = 11; // use numbers[6] as index
```

<table>
<thead>
<tr>
<th>x</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>0</td>
</tr>
<tr>
<td>numbers value</td>
<td>0</td>
</tr>
</tbody>
</table>
Arrays and for loops

- It is common to use for loops to access array elements.
  ```java
  for (int i = 0; i < 8; i++) {
    System.out.print(numbers[i] + " ");
  }
  System.out.println(); // output: 0 4 11 0 44 0 0 2
  ```

- Sometimes we assign each element a value in a loop.
  ```java
  for (int i = 0; i < 8; i++) {
    numbers[i] = 2 * i;
  }
  ```
  
<table>
<thead>
<tr>
<th>index</th>
<th>0 1 2 3 4 5 6 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
</tbody>
</table>

The length field

- An array's length field stores its number of elements.
  ```java
  name.length
  ```
  ```java
  for (int i = 0; i < numbers.length; i++) {
    System.out.print(numbers[i] + " ");
  }
  // output: 0 2 4 6 8 10 12 14
  ```
  - It does not use parentheses like a String's .length().

- What expressions refer to:
  - The last element of any array?
  - The middle element?
Weather question

- Use an array to solve the weather problem:

  How many days' temperatures? 7
  Day 1's high temp: 45
  Day 2's high temp: 44
  Day 3's high temp: 39
  Day 4's high temp: 48
  Day 5's high temp: 37
  Day 6's high temp: 46
  Day 7's high temp: 53
  Average temp = 44.6
  4 days were above average.

Weather answer

// Reads temperatures from the user, computes average and # days above average.
import java.util.*;
public class Weather {
  public static void main(String[] args) {
    Scanner console = new Scanner(System.in);
    System.out.print("How many days' temperatures? ");
    int days = console.nextInt();
    int[] temperatures = new int[days]; // array to store days' temperatures
    int sum = 0;
    for (int i = 0; i < days; i++) {
      System.out.print("Day "+ (i + 1) + "'s high temp: ");
      temperatures[i] = console.nextInt();
      sum += temperatures[i];
    }
    double average = (double) sum / days;
    int count = 0;
    // see if each day is above average
    for (int i = 0; i < days; i++) {
      if (temperatures[i] > average) {
        count++;
      }
    }
    // report results
    System.out.printf("Average temp = %.1f", average);
    System.out.println(count + " days above average");
  }
}