Topics: One-dimensional array, generating random numbers

Reading (T): Sec 8.1-8.3

Arrays

- Arrays are objects. An array is an ordered list of values (or objects) of one type
- The entire array has one name (identifier)
- Each element in the array has an integer index (begins at 0)
- An array of length \( N \) is indexed from 0 to \( N-1 \)

Array declaration and construction

- Declaration syntax: \texttt{type[]} \texttt{identifier};
  Examples:
  \begin{verbatim}
  int[] counts;
  double[] price;
  String[] names;
  Interval[] series;  // assuming an Interval class has been defined
  \end{verbatim}
- Instantiation syntax: \texttt{new type[ size ]}
  size is an integer expression
  Example:
  \begin{verbatim}
  new int[4]
  \end{verbatim}
- Declaration and instantiation
  \begin{verbatim}
  int limit = 4;
  double[] price;   // declaration
  price = new double[limit];  // instantiation and assignment
  \end{verbatim}
- Creating an array using an initializer list

The size of an array is held in the constant \texttt{length}. \texttt{length} is automatically defined when an array is created and \textit{cannot be changed}. In the above example, the expression \texttt{price.length} gives the size of the array \texttt{price}.

Index operator \[\texttt{[]}\]

The expression \texttt{identifier[integer_expression]} accesses an element in the array referred to by \texttt{identifier}

Examples:
\begin{verbatim}
int[] freq = new int[101];  // declaration & instantiation
freq[70+9] = 17;  //set freq[79] to 17
int grade = JLiveRead.readInt();  //assume the value to be 1 to 100 inclusive
freq[grade] = freq[grade] + 1;
freq[grade]++;
\end{verbatim}

In the example above, the expression \texttt{freq[2]} represents an integer and can be used anywhere an \texttt{int} variable can be used.

Notation (see Sec 8.2.1 for detail)

\texttt{i..j} represents the set of integers \( i, i+1, ..., j-1, j \)
\texttt{a[i..j]} represents the cells \( a[i], a[i+1], ..., a[j-1], a[j] \)
Pattern for processing an array

```java
// assume an array has been created and is referred to by variable arr
for (i=0; i<arr.length; i++) {
    // perform some process (on arr[i])
}
```

Example

Create an array of length 6 filled with random numbers in the range of 0 to 5. Calculate the sum.

Generating random numbers

`Math.random()` generates a `double` value in the range of `[0,1)`. How do you generate a random number (`double`) in some specific range?

<table>
<thead>
<tr>
<th>Expression</th>
<th>Operation</th>
<th>Range of the resulting double value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math.random()</td>
<td>Generate a random value in [0,1)</td>
<td>[0,1)</td>
</tr>
<tr>
<td>Math.random() * w</td>
<td>Scale by a factor of w</td>
<td>[0, w)</td>
</tr>
<tr>
<td>Math.random() + b</td>
<td>Shift by base value b</td>
<td>[b, 1+b)</td>
</tr>
<tr>
<td>Math.random() * w + b</td>
<td>Scale and shift a random number</td>
<td>[b, w+b)</td>
</tr>
</tbody>
</table>

So how do you generate a random number in the range a to b?

How do you generate a random `integer` in a specific range? Consider the scaling operation carefully—scale by the number of possible values in the range you want. For example, when you generate a random integer in 0..3, the number of possible values is four, not three. Finally, you must cast the `double` value into an `int`. Notice the placement of the parenthesis around the expression for generating a random number below:

```
(int) (Math.random() * 4) + 0
```

How do you generate a random integer in the range a..b?