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: ***type*`[]` *identifier*`;`**
  Examples:
```
int[] counts;
double[] price;
String[] names;
Interval[] series;  // assuming an Interval class has been defined
```
- Instantiation syntax: **`new` *type*`[` *size* `]`**
  size is an integer expression
  Example:
```
new int[4]
```
- Declartion and instantiation
```
int limit = 4;
double[] price;               // declaration
price = new double[limit];    // instantiation and assignment
```
- Creating an array using an initializer list

The size of an array is held in the constant **`length`**. **`length`** is *automatically* defined when an array is created and *cannot be changed*. In the above example, the expression **`price.length`** gives the size of the array **`price`**.

Index operator `[]`

The expression ***identifier*`[`*integer_expression*`]`** accesses an element in the array referred to by ***identifier***

Examples:
```
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]++;
```

In the example above, the expression **`freq[2]`** represents an integer and can be used anywhere an **`int`** variable can be used.

Notation (see Sec 8.2.1 for detail)

`i..j` represents the set of integers **i**, **i+1**, …, **j-1**, **j**
`a[i..j]` represents the cells **`a[i]`**, **`a[i+1]`**, …, **`a[j-1]`**, **`a[j]`**

Pattern for processing an array

```
// 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?

Expression	*Operation*	*Range of the resulting `double` value*
`Math.random()`	Generate a random value in [0,1)	[0,1)
`Math.random()*w`	Scale by a factor of w	[0, w)
`Math.random()+b`	Shift by base value b	[b, 1+b)
`Math.random()*w + b`	Scale and shift a random number	[b, w+b)

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?