Exercises: Week 3

Introductory Programming 2020

Exercise 7.1

Explore the *weblog-analyzer* project by creating a `LogAnalyzer` object and calling its `analyzeHourlyData` method. Follow that with a call to its `printHourlyCounts` method, which will print the results of the analysis. Which are the busiest times of the day?

Exercise 7.2

Write a declaration for an array variable `people` that could be used to refer to an array of `Person` objects.

Exercise 7.3

Write a declaration for an array variable `vacant` that could be used to refer to an array of boolean values.

Exercise 7.4

Read through the `LogAnalyzer` class and identify all the places where the `hourCounts` variable is used. At this stage, do not worry about what all the uses mean, as they will be explained in the following sections. Note how often a pair of square brackets is used with the variable.

Exercise 7.5

What is wrong with the following array declarations? Correct them.

```
[]int counts;
boolean[5000] occupied;
```

Exercise 7.6

Given the following variable declarations,
```
double[] readings;
String[] urls;
TicketMachine[] machines;
```
write assignments that accomplish the following tasks: (a) Make the `readings` variable refer to an array that is able to hold 60 `double` values; (b) Make the `urls` variable refer to an array that is able to hold 90 `String` objects; (c) Make the `machines` variable refer to an array that is able to hold 5 `TicketMachine` objects.

Exercise 7.7

How many String objects are created by the following declaration?
```
String[] labels = new String[20];
```

Exercise 7.8

What is wrong with the following array creation? Correct it.
```
double[] prices = new double(50);
```

Exercise 7.9

Check to see what happens if the for loop's condition is incorrectly written using the <= operator in `printHourlyCounts`:
```
for(int hour = 0; hour <= hourCounts.length; hour++)
```

Exercise 7.24

Open the *automaton-v1* project and create an `AutomatonController` object. A line containing a single * should be output in the terminal window, representing the initial state of the automaton. Call the `step` method a few times to see how the state progresses. Then try the `run` method.

Exercise 7.28

Rewrite the two if-else statements in the loop of the `update` method of the `Automaton` class of *automaton-v1* so that the assignments to `left` and `right` use conditional

operators.

Exercise 7.37

Open the *brain* project, create an `Environment` object and use the GUI controls to create a random initial setup for the automaton. Then either single-step it or run/pause it to obtain a feel for how it behaves (Figure 7.3, page 245).