COP2800C Module 8 Practice Exercise

In this practice exercise we will incorporate some statistical functions in our BoxFactory application.

Summary statistics are used to summarize a set of observations in order to communicate a large amount of information in a simple way. Summary statistics frequently include measures such as the mean (average) of the data, the minimum and maximum, and standard deviation (the amount of variation, or distance of the values from the mean).

The Java Math API provides some of these functions (min and max, for example) but in order to calculate the other values we either need to depend on an external library (e.g. the Apache Commons Math library https://commons.apache.org/proper/commons-math/userguide/stat.html) or "roll our own". For this exercise we will do the latter.

Let's start by declaring an interface which describes the methods we need. This will be stored in the file StatisticsCalculator.java, in a new sub-package named "statistics".

```
package edu.fscj.cop2800c.statistics;

// interface for statistical calculations
public interface StatisticsCalculator {

    // returns max from array of Double
    public Double max(Double[] inArray) throws ArrayIndexOutOfBoundsException;

    // returns mean from array of Double
    public Double mean(Double[] inArray) throws ArrayIndexOutOfBoundsException;;

    public Double stddev(Double[] inArray) throws ArrayIndexOutOfBoundsException;
}
```

Each method accepts a Double array parameter. Each method also tests that the length of the passed array parameter is not of length 0, and throws an ArrayIndexOutOfBoundsException if this test fails.

The BoxFactory class is modified to implement both the TableFormatter (from Module 7) and the new StatisticsCalculator interface:

```
package edu.fscj.cop2800c.container;

import java.util.Arrays;
import edu.fscj.cop2800c.table.TableFormatter;
import edu.fscj.cop2800c.statistics.StatisticsCalculator;
import java.lang.ArrayIndexOutOfBoundsException;

public class BoxFactory implements TableFormatter, StatisticsCalculator {
```

Since the class implements the StatisticsCalculator interface, it must also implement the methods. To calculate the mean, just add the array elements and divide by the length of the array. Notice that because we are using the Double wrapper type for our data, we can use foreach loops instead of the standard index-based array traversal.

```
    // stats methods to implement StatisticalCalculator interface
    // find the max value in the array
    public Double max(Double[] inArray) {
        if (inArray.length == 0)
            throw new ArrayIndexOutOfBoundsException("error: 0-length array");

        Double max = 0.0;
        for (Double d : inArray) {
            if (max < d)
                max = d;
        }
        return max;
    }

    // find the mean
    public Double mean(Double[] inArray) {

        if (inArray.length == 0)
            throw new ArrayIndexOutOfBoundsException("error: 0-length array");

        Double total = 0.0;
        for (Double d : inArray)
            total += d;

        // divide by the array length to get the mean value, store in total
        total /= inArray.length;

        return total;
    }
```

The standard deviation method is a little tougher, let's look at it more closely. Here's the formula using the fun math notation that we all know and love (full disclosure: I had to look this up, I don't have it memorized):

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation

N = the size of the population

x_i = each value from the population

μ = the population mean

In plain English, this translates roughly to:

```
standard deviation = the square root of (
                                           the sum of (
                                                  (each sample minus the mean) squared)
                                                   divided by the population size)
```

For our purposes, each sample corresponds to one array element and the population size is our array length.

```
    // find the standard deviation
    public Double stddev(Double[] inArray) {

        if (inArray.length == 0)
            throw new ArrayIndexOutOfBoundsException("error: 0-length array");

        // call our mean method to get the mean
        Double theMean = mean(inArray);
        // calculate the deviation from the mean for each element by subtracting
        // the mean, then square it and store it in this variable
        Double devFromMeanSq = 0.0;
        // accumulator to add each processed value
        Double total = 0.0;

        // loop through each element, subtract the mean and square it
        for (Double d: inArray) {
            devFromMeanSq =
                Math.pow((d - theMean), 2);
            total += devFromMeanSq;
        }

        // divide by the population size and get the square root
        total /= inArray.length;
        total = Math.sqrt(total);

        // done!
        return total;
    }
```

Finally, let's create a method to collect all of our summary stats and display them:

```
    public void showSummaryStats() {
        System.out.println("Ready to print summary statistics!");
```

```java
        // extract arrays for relevant data
        Double[] lengths = new Double[boxes.length];
        Double[] widths = new Double[boxes.length];
        Double[] heights = new Double[boxes.length];

        int count = 0;
        for (Box b : boxes) {
            lengths[count] = b.getLength();
            widths[count] = b.getWidth();
            heights[count] = b.getHeight();
            count++;
        }

        System.out.printf("length max: %4.2f\n", max(lengths));
        System.out.printf("length mean: %4.2f\n", mean(lengths));
        System.out.printf("length stdev: %4.2f\n", stddev(lengths));

        System.out.printf("width max: %4.2f\n", max(widths));
        System.out.printf("width mean: %4.2f\n", mean(widths));
        System.out.printf("width stdev: %4.2f\n", stddev(widths));

        System.out.printf("height max: %4.2f\n", max(heights));
        System.out.printf("height mean: %4.2f\n", mean(heights));
        System.out.printf("height stdev: %4.2f\n", stddev(heights));
    }
```

I call this method at the end of the BoxFactory's main method:

```java
        // show the statistics
        boxFact.showSummaryStats();
```