

MAT 2170: Laboratory 13

Key Concepts

1. `ArrayList`: creating, filling, processing
2. using the `Temperature` class

Exercises

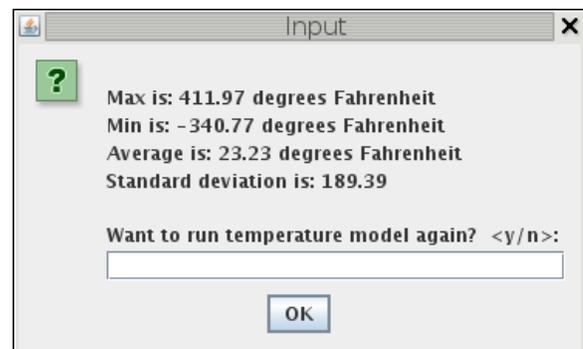
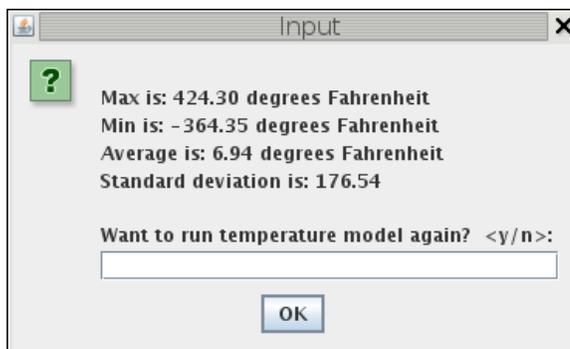
In a lab13 directory:

1. Write a program, `ArrayListStats`, which:
 - (a) Prompts for and gets the number of integers in the users' list
 - (b) Gets the list entries from the user and stores them in an `ArrayList`
 - (c) Determines the maximum value in the list
 - (d) Determines the minimum value in the list
 - (e) Determines the average, μ , of the values in the list
 - (f) Determines the standard deviation, σ , of the list values, given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\mu - x_i)^2}{n}}$$

where the x_i are the list elements, n is the length of the list, and $\sum_{i=1}^n (\mu - x_i)^2$ is the sum of the squares of the average minus each list element.

- (g) Displays all of the calculated values (displayed to two decimal places) in a single dialog window.
 - (h) Repeats until user enters a list length less than 1.
 - (i) Thanks the user for using the program and notes that it has finished running.
2. Write a program, `TemperatureStats`, which fills an `ArrayList` of 100 `Temperature` class objects with random (but correct) values, then supplies the same statistics as above — maximum, minimum, average, and standard deviation. The program should repeat as long as the user wishes. The degree values should range from -225.0 to 225.0 , and the scale equally apt to be 'C' or 'F'. Get the double value first, then the scale character based on a boolean, with `true` \rightarrow 'C' and `false` \rightarrow 'F'.
 - Set the random generator seed to 1, i.e., `rgen.setSeed(1)`. The execution should then produce first the dialog box on the left, then the one on the right (when user chooses to continue execution). Note that temperatures were converted to Fahrenheit to compare, record the max and min, and to average them.



Notes

- Create the project `TemperatureStats` in your `lab13` folder, add the `acmLibrary` jar file and `myLibrary` project to it. Create an empty java file named `TemperatureStats` in the default package, and import `util.*` as you did last week.
- You will need both `RandomGenerator` and `DecimalFormat` objects. Format output so decimal numbers are displayed with two digits after the decimal point.
- Validate the user's input — force them to enter either 'y', 'Y', 'n', or 'N' before continuing or halting. You are to use one or more methods to achieve this. Do *not* use an infinite loop with `break` statement.
- To create an `ArrayList` capable of storing `Temperature` objects, use:

```
ArrayList<Temperature> MyList = new ArrayList<Temperature>();
```
- To append a new `Temperature` to the end of a list, first determine the random values and store in, say, `val` and `degreeType`. Then, make an assignment such as:

```
MyList.add(new Temperature(val, degreeType));
```
- To assign a new `Temperature` to the i^{th} item in a list:

```
MyList.set(i, new Temperature(val, degreeType));
```
- To access the i^{th} `Temperature` list item's value in Fahrenheit, one could use something like:

```
MyList.get(i).getDegreesF()
```

Finishing Up

1. When you have completed both exercises, submit and publish. Remember to update `myLibrary` in your `www` folder if you modified anything in the `Temperature` class. Submit `myLibrary` as well as `lab13`.

End Notes

1. Use dialog, rather than console program.
2. The programs should be **modular** — i.e., broken down into logical sections and **methods**.
3. Plan ahead. Design first, then implement.