Software Configuration and Execution Guide

Media Temple Coding Demo (processingHostValues.py)

Programmed by: Sunny (Sunoh) Yoo

Date: January 11, 2017

Overview

To run the software successfully, you need:

- processingHostValues.py: python application to parse and process input data
- CodingDemoData.txt: input text file containing host names and their values

System Requirements

This chapter includes the following topics:

Operating System Requirements

Hardware and Software Requirements

Operating System Requirements

- This software was tested on Mac OSX Yosemite.

Hardware and Software Requirements

- This software needs at least 2GB RAM.
- This software was tested on Python 2.7.x or higher version.
- This software can run with an input text file (CodingDemoData.txt)

Software Architecture

processingHostValues.py is a software to parse a text file and write/print out average, maximum, and minimum values for each host in a sorted order.

Parsing Input Data: In order to parse each row in the input file, I used a delimiter **'|'** to separate a part containing hostname and a value part. Then, I parsed each part with a delimiter **','**. The first element in the first part is the host name, and all elements within the second part are values for the host. To process the values for each host, I ignored 'None' values and processed only number values.

Main function gets arguments for input file and an optional mode, creates ProcessInputFile instance with an input file name, and calls methods to create the list and print out the nodes of the list.

Figure 1 shows the elements in each core class and their relationships.

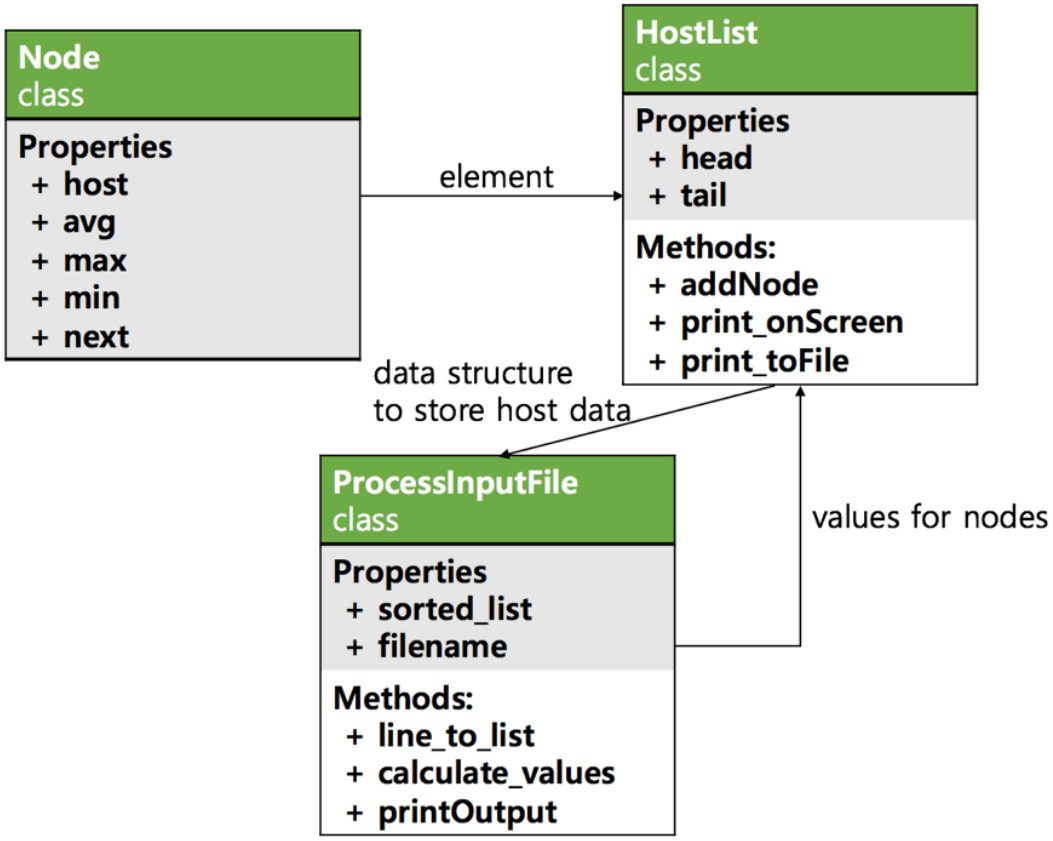

Fig. 1. This diagram demonstrate of relationships between classes within the application

Node is a node of a list having host, average, maximum, minimum, and next Node as member variables.

HostList is a linked list data structure to store each Node for each host. Each node is stored in a sorted order from maximum to minimum average values. It has head and tail nodes as member variables.

- addNode: method to add each node in a sorted order into the linked list.
- print_onScreen: method to print out each node stored in the list.
- print_toFile: method to write out each node stored in the list.

ProcessInputFile is a mainly functioning class to parse an input text file, create a HostList instance, and print out the nodes from the list. It has sorted_list and an input filename as its member variables.

- line_to_list: method to parse input text file for each line to retrieve data.
- calculate_values: method to get average, max, and min values for each host.
- printOutput: method to print out the result or write out to file if needed.

Running Application

After installing Python and basic Python configurations,

1. Get into the directory containing the processingHostValues.py using terminal.

2. To get a result only printed on the screen, type

python processingHostValues.py [input text file]

3. To get a result printed on the screen and written out to a text file (CodingDemoOutput.txt), type

python processingHostValues.py [input text file] f

Software Demonstration

```
Sunohui-MacBook-Pro:~ sunohyoo$ python processingHostValues.py /Users/sunohyoo/D
ownloads/CodingDemoData.txt
n32: Average: 92.5 Max: 100.0 Min: 73.0
n29: Average: 87.2 Max: 100.0 Min: 46.0
n30: Average: 85.8 Max: 100.0 Min: 44.0
n24: Average: 84.4 Max: 100.0 Min: 49.0
n23: Average: 83.3 Max: 100.0 Min: 31.0
n14: Average: 78.7 Max: 100.0 Min: 32.0
n34: Average: 75.0 Max: 100.0 Min: 36.0
n31: Average: 74.3 Max: 99.0 Min: 44.0
n16: Average: 73.1 Max: 100.0 Min: 33.0
n11: Average: 72.9 Max: 100.0 Min: 35.0
n33: Average: 72.9 Max: 100.0 Min: 29.0
n10: Average: 72.6 Max: 100.0 Min: 28.0
n27: Average: 68.8 Max: 100.0 Min: 36.0
n26: Average: 66.3 Max: 97.0 Min: 29.0
n12: Average: 61.6 Max: 100.0 Min: 20.0
n15: Average: 59.8 Max: 88.0 Min: 15.0
n17: Average: 59.2 Max: 97.0 Min: 30.0
n13: Average: 53.4 Max: 86.0 Min: 18.0
```

Fig. 2. This demonstrates the screen showing each host in a sorted order with average

Figure 2 shows the result without ‘f’ mode and Figure 3 shows the result with ‘f’ mode to write out the result in a text file.

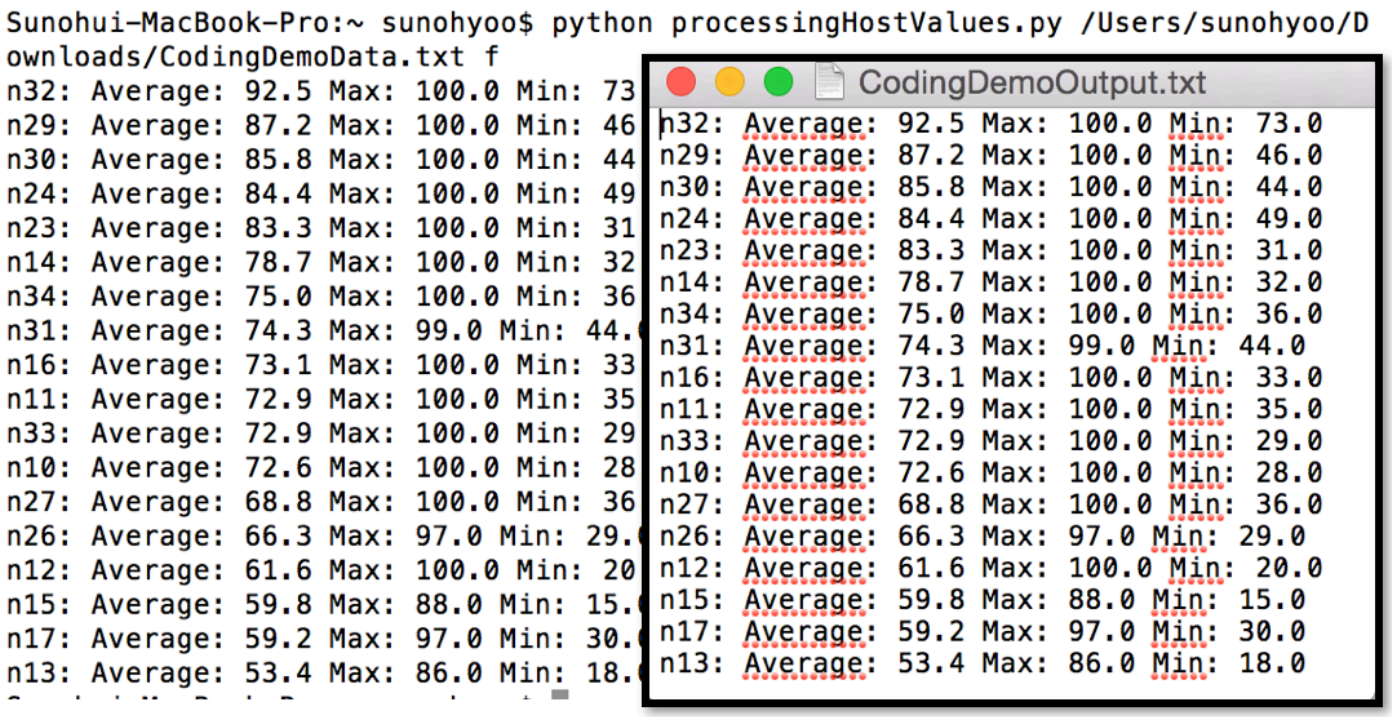

Fig. 3. This shows the printed result and created text file if run with the mode ‘f’