



Assignment 6 – Fall 2019

In this assignment we will work with Java's Lambdas and Streams as an alternative to traditional procedural programming. None of the required work should use a *FOR*, *WHILE*, or *DO ... WHILE* loops; you can only use functional programming reductions and intermediate operations. For this assignment, we will reuse code from a previous assignment; however, all required classes are given to you in separate JAVA files on Blackboard. You should only modify the required sections.

Note: please do your own work, sharing and/or copying code and/or solution ideas with/from others will result in a grade of 0 and disciplinary actions for all involved parties. If you run into any problems and have done your best to solve them, please see me before/after class or e-mail me.

Problem Description:

Change the implementation of all LOOP occurrences in the Java files as described next. The class diagram for this assignment is shown in Figure 1; and the output of the *current code* (non-functional programming) is shown in Figure 2 and Figure 3. Your changes should produce the *SAME* output. After making the required changes, I strongly recommend that you compare the number of lines in the new code with the old one; this will highlight some of the advantages of functional programming.

Class *LogEntry*:

- ✓ No changes are required. However, note that this classes uses pattern matching instead. Please review the new implementation as a self-study exercise.

Class *A3Driver*:

- ✓ **Static block:** modify this method to use Java Streams. You should not use a loop structure.

Hints:

- ✓ See slide 22 of Chapter 17

Class *NetworkLogManager*:

- ✓ **loadFile:** modify this method to use Java Streams. You should not rely on *Scanner* or *while*.

Hints:

- ✓ See slide 182 of Chapter 17
- ✓ *forEach* is useful here. Process every line as before.

- ✓ **searchByRange:** modify this method to use Java Streams. You should not use a loop structure.

Hints:

- ✓ Every instance of *List* has a *streams()* method. e.g. print all the values in a list:
`myList.streams().forEach(x-> System.out.println(x));`
- ✓ *filter*, *map* and *collect* are useful here – See slides 121 of Chapter 17

- ✓ **searchBy:** modify this method to user Java Streams. You should not use a loop structure.

Grades:

Item	Points
A3Driver	
Static block	10
Class NetworkLogManager	
loadFile	25
searchByRange	25
searchBy	20
Correct output	20
	100

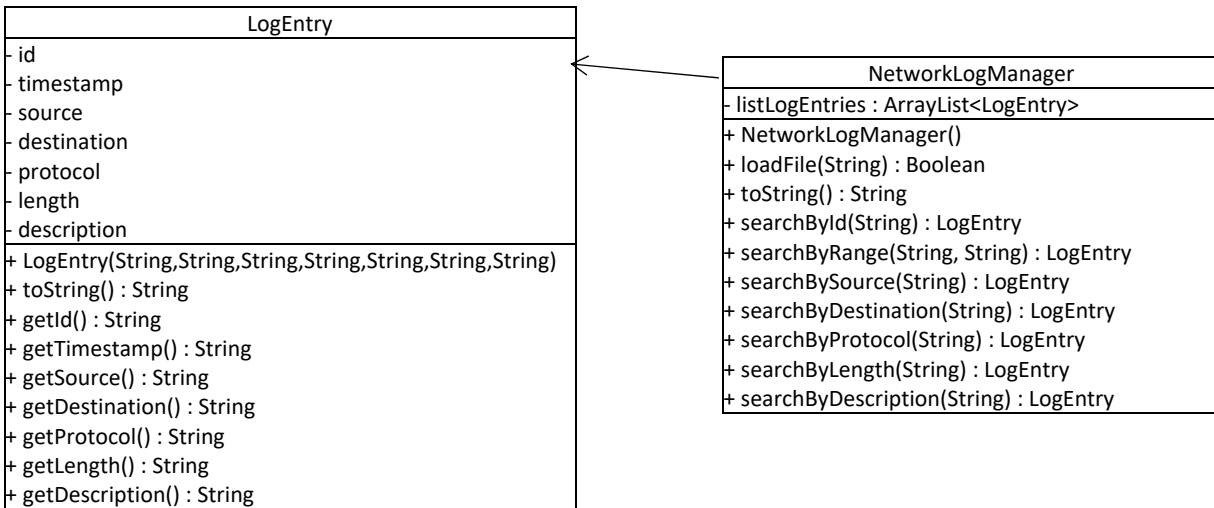


Figure 1: Class diagram.

Skipping line: #,Time Stamp,Source,Destination,Protocol,Length,Description
 Skipping line: ,Oct 13 2019 05:09:42,192.168.5.138,192.229.189.15,-,491,communication with play.google.com
 Skipping line: ,Mar 01 2019 07:46:18,192.168.5.102,,55.163.53,TCP,174,TCP communication with microsoft.com
 Skipping line: ,Oct 12 2019 10:29:46,192.168.5.140,g208.82.237.226,TCP,1049,TCP communication with 192.168.5.148
 Skipping line: 11,Feb 03 2018 01:02:52,192.168.5.216,192.168.1.1,-,1269,communication with ebay.com
 Skipping line: 12,Mar 09 2018 23:40:30,192.168.5.242,31.13.66.35,-,0,communication with linkedin.com
 Skipping line: 19,Jul 16 2018 00:57:45,192.168.5.203,a.org208.80.154.224,-,424,communication with reddit.com
 Skipping line: ,Aug 06 2018 19:16:00,192.168.5.111,104.244.42.193,TCP,502,TCP communication with youtube.com
 Skipping line: ,Aug 09 2018 12:13:19,192.168.5.195,51.101.66.49,TCP,1037,TCP communication with youtube.com
 Skipping line: 26,Jul 18 2019 01:43:11,192.168.5.114,104.244.42.193,-,324,communication with reddit.com
 Skipping line: 30,Jan 03 2018 15:20:11,192.168.5.113,192.229.189.15,-,1288,communication with google.com
 Skipping line: 31,Mar 26 2019 20:58:02,192.168.5.140,35.201.95.83,-,1199,communication with merriam-webster.com
 Skipping line: 32,Aug 21 2019 15:03:59,192.168.5.239,.101.1.224,TCP,114,TCP communication with 192.168.5.244
 Skipping line: 35,Nov 28 2018 01:02:23,192.168.5.123,31.13.66.35,-,399,communication with espn.com
 Skipping line: ,Dec 12 2019 09:55:34,192.168.5.137,.101.1.224,TCP,1106,TCP communication with 192.168.1.1
 Skipping line: 37,Oct 10 2019 23:58:54,192.168.5.220,.101.1.224,UDP,171,UDP communication with tripadvisor.com
 Skipping line: 38,Apr 15 2019 15:58:59,192.168.5.167,a.org208.80.154.224,-,575,communication with nytimes.com
 Skipping line: 39,Aug 24 2019 18:48:55,192.168.5.228,4.16.66.82,-,596,communication with tripadvisor.com
 Skipping line: 41,Feb 13 2018 20:11:20,192.168.5.153,05.251.242.103,-,505,communication with forbes.com
 Skipping line: 43,Apr 04 2019 18:51:21,192.168.5.140,.101.1.224,-,55,communication with homedepot.com
 Skipping line: ,Jun 15 2019 13:18:45,192.168.5.127,192.168.1.1,TCP,683,TCP communication with 192.168.5.168
 Skipping line: 52,Feb 16 2018 14:07:52,192.168.5.238,.101.1.224,TCP,388,TCP communication with 192.168.5.168
 Skipping line: 55,Dec 21 2019 22:39:06,192.168.5.249,a.org208.80.154.224,-,0,communication with 192.168.5.177
 Skipping line: ,Nov 09 2019 10:45:26,192.168.5.213,172.217.11.14,TCP,0,TCP communication with imdb.com
 Skipping line: 59,Apr 08 2019 03:05:17,192.168.5.111,172.217.11.14,-,972,communication with espn.com
 Skipping line: 60,May 27 2019 14:05:31,192.168.5.174,192.168.5.197,-,932,communication with linkedin.com
 Skipping line: 61,Nov 13 2019 17:03:35,192.168.5.157,192.168.5.177,-,904,communication with linkedin.com
 Skipping line: 62,May 04 2019 10:38:36,192.168.5.214,34.194.103.209,-,0,communication with pinterest.com
 Skipping line: 67,Nov 18 2019 20:06:27,192.168.5.141,,55.163.53,UDP,1369,UDP communication with 192.168.5.148
 Skipping line: ,Jul 04 2019 00:45:36,192.168.5.163,172.217.10.238,TCP,1428,TCP communication with pinterest.com

Figure 2: first 30 skipped Records

 NetworkLogManager: there are 371 valid records

There are 0 records with id 1
 There are 1 records with id 9
 There are 286 records with protocol TCP
 There are 85 records with protocol UDP

There are 146 entries from 2018

Figure 3: output of test class