CSC-240 Lesson 10: Strings, Characters and Regular Expressions

Assignment 10: Implementing a Simple Data Mining Program

The file, `Text.java` (in this zipfile), is the source code for a class named **Text** (in the package `edu.frontrange.csc240.text`) that exports just one **static** constant, *TEXT*, which is a **String** containing text as described below. (Hence the fully qualified name of this constant is `edu.frontrange.csc240.text.Text.TEXT`—sorry about the lack of imagination in the naming.)

Please implement the program as described below. When finished you will submit the following to the Dropbox:

- `DataMiner.java`—the class definition that contains the *main* entry point, implementing the program described below. You may submit as many additional classes (as the corresponding `.java` files) as are needed to appropriately decompose the solution into independent parts. (You should not submit the supplied Java class file `edu.frontrange.csc240.text.Text.TEXT`.)

This program is an exercise in scanning text for patterns of interest using just regular expressions (as implemented by the **Pattern** class). The text that has been supplied has been downloaded (some time ago) from the Front Range Community College web site, and formatted into a **String** to make access more convenient.

A program like this is performing “data mining”. That is the term for searching sources of data, such as web pages, looking for items of interest to research or business. In this exercise, a **String** will be read that was copied from a web page to see if that web page is likely to have course scheduling information. Pages that refers to courses and that have a lot of dates and phone numbers are likely to be of interest for this purpose.

Specifically, for the given **String** `Text.TEXT` in `Text.java`, do the following:

- Count the total number of input lines;
- Count the number of references to courses using formats similar to “aaa-ddd”, where “a” represents a letter and “d” represents any digit, such as “CSC-240”.
  Keep in mind that the “**-**” is sometimes carelessly omitted, or is replaced by a space, and the letters which might be in uppercase may not be, and that letters and numbers may occur next to each other is other contexts;
- Count the number of references to phone numbers in formats similar to but not necessarily the same as “ddd-ddd-dddd”. Remain aware that there are other reasonable and possible formats for telephone numbers, since people write telephone numbers in several different ways—some of which may contain spaces;
- Count the number of dates in formats similar to but not necessarily the same as “mm/dd/yyyy”. Again, there are as well other very reasonable ways of writing dates or in other possible formats for dates that might be written on a web-site. It is worthwhile to spend some time thinking about the many different ways in which dates are written; and,
- Find, count and also print a list of any URLs referring to web pages. Only the references to web pages are required (but ***not*** to other Internet locations that are unlikely to be web pages, such as those accessed by schemes such as `gopher`, `ftp`, `ymsgr`, `xmpp`, *etc.*).

  Web pages are those accessed with the scheme known as hypertext transfer protocol (`http`), or the secure version of that scheme (`https`), or (by Internet convention) have “`www`” as the name of their lowest level sub-domain. Try to print as much of the URL as possible (not just the server address).

Make just one search **Pattern** for each of the above items. Avoid making multiple searches for any one item--regular expressions have enough expressive power to not require more than one pattern (and one search) for each of the requested items.

It is not necessary to copy the TEXT constant. Just importing it as a **static** value is sufficient.

The following *About the Assignment* topic contains further explanation, directions, and hints about this assignment. Please read that topic before starting this assignment.

Important

Please follow the directions of Programming Assignment Identification in submitting your programming solutions.

If you have any questions or concerns about the Assignment or the *About the Assignment* topic, please use the Lesson Question discussion topic, or send a message by the D2L Internal Messaging system if you prefer.

Note: you may submit to the Assignment Submission Folders as many times and as often as you wish, up to the deadline time. Each submission is tagged with the date/time, and so each submission remains separate and distinct. Unless you leave instructions to the contrary, only the most recent of each file with the same name will be viewed for the purposes of grading. Details may be found in the topic *How To Submit and Get Feedback on Assignments* in the *How To* module.

Messages that accompany Assignment Submissions are read, and responded to, ***only*** when assignment submissions are graded (which is after the Assignment Submission Folder closing date/time). If you have a comment or question about an assignment, or a request for assistance, that needs an earlier response, then that comment, question or request should be made or asked *via* an Internal Message or the Discussion board, as these are usually read and answered every day.