Assignment4: Hyperlinks Integrity Checker for Web Documents

Objectives:

Upon completion of this assignment you will be able to:

1- Apply the object-oriented Analysis and Design concepts studied in class
2- Communicate with web servers using HTTP and URL Connections in Java
3- Use XML Parsing to extract information from web documents
4- Incorporate threads in your program to increase execution time efficiency
5- Study and analyze program execution time performance
6- Develop a simple GUI to input and display application data.

Hyperlinks Integrity Checker:

Description

Automatic Hyperlink Validation for web documents and websites is an essential task for website and servers' administrators. It helps detect broken links, corrupt files and also is very useful in tracking changes that happen to websites over time, which are all too difficult to be done manually, especially with the rapid growth of websites and online document directories.

A typical checker that performs such task takes two input parameters; **first**, it takes the **URL of the document** at which it should start the check, considers this document as the root, and recursively checks all the hyperlinks in this document, it leads to. The **second input** parameter is the **cut-off threshold**, which is essential to prevent indefinite execution of the program.

Tasks

- Implement a simple program that takes the inputs mentioned in the description and does the required check.
- Your program should be efficient in terms of design and data structures used.
- One cut-off threshold should be defined by the user, the **level threshold** which informs the program when to stop in terms of link depth. The program should be terminated when the threshold is met.
- Your application should be **multi-threaded** i.e. it should use threads to run multiple tasks in parallel to increase efficiency and without giving erroneous output or repetition overhead.
- Links to different file types should be supported, i.e. links should be checked whether they lead to other HTML pages or other files.
- You only have to check HTML links with no consideration to scripts, buttons or other types of links.

Deliveries & Notes:

- ✓ You should write the program using **java language**.
- ✓ You should work in group of two.
- ✓ Your code should be **clean**, **readable** and **commented**.
- ✓ You should deliver a **report**, contains description of your implementation and **algorithms** you have implemented.
- ✓ You should deliver charts for each case how you selected the number of threads in your program to find the optimal solution. The x-axis is the number of threads and the y-axis is the running time of the program.
- ✓ You should deliver a **UML** diagram of your program (user case, class, activity and sequence diagrams).
- ✓ **Late submission** is accepted and is graded out of **50%.**
- ✓ Delivering a copy will be awfully penalized for both parties, so delivering nothing is so much better than delivering a copy.

References:

- **Soup library**
  - https://jsoup.org/download
  - https://jsoup.org/cookbook/extracting-data/selector-syntax
- **HttpURLConnections**
  - http://download.oracle.com/javase/1.4.2/docs/api/java/net/HttpURLConnection.html
- **Threads**
  - http://download.oracle.com/javase/1.4.2/docs/api/java/lang/Thread.html
  - http://download.oracle.com/javase/tutorial/essential/concurrency

Good Luck

Prof.Dr. Layla abo Hadeed
Dr. Mohamed kholief
Eng.Rania Ismail
Eng.Ahmed Shokry
Eng.Hagar Nassar
Eng.Salma Mostafa
Eng.Khaled Ismail
Eng. Ahmed Elsayed
Eng. Noha Mahmoud