Bài tập thực hành:

Crawler với crawler4j

Thời gian làm bài: 1 tuần (Xem dealine trong link nộp bài trên moodle)

Nộp bài:

- Nộp bài lên moodle.
- Đặt tên bài nộp theo định dạng MSSV.rar. Trong đó bao gồm:
  - Tập tin báo cáo.
  - Các tập tin dữ liệu theo yêu cầu của bài tập
- Nếu sử dụng code trên Internet, trước đoạn code sử dụng phải có chú thích đường dẫn đến trang web chứa đoạn code. Trường hợp 2 bài có đoạn code giống nhau, bài không chú thích đường dẫn đến trang web chứa source code sẽ bị tính là gian lận, và bị 0 điểm.

Các hành vi sử dụng toàn bộ/một phần bài làm của người khác sẽ bị 0 điểm cho toàn bộ phần thực hành

1 Crawler4j

Download tại: http://www.java2s.com/Code/Jar/c/crawler4j.htm

2 Crawler class

Để sử dụng crawler4j, đầu tiên ta sẽ viết một class kế thừa từ lớp WebCrawler để thực hiện tác vụ tìm kiếm.

```
public class MyCrawler extends WebCrawler {

}
```

Ở đây, ta sẽ cài đặt 2 hàm là **shouldVisit** và **visit**. Hàm **shouldVisit** là hàm kiểm tra xem đường dẫn trang web có thỏa điều kiện mà ta đặt ra hay không. Nếu trang web thỏa các điều kiện ta đặt ra, hàm visit sẽ được gọi để xử lý nội dung của trang web này.

Ta khai báo trường FILTERS làm bộ lọc các tập tin mà ta quan tâm:

```
private final static Pattern FILTERS = Pattern.compile(
                        ".*(\\.(css|js|gif|jpg"  + "|png|mp3|mp3|zip|gz))$");
```

Ta thực hiện hàm **shouldVisit** như sau:

```
/**
* This method receives two parameters. The first parameter is the page
* in which we have discovered this new url and the second parameter is
* the new url. You should implement this function to specify whether
* the given url should be crawled or not (based on your crawling logic).
* In this example, we are instructing the crawler to ignore urls that
* have css, js, git, ... extensions and to only accept urls that start
* with "http://www.ics.uci.edu/". In this case, we didn't need the
* referringPage parameter to make the decision.
*/
@Override
public boolean shouldVisit(Page referringPage, WebURL url) {
    String href = url.getURL().toLowerCase();
    return !FILTERS.matcher(href).matches()
           && href.startsWith("http://www.ics.uci.edu/");
}
```

Tiếp theo, ta chỉ định vị trí lưu các kết quả xử lý được:

```
public static String storageFolder = "F:/Teaching/crawler/root/";
```

Cuối cùng, ta sẽ viết hàm visit để xử lý các trang web:

```
@Override
public void visit(Page page) {
    String url = page.getWebURL().getURL();
    System.out.println("URL: " + url);
    if (page.getParseData() instanceof HtmlParseData) {
        HtmlParseData htmlParseData = (HtmlParseData) page.getParseData();
        String text = htmlParseData.getText();
        String html = htmlParseData.getHtml();
        Set<WebURL> links = htmlParseData.getOutgoingUrls();

        System.out.println("Text length: " + text.length());
        System.out.println("Html length: " + html.length());
        System.out.println("Number of outgoing links: " + links.size());
    }
}
```

3 Thiết lập ở hàm main

```
String crawlStorageFolder = "F:/Teaching/crawler/root";
int numberOfCrawlers = 7;
int MAX_DEPTH = 7;

CrawlConfig config = new CrawlConfig();
config.setCrawlStorageFolder(crawlStorageFolder);
```

```
config.setMaxDepthOfCrawling(MAX_DEPTH);

PageFetcher pageFetcher = new PageFetcher(config);
RobotstxtConfig robotstxtConfig = new RobotstxtConfig();
RobotstxtServer robotstxtServer = new RobotstxtServer(robotstxtConfig,
                                                      pageFetcher);

CrawlController controller = new CrawlController(config, pageFetcher,
                                                 robotstxtServer);
```

4 Chạy crawler

```
String seed = "vnexpress";
controller.addSeed("http://" + seed + ".net/");

MyCrawler.storageFolder = "F:/Teaching/crawler/root/" + seed + "/";
controller.start(MyCrawler.class, numberOfCrawlers);
```

5 Bài tập

Mỗi lần duyệt qua một trang web:

- ghi vào thư mục lưu trữ (storageFolder) một tập tin .txt với tên là Title của trang web và nội dung là những nội dung text có trong trang web.
- ghi vào thư mục lưu trữ (storageFolder) một tập tin .html với tên là Title của trang web và nội dung là nội dung trang web.

Nộp lại báo cáo, mã nguồn và thư mục chứa kết quả thực thi.

Lưu ý: Để ghi nội dung tập tin dưới dạng Unicode, ta có thể sử dụng:

```
BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(new
                                       FileOutputStream(fileName),"UTF-8"));
```

Tài liệu tham khảo:

[1] https://github.com/yasserg/crawler4j