ExecutorService and the Callable Interface

General part

- Explain the benefits from using a Thread Pool
- Explains ways to handle "returned values" from Threads

Practical part

The file; *urls.txt* on Fronter contains a small code-snippet that sets up an array with URLs pointing to the main page of all CA-2 Front-pages.

Getting Started:
Create a Web-project as the starting point for this exercise. Initially you don't need any web-functionality so just add your code in a relevant package in the source folder. For part 3-5 you need the "web-functionality".

Implement the following features, using the code-snippet in urls.txt

1) As you hopefully know, all CA-2 Front-pages should include the following IDs:

```
<div id="authors"> Peter Hansen, Ole Jensen, Ida Hansen </div>
<div id="class">A or B or COS</div>
<div id= "group"> Group number </div>
```

   - Use an `ExecutorService` and the `Callable` interface to implement a solution that can connect to all URLs and scrape the author, class and group information from all the pages[1].
   - The solution must take advantage of a multi-core System.
   - Initially just print the values to the console
2) Create a class `Group` that models the information provided via the three divs, and rewrite your code to build a List<Group> with information about all groups.
3) Implement a REST-service: GET `api/group` which should return a JSON representation of all the Scraped information in your List.
4) The REST call above is very costly (in terms of server resources). Since data are not likely to change very often, come up with a simple solution (2-3 lines of code) that will cache the result and reuse this cached value after the first call.
5) If the service we implemented in step-3+4 were to be used for a real checks of which groups have an online CA-2 page, we cannot cache the value forever. Change the code above to refresh the cache after a given amount of time (say 1hour).

[1] **Jsoup** will make this a very simple task: http://jsoup.org/ (See example below)

```
Document doc = Jsoup.connect("http://catwo-2ndsemester.rhcloud.com/CA2/ ").get();
Elements authors = doc.select("#authors");
String authors = authors.text();
```

Note: this part is NOT a part of the exercise, it's meant as FYI and this section will not be included with the real question.

You might feel that this was a silly exercise, invented by your teachers only for the sake of the exercise.

Screen Scraping or Web Scraping (https://en.wikipedia.org/wiki/Web_scraping) is actually very common thing to do for many real life scenarios.

If you don't feel that this semester has provided you with enough challenges consider this scenario.

You would like to have live data for your "momondo-exercise" but how do you get it?