APIT - Lab3 - Threading

Simon Rogers

Aims and objectives

The point of this lab is to give you some practice in making `Thread` or `Runnable` objects. Your task is to write a program that can find the maximum and minimum values of a large matrix by splitting it into smaller 1-dimensional arrays. The `MakeData.java` class (download from Moodle) will generate random matrices for you. To generate `m` rows and `n` columns, use `Double[][] a = MakeData.generateRandomData(m,n)`. You may also find it useful that you can pass one row of an array by just using one index. I.e. say you have `Double[][] a = new Double[100][20]` and a method is expecting a 1-Dimension array (i.e. `Double[] b`) then you can pass `a[4]` say (this is a reference to the 5th row which is just a 1-dimensional array).

It might be helpful to do things in this order:

- It might be useful to note down the different classes you need. There are several:
  - A class with a main to actually run things
  - A class that extends `Thread` (or implements `Runnable`) that loops through a 1-dimensional array. It needs a `search` method that prints out the maximum and minimum.
- Keep testing as you go along...
- To start with, just have each thread print their minimum and maximum values.
- Now think about how you might collate the minimum and maximum values in `main` to get the global minimum and maximum. We'll cover two approaches in the lectures:
  - Have all threads manipulate a shared object that stores global min and max
  - Use `join` so that `main` waits for them all to finish and then query each for their min / max
- When you have a complete solution, write simple code to just search through the matrix in one go. Do the mimimum and maximum values always agree? Do you think they should?