Assignment 1 (2nd half of semester)

Object-Oriented Programming III (CT3535), Year 2017/2018, Semester 1

- ***Submission deadline (strict): Thursday, 2nd November, 23:59.*** *No late submission without medical certificate!*
- *Submission via Blackboard / Turnitin. Please note that all submissions will be checked for plagiarism.*
- *You need to put your Java source code for all questions in this assignment into a single PDF document which you then upload.*
- *Also insert into this document screenshots of any output your program(s) generate.*
- **Use Java comments to explain your code.** Missing or insufficient comments may lead to mark deductions.

Question [max. 100 marks]

Create a Java class with the following features:

- There should be a static field `infiniList` with type `ArrayList<String>` (an - initially empty - global list whose elements are words).

- The class should have a `main()`-method which starts at about the same time three *in parallel* running threads.

  Each of these three threads should perform the same task (simultaneously with the other threads): each thread should repeatedly add words to `infiniList` (using an infinite loop), according to the following five rules:

  1) In each loop iteration, the thread should add exactly one word to `infiniList`.
  2) If the last word in `infiniList` is currently "This", the thread should append word "is" to `infiniList`.
  3) If the last word in `infiniList` is currently "is", the thread should append word "infinite" to `infiniList`.
  4) If the last word in `infiniList` is currently "infinite", or if `infiniList` is still empty, the thread should append word "This" to `infiniList`.
  5) At any time, `infiniList` should contain "This" only at the beginning of the list or directly after an occurrence of "infinite", "is" should occur in the list only directly after a "This", and an "infinite" should occur only directly after an "is". No other words are allowed in `infiniList`.

  Example: after running the program for a short while, `infiniList` should contain the following list of words: "This", "is", "infinite", "This", "is", "infinite", "This", "is", "infinite", "This", "is", ...

  Use thread synchronization to ensure that rule 5 above is always observed. Keep in mind that all three threads operate in parallel on the same global (i.e., shared…) list `infiniList`.

  Furthermore, each time a word has been added to `infiniList`, the updated content of `infiniList` should be printed.

  Start your program, let it run for a few seconds and create a screenshot. Submit the screenshot in addition to your program code.