Task 6

Labyrinth with threads

Task text

This task is very similar to the previous one where you were supposed to find exits in labyrinth(maze). In task 5 you found the exits using recursion. In this task, you will use threads so that your program can find all exits faster if your machine has multiple cores or processors.

When your program task 5 came to a white route where there were several possible paths ahead, your program examined one and one path in sequence.

Now use threads to explore all the paths further in parallel. You get a lot of freedom to do this as you like, the only thing we require from your program is that one possible path further should be examined by the same thread (called the old thread) that came to the route, while the other possible paths should be further explored of their own new thread. Make sure you start new threads before the old thread moves on to its next white route. What happens if the old thread first moves on to the next route and then starts new threads? Make a comment in your program that answers this question.

In the same way as in task 5, the exits (String) are to be added to a list. This must now be protected as a monitor so that multiple threads do not get into each other's legs when they add the exits at the same time.

Finally, the program will retrieve and print the exits from the list. It is only necessary to find exits in acyclic labyrinth (mazes).

Implementation

Note that your application must wait to print the list of all the exits until all the threads have gone through the entire labyrinth and found all the solutions. There are many ways to do this. One way is to call the join()-method in the threads that the old thread starts up. Then the old thread will know that all the threads it has started are terminated. Another way is to keep track on how many new threads have started in total and how many are terminated. When the number of new threads still alive is down to zero, all solutions are found and can be printed.

Requirements for submission

The requirements for submission are the same as in task 5, but in addition to the Java files, you must attach a screenshot of your program run on Labyrinth 3 (in the file “3.in” from task 5) from the starting point (5.3).