Dr. Wolfgang Emmerich
Dept. of Computer Science
University College London
Pearson Building Room 402

30-Okt-98

C340 Concurrency

Work Sheet

This lab session will give you practical experience with critical sections and mutual exclusion in Java. You will also revisit the Java thread life cycle.

For that purpose we want to implement the FSP model of the museum admission control system that we introduced in tutorial session 1. That means that we want the museum director, the entrance turnstile, the exit turnstile and the controller to be separate threads.

The director uses an applet that has two buttons to open and close the museum. Furthermore it displays the number of visitors. If the museum entrance is open the applet shows the number of visitors who have entered in blue colour. It changes the colour of this number to red if the entrance is closed. Likewise, the number of visitors who left the building is displayed in blue if the exit is open and it is displayed in red if the exit is closed. The number of visitors in the museum is shown in black.

Exercise 1:
Design classes and their relationships that are needed for the implementation of the museum admission control system. Draw a UML class diagram to document your design. (Hint: opening and closing of the entrance and exit turnstiles can be implemented as starting and stopping of threads)

Exercise 2:
What are the critical sections in your implementation? How do you synchronise them?

Exercise 3:
Implement the design in Java.