

Chapter 5 - exercises

- 5.1 A single slot buffer may be modeled by:

$$\text{ONEBUF} = (\text{put} \rightarrow \text{get} \rightarrow \text{ONEBUF}) .$$

Program a Java class `OneBuf` that implements this one slot buffer as a monitor.

- 5.2 Replace the condition synchronization in your implementation of the one slot buffer by using semaphores. Given that Java defines assignment to scalar types (with the exception of long and double) and references types to be atomic, does your revised implementation require the use of the monitor's mutual exclusion lock?
- 5.3 In the museum example (assessed coursework), identify which of the processes, `EAST`, `WEST`, `CONTROL` and `DIRECTOR`, should be threads and which should be monitors. Provide an implementation of the monitor(s).
- 5.4 FSP allows multiple processes to synchronize on a single action. A set of processes with the action `sync` in their alphabets must all perform this action before any of them can proceed. Implement a monitor called `Barrier` in Java with a method `sync` that ensures that all of N threads must call `sync` before any of them can proceed.
- 5.5 *The Savings Account Problem:* A savings account is shared by several people. Each person may deposit or withdraw funds from the account subject to the constraint that the balance of the account must never become negative. Develop a model for the problem and from the model derive a Java implementation of a monitor for the savings account.