Programming in Java

Describe briefly the facilities provided in Java for synchronising concurrent threads. [6 marks]

An alternative scheme would be to model the system used in some shops where a machine issues numbered tickets to customers, and customers are served in numeric order. A ticket machine holds an integer, initially zero, and has a single atomic operation:

\[ \text{turn()} \quad \text{increments the integer and returns its previous value} \]

A scheduler also holds an integer, initially zero, and has two atomic operations:

\[ \text{next()} \quad \text{increments the integer count} \]
\[ \text{queue(value)} \quad \text{suspends the calling thread until the count is at least as large as the value given as an argument} \]

Given a ticket machine, \( m \), and a scheduler, \( s \), a critical region could then be coded as follows:

\[
\text{number} = m.\text{turn}(); \\
\text{s.queue(number);} \\
. \\
. \quad \text{protected code} \\
. \\
\text{s.next();}
\]

Write Java classes \texttt{TicketMachine}, with a \texttt{turn} method, and \texttt{Scheduler}, with \texttt{next} and \texttt{queue} methods. [8 marks]

Show how a synchronised buffer holding a single value could be implemented using this new scheme. [6 marks]