Marcel ZAUDER 16-124-836
Pascal GERIG 16-104-721

7.1 Several Questions

a) *When should you consider using asynchronous invocations?*

b) *In what sense can a direct invocation be asynchronous?*

c) *What is an "early reply"?*

d) *What are "futures"?*

e) *When are futures better than early replies?*

7.2 Several Questions 2

a) *Why does the call Thread.currentThread().join() not make much sense in a thread's concurrent run method?*

b) *Why can you encounter an IllegalMonitorStateException when calling notifyAll() in a code block that is not synchronized?*

c) *What happens with the thread when the code execution gets to the end of the thread's run method (suppose no loop is involved in the run method)?*
After a thread reached the end of its run method it will terminate and will not be in the state of a Runnable anymore.

d) *Why do some Java apps not terminate, even though their GUI has been closed, and how can you mitigate that problem?*
If a thread that has a loop in its run method it will never reach the end of it, so it will never terminate. Another cause can be that a thread is still waiting on something and was never notified. Because the thread is still in the Runnable state the Java app will not terminate. In order to mitigate this we can implement a stop method, which is called when the GUI is called which will terminate all threads that are still running, like interrupting them or "forcefully kill" them (the first one is the better option).

e) *Name one reason for using a while loop in a thread's run method. Justify your selected reason.*

7.3 Snow Flakes