

Colloquium G - Concurrent Programming

Simon Hasir – 7006072

June 19, 2022

1 TE-1

- (a) Yes, if java is executed with the *-ea* flag which enables Assertions and the Exception is not handled.
- (b) No, this behavior would involve too much overhead.
- (c) No, not in the Bytecode
- (d) No, comments dont have any semantic ;)
- (e) Yes, *await* just continues if it gets notified.
- (f) No, Linked Listed have to be monitors aswell.
- (g) Not really, *synchronized* can be applied almost as flexible, the only problem is that only one lock per object is used.
- (h) Yes
- (i) Yes, because multiple locks allow for more concurrency and potentially less lock blocking
- (j) Yes, they are sequential in the *synchronized* block.
- (k) Not always, only if you can ensure that *notify* is called after *wait*.
- (l) Yes, safety violations can be found in finite executions and liveness problem only appear in infinite executions.
- (m) No, see the above
- (n) No, *run()* is a sequential function which you can execute concurrently with *Thread.start()*

- (o) Yes
- (p) Yes, *wait()* stops execution.
- (q) Yes, *notifyAll()* would be correct.
- (r) No, you can only make methods and blocks synchronized.
- (s) What is incorrect? But the signal call may valid, but the functionality may be impaired.
- (t) Yes.
- (u) No, it's a correctness property (finite).
- (v) False, liveness
- (w) No, it only sets the boolean flag *interrupted* (fetchable with *isInterrupted()*)
- (x) Yes

2 TG-2

1. Monitor *pseuCo* -> explicit locks Java

- (a) Replace *monitor X* in *pseuCo* with *class X* in Java.
- (b) Add a *private final* instance-wise lock *l* in Java.
- (c) Replace condition *c* with *b* in *pseuCo* with *private final c = l.newCondition()* in Java.
- (d) Add lock at beginning of every function (wrapped in a try) and an unlock in the final block.
- (e) Replace *waitForCondition(c)* in *pseuCo* with *while (!b) c.await();* in Java

2. Explicit locks Java -> Implicit Locks Java

- (a) remove explicit lock
- (b) Replace locking and unlocking with the *synchronized* keyword in the function definition.
- (c) Replace *c.await* with *wait()*;
- (d) Replace *signal[All]* with *notify[All]*.
- (e) Add some kind of indication which was condition should be the one that was triggered.
- (f) Enjoy :)

3 TG-3

3