COMP 346: Assignment 2 - Programming

Tsang Chi Kit - Student ID: 25692636

6th March, 2016

Task 1

Codes are grouped under "task1" package.

- An integer "stack access counter" variable is declared at line 42, in BlackStack.java.
- The code that report the final counter value when a thread is terminate, is at line 124, in BlackManager.java.

Task 2

0.1 Task 2.1

See line 29-39, at BlockStack.java, task2 package.

0.2 Task 2.2

Original code already is handling the case where push() is used on an empty stack. See line 210 in BlackManager.java. The if-clause from line 221-224 handle the case when the stack is NOT empty, and line 232 handle when it is empty.

0.3 Task 2.3

Added user defined exception to the following methods in BlockStack.java, task2 package.

- pick()
- getAt(int)
- push(char)
- pop()

Task 3

- Initialized Semaphore mutex object at line 36, BlockManager.java, task3 package.
- Added mutex.P() before all phase1(), and a finally clause to release the permit before phase2().

Task 4

From line 96-105 (BlockManager.java - task4 package), we can see there are 10 threads in total. Therefore, we need 10 blocks. My implementation is using CountDownLatch, with 10 counts. Latch will await right before phase2() start. This way, no thread will start computation until the countdown reach zero, which is exactly when all phase1() finishes.

Task 5

We can use turnTestAndSet() and a binary Semaphore to check if the thread is in the right order. If the thread is not, then it will release its permit, break the loop, and await for next turn. When it is at its correct turn, then it will execute phase2(), after it is done, it will release the permit for other threads.