

Lab 8

Part I: Binary semaphore in Java

A binary semaphore java example is given on blackboard, BinarySemaphore.java. Read, understand the code, and run it.

Part II: Producer and consumer problem in Java using Semaphore

Change the code of the producer-and-consumer java implementation using a lock and condition variables (given in Lab 7) into using semaphore.

Make sure your code can work when there are multiple producers and consumers, eg., for the following main():

```
public static void main(String[] args) {  
    // Object on which producer and consumer thread will operate  
    ProducerConsumerImpl sharedObject = new ProducerConsumerImpl();  
    // creating producer and consumer threads  
    Producer p1 = new Producer(sharedObject);  
    Producer p2 = new Producer(sharedObject);  
    Consumer c1 = new Consumer(sharedObject);  
    Consumer c2 = new Consumer(sharedObject);  
    // starting producer and consumer threads  
    p1.start();  
    c1.start();  
    p2.start();  
    c2.start();  
}
```

Part III: Build a thread-safe data structure using Semaphore

Change your BoundedIntStack.java solution in Lab 7 into using semaphore, i.e., build the concurrency control inside the methods of BoundedIntStack class, so that whatever code using a BoundedIntStack instance does not need to worry about concurrency control, which has been provided by the BoundedIntStack class itself.

To compare, the previous Part III shows an example where the buffer data structure (queue) is not thread safe, and concurrency control is enforced outside the data structure (queue).