THBSAB001

CSC3002F OS PART 2 ASSIGNMENT - CONCURRENCY

ENFORCING PANDEMIC LOCKDOWN LEVEL 3 RULES

- **Which threads run in the program?**
  CounterDisplay
  Customer
  Inspector
  ShopView
  SocialDistancingShop - Parent thread

- **Which classes are shared amongst threads?**
  CustomerLocation
  GridBlock
  PeopleCounter
  ShopGrid

- **Explain the synchronization mechanisms you added to each class and why they were appropriate.**
  GridBlock - added available semaphore initialized to one, to only allow one customer per block as per level 3 regulation, semaphore released when customer leaves a block.

  PeopleCounter - made all integer variables atomic to ensure only one thread at a time can increment or decrement the counter variables. This is to protect against data races.
  Added maxAllowed semaphore initialized to max number of people allowed inside the store as per level 3 regulation, this semaphore is released each time a customer leaves the store.

  ShopGrid - added inEntrance semaphore initialized to one, to allow only one customer at the entrance door, this is released once a customer leaves the entrance gridblock.

- **How did you ensure liveness in the code?**
  The program already has liveness through the use of threaded classes, the only left task is to insure program runs as per level 3 regulations and that shared classes do not have data races or deadlock

- **How did you protect against deadlock? Was this necessary?**
  To protect against deadlock, one has to ensure there is no circular wait and that the classes only require one lock to complete a task so it doesn't have to wait on another lock.
  This is necessary because deadlock leads an unusable program ie freezing