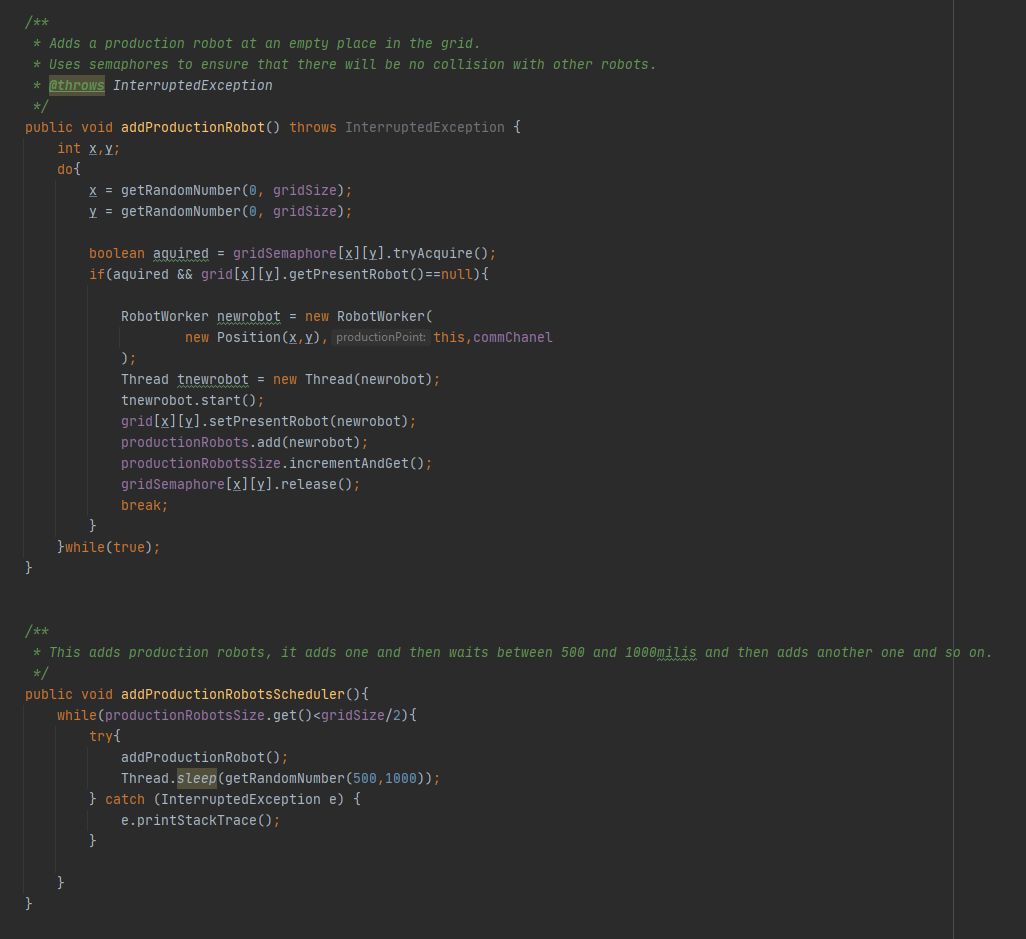
**HOW IT WORKS:** There are many parts that work together in order to solve the problem of production and delivery of vaccines.

In order to solve the production problem, each production facility has its own grid made up of squares and on each position there is a semaphore. The semaphores are used to make sure that no 2 or multiple threads access or modify the same square at the same time. This proves to be beneficial to preventing collisions.

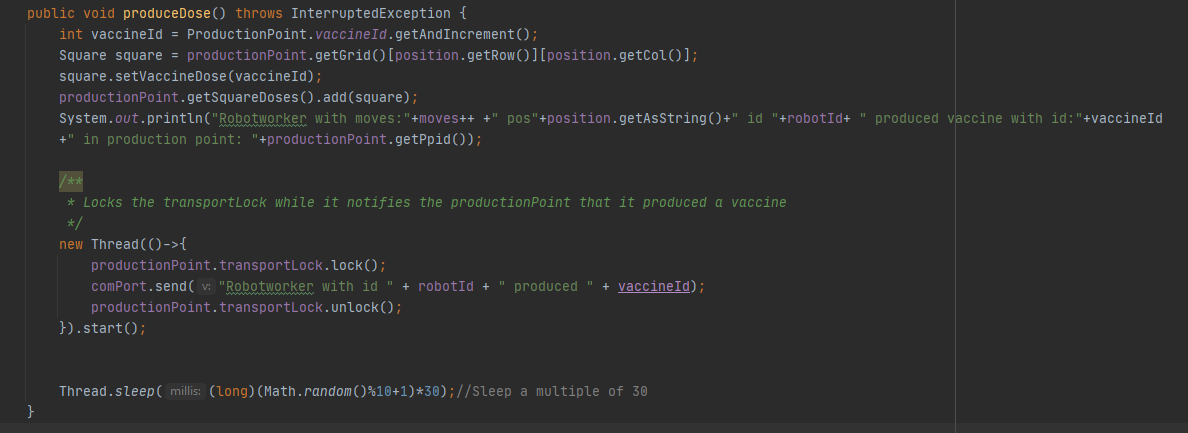
The production robots are produced by each production point at an interval of 500-1000ms. The production point makes sure that there is no other robot present in the square and nor is moving to that position by acquiring the semaphore, produces the thread that runs the new robot and adds that robot to that square.



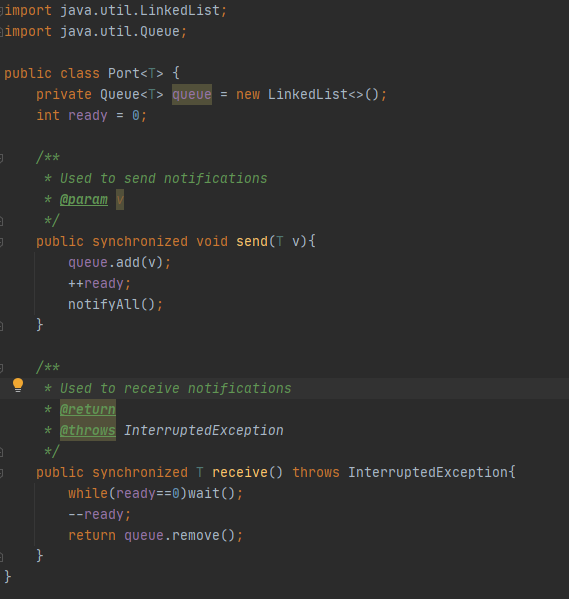
Each production robot moves by its own and tries to get on first available square that is not occupied by another robot or that is not a wall. After it moves, it produces a vaccine in the new square then rests for a multiple of 30ms.



After the does is produced, a new thread is started, using a lambda function, that communicates with the production point that it produced a vaccine. During this communication, it locks the transport lock so that the transport does not take the vaccine before the communication ended.

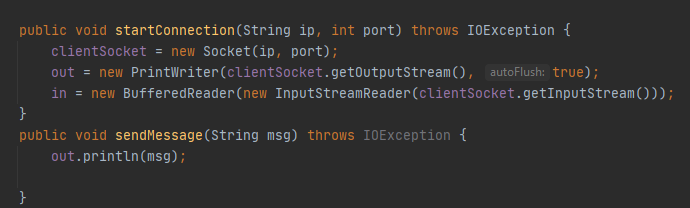


The Port class is used for that communication



In order to deliver the vaccines from the production points to the headquarter so that they can be distributed around the world, we use packaging robots, called TransportWorker that are managed and held by a TransportCommander.

Each transport worker uses TCP/IP to deliver the vaccines to the headquarter, making a connection on “127.0.0.1:5587”.



startConnection function uses a socket and BufferedReaders and PrintWriters in order to communicate with the Headquarter.



The headquarters receive vaccine doses from the transport workers by using a thread that has a lambda inside

