**Aufgabe 12.1 – Implementierung eines Warenlagers**

Schreiben Sie ein Programm, welches ein Warenlager simuliert. Das Warenlager muss mindestens aus den folgenden Klassen bestehen:

* Eine Klasse die das Lager selbst Repräsentiert und Methoden zum ein- und auslagern bereitstellt.
* Eine Klasse die einen Anlieferer repräsentiert. Soll als eigener Thread laufen
* Eine Klasse die einen Abholer repräsentiert. Soll als eigener Thread laufen.
* Eine Klasse die lediglich das Lager und die Threads erzeugt/startet

Ihr Programm muss folgende Punkte erfüllen:

* Das Lager hat nur eine bestimmte Kapazität
* Wenn das Lager leer ist, kann nichts abgeholt werden. Der Abholer muss warten bis etwas eingetroffen ist.
* Wenn das Lager voll ist, kann nichts angeliefert werden. Der Lieferant muss warten bis etwas abgeholt wurde.
* Auf das Lager kann nur wechselseitig zugegriffen werden
* Jedes Teil im Lager hat eine zufällig erzeugte ID
* Es soll 3 Liefer-, 3 Abhol- und den Hauptthread geben
* Ein Lieferthread fügt dem Lager unendlich neue Teile hinzu und wartet nach jeder Lieferungen eine Zufällige Zeit + eine min. Lieferzeit-Konstante
* Ein Abholthread nimmt unendlich Teile aus dem Lager und wartet nach jeder Abholung eine Zufällige Zeit + eine min. Abholzeit-Konstante
* Jede Aktion wird in der Konsole ausgegeben
* Verwenden Sie wo möglich geeignete Konstrukte (Vererbung, Interfaces, etc.)

Implementieren Sie den wechselseitigen Zugriff auf zwei Varianten: Locks und synchronized