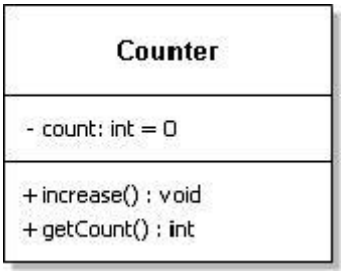
Aufgabe 12.1 – Multithreaded Counter

Schreiben Sie ein Programm, welches mit 100 Threads gleichzeitig bis 1000 zählt. Verwenden Sie hierfür die Klasse aus untenstehendem Klassendiagramm. Zur Implementierung der Zugriffsbeschränkung auf die gemeinsame Variable nehmen Sie die Variante 1 und Variante 2.

Die Methode „`increase()`“ soll zusätzlich zu dem Hochzählen der Variable „`count`“ den Zählerstand und den Thread, der die Methode aufgerufen hat, ausgeben. Eine mögliche Zeile hierfür ist z.B.

```
System.out.println(Thread.currentThread().getName() + "- " + this.count);
```



Aufgabe 12.1 (Variante 1)

Um die „Zugangskontrolle“ umzusetzen verwenden Sie bitte das aus der Vorlesung bekannte **Monitor-Konzept**. (Stichwort **synchronized**)

Aufgabe 12.1 (Variante 2)

Die Zugangskontrolle kann auch mit Hilfe von „**atomic variables**“ umgesetzt werden.

Aufgabe 12.2 – 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

Aufgabe 12.3 – Lucas-Lehmer-Test Multithreaded

Bauen Sie den Lucas-Lehmer-Test aus Aufgabe 11.3 so um, dass ein Bereich an Zahlen getestet wird. Jede Zahl soll dabei in einem eigenen Thread getestet werden. Arbeiten Sie mit so vielen Threads wie ihr Rechner Prozessorkerne besitzt (Wird im Gerätemanager unter Prozessoren angezeigt).