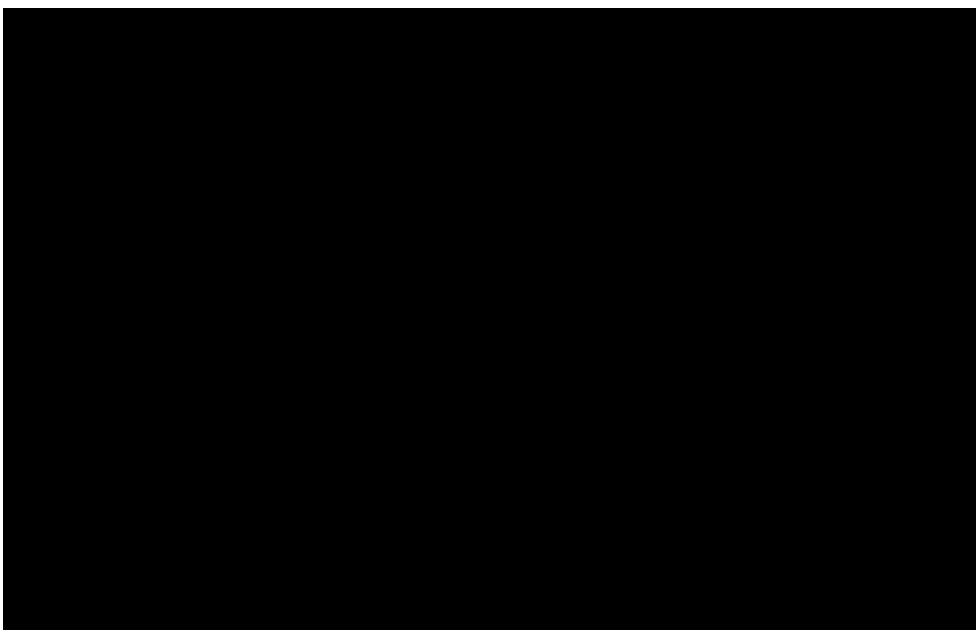


# † bung Supermarkt (Producer/Consumer)

Es soll das Producer - Consumer - Pattern anhand einer Supermarkt-Simulation angewendet werden. Dabei gilt folgender Ablauf:

1. Kunden betreten das Geschäft (werden vom `KundenGen` erzeugt)
2. Die Kunden kaufen ein (stellt ein zufälliges delay dar)
3. Die Kunden stellen sich an der Kasse (=Consumer) an und bezahlen dort, bevor sie das Geschäft wieder verlassen.

Eine Lösung könnte in etwa wie folgt aussehen:



Generell:

- ¥ Als Queue ist hier eine `LinkedList` einzusetzen. Dabei ist zu beachten, dass `LinkedList` nicht thread-safe ist und deshalb eine entsprechende Synchronisation zu berücksichtigen ist.
- ¥ Verwenden Sie die Methode `offer(o)` zum Einfügen, `poll()` zum Abholen des nächsten Elements.

Für die Klasse `Kunde` gilt:

- ¥ Jeder Kunde soll eine einmalige, fortlaufende ID erhalten.
- ¥ Der Einkaufswert soll zufällig zwischen 0 und 100 Euro festgelegt werden.
- ¥ Neue Kunden sollen über eine Factory mit dem Namen `einkaufen` erstellt werden. Der Konstruktor wird `private`. In dieser Factory soll dann Kundennummer und Warenwert ausgegeben werden.

F r die Klasse **Kasse** gilt:

  F r jede Kasse ist der Saldo (also die Summe an Einnahmen) mitzuz hlen

  Nachdem der Kasse signalisiert wurde, dass sie schliessen soll, muss sie noch die bereits angestellten Kunden abfertigen

Versuchen Sie, die Zeiten zu variieren so dass ein Stau an der Kasse entsteht. F gen Sie dann mehrere parallel laufende Kassen hinzu!