# Spätkauf-Verwaltung Schickmabitte

Die *Schick&Kannix GmbH* hat einer umfangreichen Marktanalyse festgestellt, dass die Berliner Späti-Szene noch mit Technik aus der Steinzeit arbeitet: Bestellungen werden oft noch per Telefon weitergegeben, und es passieren ständig Fehllieferungen. Als Lösung hat das Marketing-Team die Software *SchickMaBitte* konzipiert. Damit kann der moderne Spätkauf seine Waren elektronisch direkt beim passenden Lieferanten bestellen und montags bis sonntags zu jeder Tageszeit die wichtigen Artikel für die Nachbarschaft vorrätig halten.

Leider wurde die Software nicht mehr fertig, und nun ist *S&K* bekanntlich pleite. Der Insolvenzverwalter will die Idee noch nicht ganz abschreiben und bittet Sie um Ihre fachliche Analyse als Gutachter.

## Lernaufgabe 🕙 30 Min.

1. **Analysieren Sie die vorhandenen Klassen und die Implementierung der Vererbung.**  
   Orientieren Sie sich für Ihre Beobachtungen dazu folgenden Leitfragen:
2. Wie wird Vererbung in Java-Quelltexten gekennzeichnet?
3. Welche Klasse erbt im Projekt *SchickMaBitte* von wem (→ Tabelle)?
4. Wie kann ein Konstruktor auf die Oberklasse zugreifen,   
   wie können Attribute der Oberklasse gesetzt oder gelesen werden?
5. Wie werden Assoziationen vererbt?

|  |  |  |
| --- | --- | --- |
| **Klasse** | **Oberklasse** | **UnterklasseN** |
| Lieferant |  | LieferantEU |
| LieferantEU | Lieferant |  |
| Artikel |  | NonFood  Food |
| NonFood | Artikel |  |
| Food | Artikel | Getränk |
| Getränk | Food |  |

1. **Stellen Sie die Klassenhierarchie von SchickMaBitte mithilfe von Umletino als OOD-Klassendiagramm dar.**Nutzen Sie dazu Ihre Beobachtungen und die Tabelle. TIPP: Beginnen Sie mit der obersten Klasse (die Klasse ohne weitere Oberklassen), bauen Sie nach und nach die Klassenhierarchie auf.

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung