

# Aufgabenstellung

## Zum Verständnis:

Verschiedene Dateien werden täglich aus einem Drittsystem heruntergeladen und mit einer Datenbank synchronisiert.

Die Dateien werden in einem **TAB-separierten Textformat** angeliefert.

Die Datei- und Feldspezifikation kann dem Spezifikationsdokument unter **/doc/FileSpec\_1.1.pdf** entnommen werden.

Für die hier zu lösende Aufgabe geht es zunächst nur um die Datei **/sample/Substances.dat** (Spezifikation: Kap. 3.3.2 auf Seite 4 f).

Die Datei enthält eine Liste chemischer Reinstoffe (Substance). Jeder chemische Reinstoff enthält eine Reihe von internationalisierten Reinstoffbezeichnungen (Synonym).

z.B.

R	1061	7439-92-1		231-100-4	1	0	1	0	0	0
RN	1061	0	DE	Blei						
RN	1061	0	EN	Lead						

Darstellung in UML:



## Aufgabe:

Bitte entwickeln Sie eine Klassenbibliothek, die die Datei **/sample/Substances.dat** einliest und als Java-Objekte für die weitere Verarbeitung zur Verfügung stellt.

Folgende Anforderungen sind zu beachten:

- Für jede Zeile sind die Anzahl der enthaltenen Attribute, sowie die Länge der einzelnen Attribute lt. Spezifikationsdokument zu prüfen.
- Weitere Dateitypen (z.B. die beiliegende **/sample/Companies.dat**) sowie daraus resultierende Objekte sollen sich mit möglichst geringem Eingriff in den bestehenden Code bei dessen maximaler Wiederverwendung ergänzen lassen.
- Da die Dateien nahezu beliebig lang werden können wird darauf Wert gelegt, dass das Einlesen blockweise – also bei der Substances.dat je Reinstoff – erfolgt, also nicht die ganze Datei bzw. die daraus resultierenden Objekte vor der Weiterverarbeitung in den Hauptspeicher geladen werden müssen.  
Hierbei kann man sich die Sortierung der Datei zunutze machen.

Diese Klassen sollen vom Test getrennt unter **/src/prod/** abgelegt sein.  
Die Package-Struktur kann frei bestimmt werden.

Bitte entwickeln sie einen Unit-Test der die Datei **/sample/Substances.dat** mit Ihrer Klassenbibliothek einliest und im Format

```
CAS-NR, erstes Synonym der aktuellen Locale
```

auf der Konsole ausgibt.

Beispiel:

```
118725-24-9, (1,3-Dioxo-2H-benz (de) isochinolin-2-ylpropyl) hexadecy...
```

Prüfen Sie in Ihrem Test ob die letzte Substanz entsprechend der gesetzten Locale „**TestD**“ oder „**TestE**“ heißt:

R	900081458	22222-22-2		1	-1	1	0	0	1
RN	900081458	0	DE	TestD					
RN	900081458	0	EN	TestE					

Source Folder für den Unit-Test ist **/src/test/**.

## Hinweise:

Die beiden Klassen **Dummy.java** und **DummyTest.java** sind nur vorhanden, damit die beiden Source Folder nicht leer sind und dürfen nach belieben umbenannt oder entfernt werden.

Die Sprache, bzw. die in der JVM verwendete Locale kann mit der Kommandozeilenoption

```
-Duser.language
```

festgelegt werden.