

## 2. Praktische Übung zur

**Logische und funktionale Programmierung****GRUPPENÜBUNGEN:**

PROLOG-Systeme lassen sich als relationale Datenbank-Systeme (DBSysteme) einsetzen, indem Beziehungen zwischen Daten in geeigneter Form als Prädikate formuliert und die Wiba (als Gesamtheit der Fakten) als Datenbasis aufgefasst wird.

Tabelle mit den Vertreternamen:

Vertreternummer	Vertretername	Vertreterwohntort	Vertreterprovision	Kontostand
8413	meyer	bremen	0.07	725.15
5016	meier	hamburg	0.05	200.00
1215	schulze	bremen	0.06	50.50

Tabelle mit den Artikelstammdaten:

Artikelnummer	Artikelname	Artikelpreis
12	oberhemd	39.80
22	mantel	360.00
11	oberhemd	44.20
13	hose	110.5

Tabelle mit den Umsatzdaten:

Vertreternummer	Artikelnummer	Artikelstück	Verkaufstag
8413	12	40	24
5016	22	10	24
8413	11	70	24
1215	11	20	25
5016	22	35	25
8413	13	35	24
1215	13	5	24
1215	12	10	24
8413	11	20	25

So entnehmen wir z.B. der 3. Zeile der letzten Tabelle, dass der Vertreter mit der Nummer 8413 (also der Vertreter **meyer** aus **bremen** mit der Vertreterprovision von 7% und dem Kontostand in Höhe von EUR 725.15) 70 Artikel der Nummer 11 (also **oberhemden** zum Preis von EUR 44.20) am 24. des laufenden Monats verkauft hat.

**(G 5)**Relationale Datenbanken

Gib ein PROLOG-Programm an, bei dem die oben aufgeführten Tabelleninhalte als Argumente von geeigneten Prädikaten enthalten sind!

**(G 6)**Goals

Formuliere geeignete Goals zu den folgenden Anfragen an die Datenbasis:

- a) Welcher Artikel trägt die Artikelnummer 12?
- b) Hat der Vertreter mit der Nummer 1215 am 25. des laufenden Monats einen Umsatz getätigt?
- c) Wurden vom Vertreter mit der Nummer 8413 am 24. des laufenden Monats Hosen verkauft?
- d) Gibt es Umsätze für Artikel mit der Nummer 11 oder der Nummer 13 am 25. des laufenden Monats?

### (G 7)Prädikate

Nimm ein Prädikat namens `tätigkeit` in die Wissensbasis (Wiba) auf, mit dem angefragt werden kann, ob ein bestimmter Vertreter einen bestimmten Artikel an einem bestimmten Tag des laufenden Monats verkauft hat! Ermittle mit Hilfe dieses Prädikats das Ergebnis der folgenden Anfrage:

?- `tätigkeit(meyer,hose,24)`.

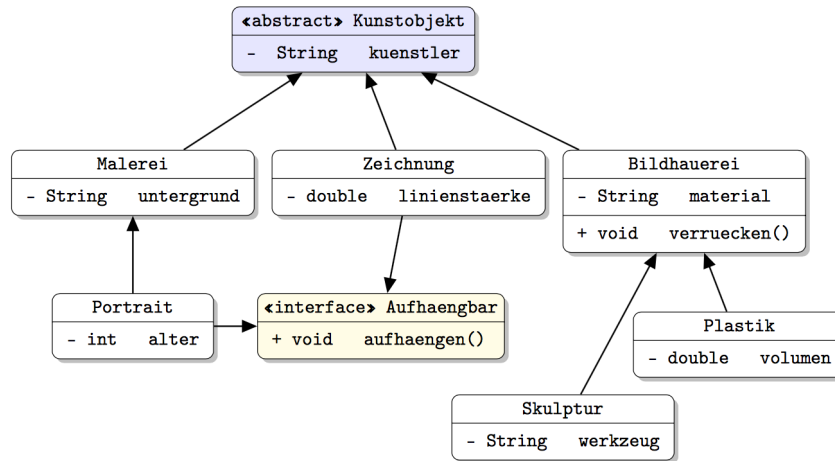
### (G 8)Abhängigkeiten in Prolog

Erfinde eine fiktive Situation einer IT Firma mit verschiedene Departments und Rollen, so dass folgende Prädikaten anwendbar sind. Natürlich musst du die entsprechende Wissensbasis definieren! Schreiben keine Prädikate außer den geforderten und nutze bei deiner Implementierung jeweils Prädikate aus den vorangegangenen Aufgabenteilen.

- a) Übertragen die Informationen aus deiner fiktiven Firma in eine Wissensbasis für Prolog. Gebe hierzu Fakten für die Prädikatssymbole `person` und `hatRang` an. Hierbei gilt `person(X)`, falls  $X$  eine Person ist und `hatRang(X, Y)`, falls  $X$  den Rang  $Y$  hat.
- b) Stelle eine Anfrage an das im ersten Aufgabenteil erstellte Programm, mit der man herausfinden kann, wer ein Junior Programmier ist.  
HINWEISE: Durch die wiederholte Eingabe von `;` nach der ersten Antwort werden alle Antworten ausgegeben.
- c) Schreibe ein Prädikat `bossVon`, womit man abfragen kann, wer innerhalb der Firmenhierarchie einen Rang direkt über dem eines anderen bekleidet.
- d) Stelle eine Anfrage, mit der man herausfinden kann, zu wem es jemand anderen gibt, der in der Firmenhierarchie genau eine Stufe darunter steht. Es soll bei jeder Antwort nur derjenige auf dem jeweils höheren Rang, nicht aber derjenige auf dem niedrigeren Rang ausgegeben werden. Mehrfache Antworten mit dem gleichen Ergebnis sind allerdings erlaubt.
- e) Schreibe nun ein Prädikat `hatGleichenRang` mit einer Regel (ohne neue Fakten), mit dem man alle Paare von Personen abfragen kann, die den gleichen Rang innerhalb der Firmenhierarchie bekleiden. Stelle sicher, dass `hatGleichenRang(X, Y)` nur dann gilt, wenn  $X$  und  $Y$  Personen sind.
- f) Schreibe schließlich ein Prädikat `vorgesetzt` mit zwei Regeln (wieder ohne neue Fakten), mit dem man alle Paare von Personen abfragen kann, sodass die erste Person in der Firmenhierarchie der zweiten Person vorgesetzt ist. Eine Person  $X$  ist einer Person  $Y$  vorgesetzt, wenn der Rang von  $X$  größer als der Rang von  $Y$  ist. Stelle auch hier sicher, dass `vorgesetzt(X, Y)` nur dann gilt, wenn  $X$  und  $Y$  Personen sind.

### (G 9)Java Klassen mit Prolog

Wir betrachten folgende Java-Klassenhierarchie



- Übertrage die Klassenhierarchie in eine Wissensbasis für Prolog. Beschränken dich hierbei auf die Namen der Klassen und des Interfaces. Verwende die zweistelligen Prädikate **extends** und **implements**, so dass **extends(A, B)** gilt wenn die Klasse *A* (direkt) die Klasse *B* erweitert (also **A extends B** im Quellcode stehen würde). Die Bedeutung von **implements** ist analog zu verstehen.
- Schreibe eine Anfrage, die alle Klassen als Antwort zurückgibt, die die Klasse **Kunstobjekt** direkt erweitern.
- Schreibe das zweistellige Prädikat **instanceof**. Der Aufruf **instanceof(A, B)** soll wahr sein, wenn
  - A* und *B* identisch sind, oder
  - A* eine Klasse bzw. Interface *M* direkt erweitert/implementiert und **instanceof(M, B)** gilt.
- Erweitere die Wissensbasis um Fakten **nichtAbstract(X)** für alle Klassen *X*, die weder abstract noch ein interface sind. Schreibe anschließend das zweistellige Prädikat **instanzMoeglich**. Der Aufruf **instanzMoeglich(A, B)** soll wahr sein, wenn eine Java-Variable vom Typ *A* auf ein Objekt der Klasse *B* verweisen kann. Berücksichtige hier, dass es zwar Variablen vom Typ einer abstrakten Klasse oder eines Interfaces geben kann, aber in dieser Variable nur Objekte von nicht-abstrakten Klassen stehen können.