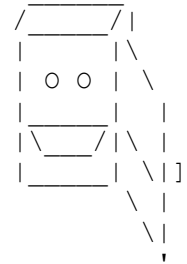


Übung 4

JavaCC: Hilfe, mein Kühlschrank spricht zu mir (15 Punkte)

Schreibe eine [JavaCC](#)-Grammatik für einen "sprechenden" Kühlschrank. Er soll auf Fragen antworten bzw. Auffüllen und Entnahme von Lebensmitteln mit Hilfe einer kleinen Grammatik für "kontrollierte Sprache" unterstützen. Der Dialog mit dem Kühlschrank besteht darin, dass auf jede über die Computertastatur erfolgte Benutzereingabe eine Reaktion mit Hilfe eines JavaCC-basierten kontextfreien Parsers generiert wird bzw. eine Fehlermeldung gegeben wird.



Benutzereingabe: 1 Zeile, fehlertolerant mit . oder ? oder ohne Satzzeichen, gefolgt von <Enter>, abgeschlossen, bereits in `FridgeParser.jj` implementiert. Numerus/Flexion ignorieren.

Kühlschrank-Antwort: vollständiger Satz, der die Frage beantwortet, oder sonstige sinnvolle Antwort wie Einkaufsliste, z.T. bereits in `FridgeParser.jj` implementiert, Numerus/Flexion ignorieren.

Vervollständige die Datei `FridgeParser.jj` (aus ihr wird mit JavaCC die Laufzeitklasse `FridgeParser.java` erzeugt) und lege eine neue Klasse `FridgeEx.java` an, die von `Fridge.java` abgeleitet ist und alle notwendigen Erweiterungen enthält (alle Files in [template_ue04.rar](#)). Zur Verwendung von `FridgeEx.java` muss in `FridgeParser.jj` die Initialisierung des `fridge` Objekts entsprechend geändert werden!

`Fridge.java` und `FridgeEx.java` (Methoden z.T. schon in `Fridge.java` implementiert): Mit Hilfe der einfachen `java.util.Properties`-Datenstruktur (s. Java API Doc; im Javabuch sind nicht alle Methoden beschrieben, z.B. fehlen dort die u.g. XML-Methoden) soll der Bestand verwaltet werden, so dass mit deren Hilfe Auskunft über den aktuellen Kühlschrank-Inhalt gegeben werden kann

Die Bestandsdaten werden über die Methode `java.util.Properties.storeToXML()` in der Datei [inventory.xml](#) gespeichert (beim Beenden des Programms) und aus ihr (beim Programmstart) mit `java.util.Properties.loadFromXML()` ausgelesen. `java.util.Properties` verfügt auch über einen Iterator, mit dem man die Einkaufs- und die Bestandsliste erzeugen kann. Die Einkaufsliste ist dabei ein Spezialfall der Bestandsliste mit Produkten, deren Bestand = 0 ist. Produkte sind in der `Properties`-Datei [inventory.xml](#) vordefiniert, es sollen jedoch beliebige neue hinzugefügt werden können (s. Tokendefinition <PRODUKT> bzw. Regel Bestand).

Regeln für `FridgeParser.jj` (bitte dort vervollständigen):

Eingabe des Benutzers	Aktion und Ausgabe des Kühlschranks
(Wie spät ist Wieviel Uhr ist es)	Uhrzeit ausgeben, teilweise bereits implementiert
(Ich gebe dir Hier (ist sind)) <Anzahl> <Produkt>	Bestand für Produkt ändern und neuen Bestand für Produkt ausgeben; ggf. noch nicht im Bestand vorhandene Produkte hinzufügen
Ich entnehme [<Anzahl>] <Produkt>	Bestand für Produkt ändern und neuen Bestand für Produkt ausgeben, default für Anzahl ist 1
Haben wir noch <Produkt> Ist (ist sind) [noch] [<Anzahl>] <Produkt> da	nein oder ja + Bestand für Produkt

Java II – Wintersemester 2010/2011

Eingabe des Benutzers	Aktion und Ausgabe des Kühlschranks
Wieviel <Produkt> (ist sind) [noch] da	Bestand für Produkt ausgeben
Was ist <Expr> Wieviel ist <Expr> Berechne <Expr>	Resultat Kettenrechnung ausgeben; dazu chaincalc-Regeln aus Vorlesung einfügen und abwandeln, <LF> und ()+ aus Expr() entfernen
[Zeige] (Bestand Bestandsliste Inhalt) [<Produkt>]	Bestand für alle Produkte oder ein Produkt
Was muss ich kaufen Einkaufsliste	Liste aller Produkte mit Bestand = 0
Auf Wiedersehen Schluss Ende Aus Tschüss Beende Beenden	Beenden und Bestand speichern in inventory.xml

Alle konstanten Wörter in den Benutzereingaben müssen durch entsprechende <TOKEN>s vordefiniert werden (d.h. FridgeParser.jj um weiteren Tokendefinitionen ergänzen) Fett gedruckte Worte: Konflikt durch gemeinsame Teilregel manuell auflösen (nur notfalls in Optionen am Anfang LOOKAHEAD = 2 setzen; dann wird jedoch eine Warnung ausgegeben).

Verwende das JavaCC-Plugin für Eclipse: <http://sourceforge.net/projects/eclipse-javacc> Das zip-file [sf.eclipse.javacc-1.5.22-updatesite.zip](#) (enthält bereits javacc-5.0) runterladen, in Eclipse über das Menü Help/Software Updates/Add Site/Local... öffnen und installieren, Eclipse neu starten; JavaCC wird über das Kontextmenü mit rechter Maustaste auf der Datei FridgeParser.jj gestartet, anschließend rechte Maustaste/RunAs... auf der daraus generierten Datei FridgeParser.java zum Starten des Kühlschranks-Dialogs (Ein-/Ausgabe über das "Console"-Fenster in Eclipse).

Bitte soviel Java-Code wie möglich in die Klasse FridgeEx.java stecken, nicht in die JavaCC-Datei FridgeParser.jj. Grund: Eclipse unterstützt autocompletion und Java code syntax checks nicht im JavaCC-Modus, so dass das Code-Schreiben dort etwas mühsam ist. Manchmal gerät dort auch das Syntax-Highlighting durcheinander, dann einfach die *.jj-Datei in Eclipse schließen und wieder öffnen.