



Softwaretechnik 2002

Prof. Dr. H. Klaeren und M E Leypold

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



Copyright © 2002 M E Leypold.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 published by the Free Software Foundation; with no invariant Sections, with no Front-Cover Text, and no Back-Cover Texts.

Since the license is rather long, I have chosen not to attach the license itself to this document. If this document is distributed (i. e. during a course) with other documents with the same license notice it suffices to distribute just one separate copy of the license. If on the other side this document is distributed alone, I require that a copy of the GNU Free Documentation License, Version 1.1, be attached.

Übungsblatt 1

Spezifikation eines FiFo in VDM-SL

Ausgabe: 23.4.2002
Abgabe: 7.5.2002, 11:45
Verlängert!

Inhaltsverzeichnis

1 Spezifikation eines FiFo

2

Einleitung

Wir werden uns in den folgenden Wochen in den Hausaufgaben mit Spezifikationen und deren Implementation befassen.

Spezifikationen können recht kompliziert werden, deshalb ist es nützlich ein Tool zur Verfügung zu haben, welches – soweit das möglich ist – die Spezifikation auf Syntax und innere Konsistenz prüft. Wir haben uns für die Spezifikationssprache VDM-SL entschieden, für die das Tool *IFAD VDMTools* existiert. Die Konsistenzprüfung ist hier als eine Typprüfung realisiert, wie das ja auch bei so vielen Programmiersprachen der Fall ist.

Eure Aufgabe wird diese Woche hauptsächlich darin bestehen, Euch mit den *VDMTools* vertraut zu machen. Zu diesem Zweck stellen wir Euch eine recht einfache Aufgabe, die aber mit den *VDMTools* erledigt werden soll. Bitte lasst Euch von dem scheinbar simplen Problem nicht täuschen: Wir glauben, dass Ihr wisst, wie eine Warteschlange funktioniert, zumindest intuitiv. Die Krux an dieser Aufgabe ist lediglich, dies auch *formal* zu notieren und mit den Eigenheiten und der Bedienung der *VDMTools* vertraut zu werden.

Als Hilfe werden Euch mit diesem Übungsblatt drei weitere Handouts ausgegeben:

1. Eine Kurzanleitung zur Bedienung der bei uns installierten *VDMTools*: *VDMTools am WSI*. Hier wird erklärt, auf welchen Rechnern die Tools installiert sind, wie Ihr da herankommt, und welche Handbücher für Euch relevant sind.
2. Ein einfaches Beispiel für eine Spezifikation in VDM-SL: *Spezifikation eines Assoziativen Arrays in VDM-SL*. Ihr könnt Euch für die Lösung dieser Aufgabe an diesem Beispiel orientieren: Das Beispiel hat einen ähnlichen Schwierigkeitsgrad, die eingesetzten Mittel sollten die gleichen sein.

Aus gegebenem Anlass ist hier auch ein Wort der Warnung angebracht: Ein *Assoziatives Array* ist strukturell etwas anderes als ein *FiFo*. Dieses Beispiel hilft nur dann, wenn man es versteht und die angewandten Spezifikationstechniken und Notation dann auf die Lösung der Aufgabe anwendet. Auf keinen Fall genügt es, das Beispiel oberflächlich zu variieren um die Lösung zu erhalten.

3. Die VDM-Spezifikation eines abstrakten Datentyps 'Zeile' für einen Editor: *VDM-SL-Beispiele aus dem Editor EFASS*. Diese Spezifikation ist bereits relativ kompliziert (viel komplizierter als das, was wir von Euch in der folgenden Aufgabe erwarten), wir geben Sie Euch als eine Referenz, wie eine Spezifikation in VDM-SL aussehen kann. Lasst Euch durch die Strukturierung in Module nicht irritieren.

1 Spezifikation eines FiFo

Die Firma *ThinkCrime Inc.* schreibt Auftragssoftware für diverse Industriekunden. Eine Großdruckerei lässt von ThinkCrime die Steuerungssoftware für diverse Print-on-Demand-Drucker schreiben. Diese Maschinen funktionieren praktisch wie große Laserdrucker mit einer zusätzlichen Einheit zum Falten und Binden des bedruckten Papiers. Die zu druckenden Werke befinden sich im Postskript-Format auf dem Storage-Server der Firma, und sollen jeweils bei Bedarf in kleinen Stückzahlen gedruckt werden.

Die Maschinen werden mit wenig Aufsicht rund um die Uhr laufen (man erwartet eine entsprechende Auslastung), das Auftragsannahmebüro ist jedoch nur zu den üblichen Bürozeiten besetzt. Die Person, die den Auftrag annimmt, soll deshalb die Möglichkeit haben, diesen sofort zu den Print-on-Demand-Druckern zu schicken, auch wenn diese gerade mit einem anderen Druckauftrag beschäftigt sind.

Offenkundig wird eine Warteschlange benötigt, in der die Aufträge gesammelt werden können. ThinkCrime hat beschlossen die Programmierung des zentralen Moduls – einen FiFo – als Verkauftrag an Dich zu geben, möchte jedoch zuvor eine genaue Spezifikation in VDM-SL haben.

Ein Druckauftrag soll durch einen *JobDescriptor* beschrieben werden, welcher durch die Auftragsannahmesoftware in den FiFo eingereicht wird. Jedesmal wenn die Steuersoftware des Druckers erkennt, dass ein Auftrag fertig ausgedruckt ist, fragt sie ab, ob noch Aufträge anliegen, und wenn dies der Fall ist, holt sie einen neuen Descriptor aus dem FiFo, ermittelt aus ihm die zugehörige Postskript-Datei und schickt diese vom Storage-Server an den Drucker.

Um den *JobDescriptor*, sagt man Dir, müsstest Du Dich nicht kümmern, da Du ja nicht in ihn hineinschauen musst (Tip: In VDM-SL bedeutet das, dass man solche Daten mit dem Typ “*token*” modelliert). Wichtig wäre, dass Du folgende Operationen zur Verfügung stellst:

- Initialisierung einer (neuen) Warteschlange.
- Einstellen eines Auftrags in die Warteschlange.
- Holen eines Auftrags aus der Warteschlange.
- Abfrage, ob die Warteschlange leer ist.

Spezifiziere nun diese Warteschlange in VDM mittels der VDM-Toolbox. Schreibe am besten eine ‘flache’ Spezifikation (Handbuch *The IFAD VDM-SL Language*, S. 100), dies ist für kleine Probleme am angemessensten. Wenn Du das Gefühl hast, dass Deine Spezifikation nicht auf Anhieb verständlich ist, hilft vielleicht ein erläuternder Text.

Abzugeben sind: Ein sauberer Ausdruck der Spezifikation, entweder in \LaTeX gesetzt, oder als übersichtlich formatierte ASCII-Notation, mit eventuellen Erläuterungen. Die Spezifikation sollte Syntax- und Typprüfung der IFAD-Toolbox bestehen. Bewahrt bitte auch die elektronische Form Eurer Spezifikation auf.

Es bestehen keine Einwände dagegen, dass Ihr Euch beim *Ausdruck* der Spezifikation helfen lasst, bzw. gegenseitig hilft. Nur selbstverfasst sollte die Spezifikation schon sein.

Viel Spaß!