

Programmierübungen

Wintersemester 2006/2007

5. Übungsblatt

19. Juni 2007

Abgabe bis Montag, 2. Juli.

Auf den Vortragsfolien der Programmierübungen oder im Skript zur Einführung in die Informatik abgedruckte Quelltexte können verwendet werden, müssen aber der Programmierrichtlinie entsprechend formatiert und kommentiert werden.

Bitte bearbeiten Sie die Übungsaufgaben in Kleingruppen mit bis zu 3 Teilnehmern. Alle Teilnehmer einer Kleingruppe müssen in der selben Übungsgruppe eingetragen sein. Ihre Abgabe muss eine Datei `contributions.txt` enthalten, in der aufgelistet ist welche Anteile der Bearbeitung von jedem einzelnen Teilnehmer erstellt wurden. Jeder Teilnehmer muss die Funktionsweise aller Teile der Bearbeitung erläutern können. Beachten Sie die Programmierrichtlinie und kommentieren Sie Ihren Quelltext. Dokumentieren Sie unbedingt Ihre Lösungsidee in den Quelltext-Kommentaren.

<http://www.iste.uni-stuttgart.de/ps/Lehre/SS2007/inf-prokurs>

Aufgabe 5.1: AVL-Bäume

(6 Punkte)

Sie erhalten in Kürze den Quelltext des generischen `AVL_Trees`-Pakets, der von einem anderen Team der Programmierübungen erstellt wurde. Es wird Ihre Aufgabe sein, diesen Quelltext zu erweitern und evtl. zu korrigieren. Ändern Sie dabei so wenig wie möglich. Es ist *nicht* Ihr Ziel, das Paket neu zu schreiben, sondern den vorliegenden Quelltext zu verstehen. Sollten Sie Fehler korrigieren müssen, so versuchen Sie möglichst wenig zu verändern.

Auf der Webseite steht eine erweiterte Fassung der Paketspezifikation zum Download bereit. Kopieren Sie die zusätzlichen Deklarationen in die vorliegende Spezifikation. Dies können Sie wahlweise automatisch mit dem Programm `patch` und der Datei `avl_trees.ads.patch`, oder manuell anhand der Datei `avl_trees.new.ads.tun`. Ergänzen Sie die Implementierung des `Cursor`-Typs.

Aufgabe 5.2: Test

(8 Punkte)

Erstellen Sie ein Testprogramm für eine Instanz des generischen Pakets `AVL_Trees`. Messen Sie die Testüberdeckung mit `gcov` und stellen Sie sicher, dass Ihr Testprogramm Anweisungsüberdeckung in den Unterprogrammen der generischen Instanz erreicht, mit Ausnahme von `Delete` und `Exclude` (sowie allen weiteren Unterprogrammen, die nur von `Delete` oder `Exclude` aufgerufen werden). Hinweise zu `gcov` sind am Ende dieses Übungsblatts zu finden. Weitere Erklärungen auch zur Verwendung von `gcov` und zu Testüberdeckung werden in der Vortragsübung gegeben.

Ein Testprogramm führt eine Reihe einzelner Testfälle durch. Jeder Testfall besteht dabei aus einer Eingabe (z. B. Sequenz von Einfüge- und Löschoperationen, Sequenz von Vorwärts- und Rückwärts-Bewegungen eines Cursor, ...) und der Überprüfung der erwarteten Ausgabe (d. h. Konsistenz und korrekter Inhalt des AVL-Baums).

Geben Sie den Quelltext des Testprogramms und die von der `gcov`-Analyse erzeugte Datei `avl_trees.adb.gcov` ab. Nennen Sie Ihr Testprogramm `test_avl.adb`. Verwenden Sie zum Testen unbedingt die selbe Version des Pakets `AVL_Trees`, wie in Aufgabe 1 abgegeben.

Es kann sein, dass nicht alle Zeilen des Quelltexts ausgeführt werden können, falls der Quelltext überflüssige Verzweigungen enthält (sog. *totter Code*). Ist dies der Fall, so geben Sie bitte zusätzlich eine Datei `deadcode.txt` ab, in der Sie kurz und präzise erläutern, warum diese Zeilen nicht ausgeführt werden können. Nehmen Sie in Ihrer Erklärung Bezug auf die entsprechenden Zeilennummern aus der `test_avl.adb.gcov` (2. Spalte).

Hinweis: Schreiben Sie ein Unterprogramm, das die AVL-Eigenschaft des Baums überprüft und den Inhalt des Baums mit dem erwarteten Inhalt vergleicht. Legen Sie dieses Unterprogramm als Child-Unit zu dem Paket `AVL_Trees` an, um Zugriff auf die Datenstrukturen zu bekommen (analog zu `AVL_Trees.Put` von dem letzten Übungsblatt). Überlegen Sie dann geeignete Sequenzen von Aufrufen an die zu testenden Unterprogramme und rufen Sie Ihre Konsistenz-Prüfung auf.

Aufgabe 5.3: Lagerverwaltung

(6 Punkte)

Schreiben Sie ein Programm, das ein Warenlager verwaltet. Jede Ware ist gekennzeichnet durch einen eindeutigen Namen. Zu jeder Ware wird ein geplanter Verkaufspreis sowie eine Stückzahl gespeichert. Implementieren Sie eine einfache Benutzungsschnittstelle, mit der folgende Aktionen möglich sind:

- Einkaufen einer bestimmten Stückzahl einer Ware, mit Eingabe des Einkaufspreises
- Ändern des Verkaufspreises für eine bestimmte Ware
- Anzeigen einer alphabetisch sortierten Liste aller gelagerten Waren inkl. Stückzahl und Verkaufspreis
- Verkaufen einer bestimmten Stückzahl einer Ware
- Ausgabe des momentanen Gewinns bzw. Verlusts

Zu Beginn des Programms soll das Lager leer sein. Verwenden Sie das `AVL_Trees`-Paket zur Verwaltung des Lagers. Lassen Sie das Programm sinnvolle Fehlermeldungen ausgeben, falls eine Operation nicht durchgeführt werden kann.

Hinweise zu gcov

Das Werkzeug `gcov` wird dazu verwendet zu messen, welche Teile eines Programms tatsächlich ausgeführt wurden. Somit kann gemessen werden, ob ein Test tatsächlich den gesamten Quelltext einmal ausgeführt hat. Um `gcov` zu nutzen muss ein Programm *instrumentiert* werden. Dies geschieht durch Übersetzung mit besonderen Kommandozeilen-Parametern. Wie folgt wird `gcov` angewendet (Annahme: Sie verwenden eine Linux-Shell im Grundstudiumspool und haben Ihr Testprogramm `test_avl.adb` genannt):

```
$ gnatmake -f -g -fprofile-arcs -ftest-coverage test_avl.adb
```

Es wird zuerst das Test-Programm übersetzt mit den Parametern `-fprofile-arcs` und `-ftest-coverage`, dadurch wird das Programm instrumentiert und es werden zusätzlich die Dateien `*.gcno` erzeugt.

```
$ ./test_avl
```

Dann wird das Programm einmal ausgeführt. Das instrumentierte Programm hat die von Ihnen programmierte Funktionalität, darüber hinaus schreibt es ein Protokoll seiner Ausführung in die `*.gcda`-Dateien.

```
$ gcov test_avl.adb
```

Mit `gcov` wird das Protokoll analysiert und in `*.gcov` wird der Quelltext annotiert mit Anzahl der Ausführungen jeder Zeile geschrieben. Für den Test des Pakets `AVL_Trees` ist somit nur die Datei `avl_trees.adb.gcov` interessant.

In der Vortragsübung wird etwas genauer auf die Anwendung von `gcov` eingegangen und Sie haben Gelegenheit Fragen zu stellen.