Prof. Dr. Franz Schweiggert
Norbert Heidenbluth Abteilung Angewandte Informationsverarbeitung 08.06.2006
Blatt 6

Allgemeine Informatik II (SS 2006)

für
(Wirtschafts-)Mathematiker/Physiker und E-Techniker

Abgabetermin: 20. Juni 2006

Vorab zur Terminplanung: Wegen des Fronleichnam-Festes in der nächsten Woche und der Fußball-WM finden die nächsten Übungen am **Dienstag, dem 20.06.2006** statt (obwohl zeitgleich Deutschland gegen Ecuador spielt...). Entsprechend ist der Abgabetermin des Blattes der Beginn dieser Übungen (16:15 Uhr) – entweder per E-Mail oder zuvor im Tutorium. Anstelle der Übungen findet am Donnerstag, dem 22.06.2006, dann die Vorlesung statt.

Aufgabe 12: Doppelt verkettete Listen (6 Punkte)

Beim intensiven Studium der in der Vorlesung vorgestellen Listen (egal, ob sortiert oder unsortiert), ist Ihnen aufgefallen, daß es sich um sogenannte "einfach verkettete" Listen handelt: die Verkettung erfolgt nur in eine Richtung.

Mitunter wäre es aber durchaus wünschenswert, eine Liste zu verwenden, die in beide Richtungen verkettet ist ("doppelt verkettete Liste"). In solchen Listen zeigt jedes Element nicht nur auf seinen Nachfolger sondern zusätzlich noch auf seinen Vorgänger. Desweiteren gibt es analog zur Referenz auf das erste Element (`first`) auch eine Referenz auf das *letzte* Element.

Da wir ja bereits in Aufgabe 1 in diesem Semester die Feststellung getroffen haben, daß Sie immer gleich alles in der De-Luxe-Version besitzen möchten, kommen Ihnen die beiden Teilaufgaben gerade recht:

Teilaufgabe a (4 Punkte)

Erweitern Sie das *Programm 10.12* aus der Vorlesung dahingehend, daß es der Datenstruktur einer doppelt verketteten Liste entspricht. Implementieren Sie dabei auch

- ein Element aus der Liste zu entfernen und
- die Möglichkeit, die Liste (via `toString()`) wahlweise aufsteigend oder absteigend sortiert auszugegeben.

Achtung: Die Signatur der `toString()`-Methode darf dabei nicht verändert werden!

Teilaufgabe b (2 Punkte)

Schreiben Sie ein kleines Testprogramm, mit dem Sie die ordnungsgemäße Funktion Ihrer Lösung aus Teilaufgabe 12a) überprüfen und demonstrieren können.

Aufgabe 13:

Für die Abergläubischen unter Ihnen gestrichen... :-)

Aufgabe 14: Rechnen mit Polynomen (14 Punkte)

Beim Ihrem letzten Rechnerabsturz ist leider die gesamte Mathematik-Software (Maple, Matlab und Co.) gelöscht worden, und nun stehen Sie ohne die Hilfen dieser Programme da. Und das ausgerechnet jetzt, wo Sie gerade ein Programm benötigen, das Polynome addieren kann (falls Sie männlich sind: um Ihre neue Nachbarin zu beeindrucken, falls Sie weiblich sind: um sich die Zeit während der WM mit Polynom-Berechnungen zu vertreiben).

Also stehen Sie nun vor der Aufgabe, selber ein Programm zu schreiben, das zwei Polynome einliest und mit diesen ein wenig rechnen (hier: sie addieren) kann. Ein entsprechendes Hauptprogramm dazu haben Sie beim Surfen im Internet "wie zufällig" auf unserem *FTP-Server* gefunden. Leider macht es Verwendung von einer Klasse namens `Polynomial`, deren Instanzen die eigentlichen Polynom-Repräsentationen darstellen und welche auch die Methode `add(Polynomial q)` enthält, mit Hilfe derer sich ein Polynom (`q`) zu einem anderen addieren lässt. Leider ist diese Klasse nicht auffindbar, sodaß Sie sie selber implementieren müssen. Wie gut, daß Sie da in der Informatik-Vorlesung etwas von Listen gehört haben, denn mit denen lässt sich diese Aufgabe vortrefflich lösen.

Nehmen wir also an, unser Polynom bestehe aus einer (beliebigen) Anzahl von Termen der Form $a \cdot x^{\mathrm{A}} y^{\mathrm{B}} z^{\mathrm{C}}$, wobei $a \in \mathbb{Z}$ der Koeffizient sei und die Terme bereits in absteigender Reihenfolge bezogen auf die Exponenten von x, y, z (d.h. A, B, C, alle $\in \mathbb{N}$) sortiert seien. Dann ist jedes Polynom als Liste von Termen darstellbar. Das heißt, wir benötigen ebenfalls noch die Klasse `Term`, welche die folgenden Informationen beinhaltet:

- Koeffizient
- Exponent von x
- Exponent von y
- Exponent von z

- Verweis zum nachfolgenden Term

Alles klar? Nein? Gut, dann sehen wir uns mal ein Beispiel an:

Beispiel:

$$\begin{array}{cccccc} & & & x & +y & +z \\ + & 2x^2 & & & -2y & -z \\ = & 2x^2 & +x & & -y & \end{array}$$

Das Polynom der ersten Zeile $(x+y+z)$ ist demnach eine Liste der Terme $\underbrace{1 \cdot x^1y^0z^0}_{\text{Term1}}$, $\underbrace{1 \cdot x^0y^1z^0}_{\text{Term2}}$ und $\underbrace{1 \cdot x^0y^0z^1}_{\text{Term3}}$. Analog ist das Polynom der zweiten Zeile eine Liste der Terme $1 \cdot x^2y^0z^0$, $-2 \cdot x^0y^1z^0$ und $-1 \cdot x^0y^0z^1$. Die dritte Zeile ist das Ergebnis der Addition der beiden Polynome (und nunmehr der Inhalt des ersten Polynoms).

Ein zu lösendes Problem ist die Erzeugung eines Polynoms aus einem String. Grundsätzlich gibt es viele Möglichkeiten, ein Polynom als String zu repräsentieren – eine davon wird Ihnen in den Übungen vorgestellt und im folgenden nun verwendet.

In dieser Notation könnte man das erste Polynom als String z.B. wie folgt eingeben:

```
+1x +1y +1z
```

Das zweite Polynom lautete dann:

```
+2x2 -2y -1z
```

Dann sieht ein Durchlauf des vorgegebenen Main-Programms wie folgt aus:

```
theseus$ java Main
Bitte das erste Polynom eingeben:
+1x +1y +1z
Bitte das zweite Polynom eingeben:
+2x2 -2y -1z
Polynom 1: x +y +z
Polynom 2: 2x2 -2y -z
Und nun addiere ich...
Polynom 1: 2x2 +x -y    <--- enthaelt nun die Summe
Polynom 2: 2x2 -2y -z   <--- bleibt unveraendert
```

Damit sind wir nun bei den folgenden

Aufgaben:

a) Schreiben Sie die Klasse `Term`, deren Instanzen jeweils genau einen Term eines Polynoms in den Variablen x, y, z repräsentieren.

b) Schreiben Sie die Klasse `Polynomial`, deren Instanzen eine Liste von Instanzen der Klasse `Term` enthält und somit ein Polynom repräsentiert. Der Konstruktor dieser Klasse soll einen String (das Polynom) übergeben bekommen und daraus die Term-Liste erzeugen.

Ihre Klasse muß so implementiert sein, daß sie unmittelbar durch das vorgegebene Programm *Main.java* verwendet werden kann und zu einem richtigen Ergebnis führt. Das heißt also insbesondere, daß die Klasse `Polynomial` unbedingt die Methode `toString()` sowie die Methode `add(Polynomial q)` enthalten muß (welche das Polynom q zum "aktuellen" Polynom addiert).

Hinweise zu beiden Aufgaben

zu Aufgabe 12:

- Das *Programm 10.12* steht Ihnen auf unserem FTP-Server zum Download zur Verfügung.
- Es ist ratsam, sich vor dem Programmieren anhand einer Zeichnung klarzumachen, wie die Referenzen beim Einfügen bzw. Entfernen von Elementen "umgebogen" werden müssen.
- Für das Testprogramm eignet sich zum Beispiel die Verwendung der Klasse `Person`, die in den Übungen am 01.06. vorgestellt wurde (und auf unserer Vorlesungshomepage zum Download bereits steht).

zu Aufgabe 14:

- Es ist Ihnen freigestellt, ob Sie für diese Aufgabe die Liste aus der Vorlesung oder eine eigene Implementierung verwenden.
- Einen Vorschlag zum Additions-Algorithmus bekommen Sie in den Übungen.
- Diese Aufgabe eignet sich hervorragend zum Aufteilen in der Gruppe!

Viel Erfolg und viele Tore bei der WM!