

Programmierübungen

Wintersemester 2006/2007

11. Übungsblatt

19. Januar 2007

Abgabe bis Samstag, 27. Januar 23:59 Uhr.

Die Abgabe Ihrer Bearbeitung können Sie im eClaus-System durchführen. Erarbeiten Sie Lösungsideen zu den Aufgaben möglichst in Kleingruppen. Es wird jedoch von Ihnen erwartet, dass jeder Teilnehmer eine eigene Lösung abgibt. Sollten kopierte Quelltexte abgegeben werden, so werden grundsätzlich alle Kopien mit 0 Punkten bewertet. In 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.

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/WS0607/inf-prokurs>

Am Montag 22.1.2007, 8:00 Uhr in V38.01 wird das Aufgabenblatt kurz vorgestellt und Sie haben Gelegenheit Fragen zu den Aufgaben zu stellen.

Bitte beachten Sie, dass zum Scheinerwerb unter anderem auf den letzten 4 Aufgabenblättern 50 % der Punkte erreicht werden müssen. Das letzte Aufgabenblatt wird Nummer 13 sein.

Aufgabe 11.1: Zinsrechnung

(10 Punkte)

In dieser Aufgabe soll das momentane Guthaben auf einem Sparkonto berechnet werden. Als Eingabe dienen zwei Textdateien, `einzahlung.txt` und `zinssatz.txt`. In der Datei `einzahlung.txt` werden Ein- und Auszahlungen auf das Konto mit Tag der Wertstellung aufgelistet. In der Datei `zinssatz.txt` werden Änderungen des jährlichen Guthabenszinssatzes mit Datum der Änderung aufgelistet.

Format von `einzahlung.txt`: Alle Zeilen sind chronologisch geordnet. Als erstes steht in jeder Zeile das Datum der Wertstellung im Format `TT.MM.JJJJ`, dann folgen beliebig viele, aber mindestens ein Leerzeichen, dann folgt der verbuchte Umsatz als übliche Fließpunktzahl in einer beliebigen Währung (aber immer in der selben). Positive Zahlen repräsentieren Einzahlungen, negative Zahlen Abbuchungen.

Beispiel:

```
01.03.2006 +34.5
07.03.2006 +13.25
05.05.2006 -60
02.08.2006 -30
30.11.2006 +1534.87
30.11.2006 +220.17
20.12.2006 -548.22
02.01.2007 -345
```

Format von `zinssatz.txt`: Analog zu `einzahlung.txt`, jedoch wird die Fließpunktzahl als Zinssatz in Prozent pro Jahr interpretiert.

Beispiel:

05.07.2006 2.75

01.12.2006 3.0

Ihr Programm kann davon ausgehen, dass die Dateien keine Fehler enthalten und braucht keine Konsistenzprüfungen durchzuführen.

Die Berechnung des Guthabens verläuft folgendermaßen:

- Jeder Monat hat nach Definition 30 Tage.
- Da in den Eingabe-Dateien jeweils das Datum der tatsächlichen Wertstellung angegeben wird, kann der Fall nicht auftreten, dass dort ein Datum angegeben ist, das auf den 31. Tag eines Monats fällt oder kein Geschäftstag der Bank ist.
- Das Startguthaben des Kontos vor dem ersten Eintrags in `einzahlung.txt` beträgt 0.00.
- Einzahlungen zählen ab dem Tag ihrer Wertstellung zum Guthaben und geben ab diesem Tag Zinsen.
- Auszuzahlende Beträge werden nur bis zum Tag direkt vor der Wertstellung verzinst. Reicht das Guthaben für eine Auszahlung nicht aus, so wird die Buchung ignoriert. Das Guthaben kann nie negativ werden.
- Für alle Tage vor dem ersten Datum in `zinssatz.txt` werden keine Zinsen bezahlt. Der neue Zinssatz gilt jeweils ab dem in der Datei angegebenen Datum (einschließlich).
- Zinserträge werden für den Zeitraum bis 30.12. jedes Jahres berechnet und am 01.01. des darauffolgenden Jahres wie eine gewöhnliche Einzahlung gutgeschrieben. Für jeden Tag t (Achtung, jeder Monat hat genau 30 Tage!) werden die Zinsen nach der Formel $g(t) \cdot \frac{z(t)}{36000}$ berechnet, wobei $g(t)$ das Guthaben am Tag t darstellt und $z(t)$ den am Tag t gültigen jährlichen Zinssatz (z. B. 2.75).
- Wird das Sparkonto aufgelöst, so werden Zinsen bis zum Tag vor der Auflösung berechnet und mit ausgezahlt.

Schreiben Sie ein Programm `guthaben`, das den Benutzer ein Auflösungsdatum eingeben lässt und anhand der Dateien `zinssatz.txt` und `einzahlung.txt` das Guthaben des Sparkontos zu diesem Datum berechnet. Liegt das Auflösungsdatum so früh, dass noch weitere Buchungen oder Zinssatzänderungen ausstehen, so werden die ausstehenden Ereignisse ignoriert und das Konto vorzeitig aufgelöst. Lassen Sie Ihr Programm ein Protokoll ausgeben, das alle relevanten Aktionen und den am Auflösungsdatum ausgezahlten Kontostand enthält.

Ihr Programm muss die Dateien zunächst vollständig einlesen und den Inhalt in Listen verwalten. Lösungen ohne den Gebrauch von Listen werden nicht akzeptiert.

Hinweis: Auf der Webseite wird ein Paket `Dates` angeboten. Dieses Paket stellt einen Datentyp zur Verarbeitung eines Datums zur Verfügung. Sie können das Paket einsetzen (auch verändern oder erweitern) um Ihre Implementierung zu vereinfachen.

Beispiel für einen Programmablauf:

```
Lesen von einzahlung.txt...
Lesen von zinssatz.txt...
Berechnung bis (Datum): 4.4.2007
Kontobewegung am 01.03.2006: 34.50
Kontobewegung am 07.03.2006: 13.25
Fehlgeschlagene Buchung am 05.05.2006: -60.00
Zinssatzänderung am 05.07.2006 auf 2.75 % p.a.
Kontobewegung am 02.08.2006: -30.00
Kontobewegung am 30.11.2006: 1534.87
Kontobewegung am 30.11.2006: 220.17
Zinssatzänderung am 01.12.2006 auf 3.00 % p.a.
Kontobewegung am 20.12.2006: -548.22
Zinsgutschrift am 01.01.2007: 4.32
Kontobewegung am 02.01.2007: -345.00
Zinsgutschrift am 04.04.2007: 6.88
Guthaben bei Kontoauflösung am 04.04.2007: 890.77
```

Aufgabe 11.2: Externe Repräsentation

(10 Punkte)

In dieser Aufgabe soll ein gerichteter Graph aus einer Datei geladen und mit Hilfe von Adjazenzlisten repräsentiert werden. Aus dieser Darstellung soll eine Adjazenzmatrix des Graphen erstellt werden.

Eine Graph-Datei soll wie folgt aufgebaut sein: In jeder Zeile der Datei ist eine Kante spezifiziert mit Startknoten-Id, Kantengewicht und Zielknoten-Id. Die Knoten-Ids sind Zeichenfolgen bestehend aus den Buchstaben a-z und A-Z, Kantengewichte sind natürliche Zahlen im Bereich 1–9. Eine Zeile hat folgendes Format, wobei *Source* für die Knoten-Id des Quellknotens steht, *Target* für die Knoten-Id des Zielknotens und *w* für das Gewicht der Kante:

Source (W) Target

Schreiben Sie ein Programm `Put_Adjacency`, das eine Textdatei „graph.txt“ einliest und eine Adjazenzmatrix für den Graph als Tabelle ausgibt. Die Zeilen und Spalten der Tabelle müssen mit den Knoten-Ids beschriftet sein. In jeder Zelle $a_{z,s}$ der Tabelle soll das Gewicht der Kante von z nach s eingetragen sein, falls diese Kante existiert, andernfalls soll dort 0 stehen.

Ihr Programm muss den Graph zunächst komplett einlesen und als Adjazenzlisten verwalten. Hinweise zur Wahl der Datenstrukturen sind im Skript zur „Einführung in die Informatik 1“ angegeben. Achten Sie darauf, dass auch für verschieden lange Knoten-Ids die Spalten der Tabelle korrekt untereinander ausgegeben werden.

Beispiel:

Datei `graph.txt`:

```
LO (1) RU
LO (2) Spitze
LU (3) LO
LU (3) RU
```

RO (1) LU
RO (3) LO
RU (3) RO
Spitze (2) RO

Ausgabe:

	LO	RU	Spitze	LU	RO
LO	0	1		2	0
RU	0	0		0	3
Spitze	0	0		0	2
LU	3	3		0	0
RO	3	0		0	1

Hinweis: Das Analysieren der Textdatei kann durch zeilenweises Einlesen geschehen. Die Positionen der Klammern innerhalb eines Strings können durch die Funktion `Ada.Strings.Fixed.Index` leicht ermittelt werden.