Dr. Andreas Borchert
Norbert Heidenbluth

Abteilung Angewandte Informationsverarbeitung

23.06.2005
Blatt 9

Allgemeine Informatik II für Mathematiker/Wirtschaftsmathematiker (SS 2005)

Abgabetermin: 30. Juni 2005

Aufgabe 12: Was guckst Du? (8 Punkte)

Neulich ergab sich der folgende Dialog in der Mensa:

> "Schau' mal: der Typ an der Kasse, der sieht aus wie Richard Gere!" "Richard Gere? Aus welchem Film kenne ich den?" "Zum Beispiel aus Pretty Woman!". "Ach ja richtig! Der Film ist toll! Wer hat da nochmal die weibliche Hauptrolle gespielt?" "Julia Roberts!" "Na klar, die hat doch auch in Ocean's Elevens mitgespielt." "Hmm, macht in Ocean's Eleven nicht auch Brad Pitt[1] mit?". "Ja, genau, den find' ich seit seinem Film Sieben Jahre in Tibet ganz toll!" "Sieben Jahre in Tibet? Da macht doch auch B.D. Wong mit. In welchem Film hat der denn noch mitgespielt?" "Keine Ahnung!" "Ich auch nicht!"

Was hat das mit unserem Übungsblatt zu tun? Nun, jetzt, wo bald die Ferien anfangen und Sie nicht wissen, was Sie ohne unterhaltsame Übungsblätter mit Ihrer Zeit anfangen werden, sind Sie dankbar für jede Anregung, die den Geist auf Trab hält. Und warum nicht mal die *Internet Movie Database* auswendig lernen?

Und wenn Sie dann alles auswendig können, möchten Sie ihr Wissen auch testen, und dazu können Sie dann das Programm verwenden, das Sie in dieser Aufgabe schreiben sollen:

Aufgabe

Schreiben Sie ein Oberon-Programm, mit dem Sie einen Dialog wie im Beispiel angegeben "führen" können. Sie geben also zu Beginn den Namen eines Schauspielers[2] ein. Als nächstes erwartet das Programm die Eingabe eines Filmtitels (leider müssen dies hier bei uns die **Originaltitel** sein!), in dem dieser Schauspieler mitgespielt hat. War die Eingabe korrekt, fragt das Programm als nächstes nach einem anderen Schauspieler, der ebenfalls in dem zuletzt genannten Film dabei war. Bezogen auf diesen neuen Schauspieler wird nun wieder nach einem Filmtitel gefragt, in dem der genannte Schauspieler mitmacht. So geht es immer weiter. Wenn Sie eine falsche Antwort gegeben haben, ist das Spiel beendet, und Sie bekommen Ihre Punktzahl (Anzahl korrekter Antworten) genannt.

Wichtig: Ein Schauspieler soll nicht mehrfach genannt werden dürfen!

Hinweise und Tips:

- Auf unserem FTP-Server haben wir den Inhalt der Filmdatenbank *lokal verfügbar* gemacht und in ein einfaches Dateiformat überführt. Erklärungen hierzu entnehmen Sie bitte der entsprechenden `README`-Datei sowie den Übungen. Dennoch hier noch ein paar besonders wichtige Aspekte aus dieser `README`-Datei:
  - Die Filmdatenbank **darf nicht in Ihr Heimatverzeichnis** bei uns kopiert werden! Arbeiten Sie stattdessen entweder mit einem symbolischen Link[3] oder verwenden Sie in Ihrem Programm den (absoluten) Pfad zur Datenbank-Datei. Dieser lautet:

    `/home/ftp/root/pub/vorlesungen/ss05/ai2/uebungen/blatt9/daten/movies`
  - Eine Kopie dieser Datei ist nur dann zulässig, wenn Sie die Datei auf Ihrem eigenen Rechner verwenden möchten.
  - Diesen Hinweis und noch mehr finden Sie auch in der `README`-Datei unter dem o.g. Pfad.
- Die Datenbank-Datei besitzt das folgende Format:
  - Jede Zeile besteht aus einem Filmtitel, gefolgt von den Namen der Schauspieler, die an diesem Film teilgenommen haben.
  - Als Feldtrenner wird die Tilde verwendet.
  - In den Übungen wird dies nochmal anhand eines kleinen Beispiels verdeutlicht!
- Im Modul `Main` finden Sie die Prozedur `ConfGC`, welche die Garbage-Collection konfiguriert. Diese Prozedur **muß in Ihrem Programm unbedingt verwendet werden**, da es ansonsten aus Perfomance-Gründen nahezu unmöglich würde, die Datenbank einzulesen. Sie können die Prozedur so wie sie ist übernehmen!
- Ausdrücklich gefordert ist in diesem Übungsblatt die Modularisierung. Einen Vorschlag für eine sinnvolle Modularisierung finden Sie weiter unten.
- Für die Verwaltung der Schauspieler bietet sich als Datenstruktur der Hash an und insbesondere auch deren Verschachtelung ("Hash von Hashes"). Sinnvollerweise sollte

[1] Wenn Sie weiblich sind, dürfen Sie jetzt kurz hyperventilieren…
[2] Das können natürlich auch Schauspielerinnen sein!
[3] Kommando: `ln -s /home/ftp/root/pub/vorlesungen/ss05/ai2/uebungen/blatt9/daten/movies .`

  - ein Hash für das Verwalten der Schauspieler sowie
  - ein weiterer Hash für alle Filme

  verwendet sowie

  - die Beziehung zwischen Schauspielern und Filmen

  umgesetzt werden.

  Die Beziehung lässt sich entweder dadurch realisieren, daß entweder jeder Schauspieler "weiß", an welchen Filmen er teilgenommen hat oder bei jedem Film steht, welche Schauspieler daran beteiligt waren.

  Empfohlen sei hier die erstgenannte Möglichkeit, denn wenn bei den Schauspielern jeweils ein Hash mit den Filmtiteln untergebracht wird[4], reduziert das die Gesamtzahl der Hash-Tabellen, da die Zahl der Schauspieler geringer als die der Filme ist. Wichtig ist hier, daß in den Hash für die Schauspieler direkt die Film-Objekte eingetragen werden. Das geht problemlos, da ein Objekt in beliebig viele Hash-Tabellen eingefügt werden kann (zumindest ist das der Fall, wenn Hashes verwendet wird). Keinesfalls darf ein referenziertes Objekt für jede Referenz neu erzeugt werden, da dies zur Explosion des benötigten Speichervolumens führen würde.

- In Zusammenhang mit der Modularisierung empfiehlt sich diesmal insbesondere wieder die Aufteilung einzelner Module innerhalb der Gruppe, um den Arbeitsaufwand für jeden einzelnen überschaubar zu halten.

- Das Programm geht — auch bei richtiger Implementierung — an die Grenzen der Speicherauslastung der Rechner. Arbeiten Sie daher nach Möglichkeit auf der turing und nicht lokal auf unseren Workstations!
  (Zur Information: Es werden 135.006 Filme und 37.181 Schauspieler in 1.673.119 Rollen verwaltet!)

- Um nicht unnötig viel Speicherplatz zu ver(sch)wenden, sollten Sie die `CHAR`-Arrays, die zum Speichern von Filmtiteln und Schauspielernamen verwendet werden, auf die maximale Länge von 128 Zeichen begrenzen.

Die Module

Nachstehend finden Sie eine kurze Erklärung der für diese Aufgabe benötigten Module. Die genauen Spezifikationen entnehmen Sie bitte den zugehörigen od-Dateien, die sie wie immer auf unserem *FTP-Server* finden. Die Module sind in topologischer Reihenfolge erklärt und geben in der ersten (kursiv gesetzten) Zeile jeweis an, wie sie sich gegenseitig verwenden:

[4] Hier haben wir dann einen "Hash von Hashes"!

`Hashes`

Dieses Modul entspricht – zumindest in der Beispiellösung – vollständig dem Beispiel *Hashes* aus der Vorlesung.

Dies müssen Sie für diese Aufgabe nicht verwenden, wenn Sie gerne ein eigenes Hash-Modul schreiben möchten. Aber selbstverständlich können Sie es gerne verwenden!

`NameObjects`

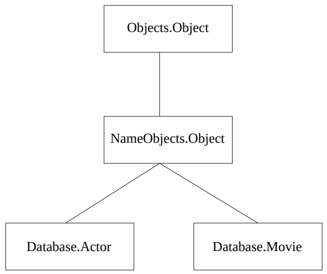
(unterstützt Hashes für Objekte mit einem Namen)

Diese Abstraktion ist als Hilfestellung gedacht, um besser mit dem Hash-Modul aus der Vorlesung arbeiten zu können, welches als Datentyp Erweiterungen von `Objects.Object` erwartet.

Dementsprechend stellt `NameObjects` einen Datentyp zur Verfügung, der eine Erweiterung von `Objects.Object` ist und Strings verwalten kann. Letztlich braucht man sowohl für das Speichern von Schauspielern als auch für das Speichern der Filmtitel jeweils nur einen String – aber eben in einem vom das Hash-Modul erwarteten Weise. Genau dies wird durch `NameObjects` zur Verfügung gestellt: abgesehen vom Datentyp selber, werden hier auch die notwendigen Funktionen zur Aufnahme einer Variablen vom Typ `NameObjects` in den Hash definiert.

Das bedeutet konkret, daß es sich bei `NameObjects` um eine *Faktorisierung* des Namensteils von Schauspielern und Filmen handelt, die es einem erspart, die drei Prozeduren für Hashes doppelt zu schreiben.

Damit ergibt sich die folgende Typ-Hierarchie:



`Database`

(nutzt NameObjects und Hashes zum Aufbau einer Datenbank)

Dieses Modul stellt die für den Aufbau der Datenbank erforderliche Datenstruktur sowie die notwendigen Zugriffsmechanismen zur Verfügung.

`DBLoader`

(füllt eine Datenbank aus dem Database-Modul)

Dieses Modul stellt die Funktion `Load` zur Verfügung, mit der eine Filmdatenbank eingelesen wird.

`Game`

(füllt ein Spiel durch mit einer gefüllten Datenbank)

Dieses Modul bietet die Prozedur `Run` an, die ein Spiel steuert.

`Main.o[dm]`

Dieses Modul ist das Hauptmodul des Spiels. Hier wird insbesondere auch der Pfad zur Datenbank-Datei angegeben. Abgesehen von dieser Anpassung können Sie das Modul aber ohne Änderungen übernehmen. Dies stellen wir Ihnen zur Verfügung.

Beispiel:

Wie immer geben wir zum Schluß der Aufgabenstellung noch ein Beispiel an (dies entspricht dem eingangs geschilderten "Mensa-Dialog"):

```
*** Schauspieler / Filme Raten ***
Schauspieler: Richard Gere
Film: Pretty Woman
Schauspieler: Julia Roberts
Film: Ocean's Eleven
Schauspieler: Brad Pitt
Film: Seven years in tibet
Schauspieler: B.D. Wong
Film: Lethal Weapon
Pech! Leider hat B.D. Wong nicht in Lethal Weapon mitgespielt!
Gut! Sie haben 3 Punkte erreicht!
```

Aufgabe 13: Eine kleine Verbesserung... (2 Punkte)

Erweitern Sie das Programm so, daß die Groß- und Kleinschreibung nicht mehr relevant ist. Die Eingabe "Seven years in Tibet" soll zum Beispiel genauso gewertet werden wie "Seven Years In Tibet".

Tip: Für diese Aufgabe ist lediglich eine Änderung im Modul `NameObjects` notwendig. Die übrigen Module brauchen nicht verändert zu werden!

Viel Erfolg!