

Programming C++

Project Stammbaum

Status:

23. Mai 2023, 17:28

Supervisor: Prof.Dr. G. Haase,

`gundolf.haase@uni-graz.at`

Grundaufgabe

Die Grundaufgabe besteht darin einen Stammbaum zu erstellen welcher passende Container¹ und Algorithmen der STL benutzt.

Wir gehen zunächst von einer klassischen Familie (keine unehelichen Kinder, keine Scheidungen, keine Patchwork-Familien) in mehreren Generationen aus und werden einige Informationen aus dem Stammbaum herausholen.

(6 Pkt.)

1. Generieren Sie einen einfachen Stammbaum (mind. 3 Generationen) und testen Sie folgende Funktionalitäten:

- Ausgabe aller Kinder einer Person.
- Ausgabe aller (vorhandenen) Eltern einer Person.
- Ausgabe des Stammbaumes nach unten (alle *kinder) einer Person.
- Ausgabe des Stammbaumes nach oben (alle *eltern) einer Person.
- Anzahl und Namen aller Geschwister, Anzahl und Namen aller Onkel/Tanten (keine Unterscheidung m/w) einer Person.
- Anzahl und Namen Cousins/Cousinen 1. Grades, desgleichen Großeltern einer Person.

2. Lesen Sie einen Stammbaum vom ASCII-File ein, nehmen Sie `ex_stamm.txt`².

In jeder sinnvollen Zeile des Files stehen folgende Daten:

```
globale_ID    Vorname      Name          gID_Elternteil    gID_Elternteil
```

mit Leerzeichen als Trennzeichen. Die Elternteile (siehe gIDs) müssen vorher im File stehen und `gid==-1` bedeutet, daß kein Elternteil bekannt ist.

Alle nicht mit einer Zahl beginnenden Zeilen, inklusive Leerzeilen, sollen als Kommentar gelten ("sinnlose Zeilen"). Diese Zeilen können Sie am Anfang gern entfernen.

- Geben Sie den Stammbaum nach unten von `gid==1` (Linna Liebig) aus und die Anzahl ihrer Urenkel.
- Geben Sie den Stammbaum nach oben von `gid==17` (Jonas Wolf) aus, desgleichen alle Geschwister von `gid==17`.
- Geben Sie alle Geschwister von `gid==3` (Manfred Gruber) aus.
- Geben Sie alle Tanten/Onkel von `gid==21` aus.
- Geben Sie alle Cousins/Cousinen von `gid==25` aus.

¹<http://www.cplusplus.com/reference/stl/>

²http://imsc.uni-graz.at/haasegu/Lectures/Kurs-C/SS21/projects/Stammbaum/ex_stamm.txt

3. Weitere Aufgaben:

- Schreiben Sie eine Methode welche für den eingelesenen Stammbaum ein Kind (*Susi Wolf*) von `gid==13` und `gid==10` hinzufügt und bestimmen Sie dessen Geschwisteranzahl. Unterscheiden Sie zwischen Geschwistern und Halbgeschwistern.
- Schreiben Sie eine Methode welche eine neue Person Leo Brunner als noch fehlenden Elternteil bei `gid==13` und `gid==14` hinzufügt (Ergebnis der Ahnenforschung).
- Schreiben Sie Ihren geänderten Stammbaum in ein File *out_stamm.txt*.
Lesen Sie diesen Stammbaum danach wieder vom File ein testen Sie diesen mit den Aufgaben von Punkt 2.

Tipps zum Herangehen an diese Aufgabe

Wie immer gibt es mehrere Möglichkeiten diesen Stammbaum zu realisieren. Daher einige Ideen welche praktikabel sind.

- Generieren Sie zuerst eine Klasse welche Personendaten (Name, Vorname) speichert, darunter eine global eindeutige Kennung (*global ID*) und Datenstrukturen für Kinder sowie Eltern.
- Die Personendaten des Stammbaumes müssen in passenden Containern³ gespeichert werden. Die Container `map`, `set`, `vector` sind benutzbar - mit unterschiedlichem Aufwand bzgl. des Zugriffs auf eine konkrete Person. Im Falle von `vector` sollten die Personendaten den Index enthalten wo die Daten im Vektor gespeichert sind.
- Zum File-IO können Sie das Modul *file_io*⁴ entsprechend adaptieren.
- Zum Anlegen einer neuen Person war mir ein Konstruktor der Form

```
Person(const int gid, const string& vorname, const string& familie,  
       const int a_gid=-1, const int b_gid=-1)
```

sehr nützlich (`a_gid` ist die globale ID des ersten Elternteils).
- Methoden wie `Add_Parent` und `Add_Child` fügen neuen Elternteile bzw. neue Kinder zur aktuellen Person hinzu.
- Methoden/Funktionen wie die Ausgabe des Baumes (up/down), Zählen aller Nachkommen sind rekursiver Natur und sollten auch so implementiert werden.
- Das professionelle Fileformat zur Stammbaumeingabe ist GEDCOM⁵ welches in Projekten wie *ancestry*⁶ benutzt wird.

³<http://www.cplusplus.com/reference/stl/>

⁴lhome/file_io.zip

⁵<https://de.wikipedia.org/wiki/GEDCOM>

⁶<http://trees.ancestry.de/pt/learnmore/gedcom.aspx>

Zusatzaufgabe I

Nunmehr kommen wir zu modernen Patchwork-Familien nach dem Motto: “Deine Kinder und meine Kinder streiten sich gerade mit unseren Kindern”.

(+2 Pkt.)

- Lesen Sie 2 Stammbäume (mind. 3 Generationen) von zwei Files ein bei denen mind. eine Person in beiden Stammbäumen auftritt und in jedem Stammbaum Kinder (und evtl. Enkelkinder) hat.
- Vereinigen Sie diese beiden Stammbäume.
- Ausgabe des neuen Stammbaumes nach oben/unten, Angabe der Enkelkinder, Geschwister/Halbgeschwister, Onkel und Tanten.
- Abspeichern des neuen Stammbaumes in einem File *patch_stamm.txt*.

Zusatzaufgabe II

Geben Sie den Stammbaum bzgl. einer Person generationsweise aus, also:

(+2 Pkt.)

- Alle Großeltern (+Großonkel)
- danach alle Eltern+Onkel/Tanten, danach alle Geschwister,
- alle Kinder+Neffen, alle Enkelkinder+Großneffen.