

3D Programmierpraktikum

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum Mittwoch den **30. April 2008, 12:00 Uhr s.t.** abzugeben.

Inhalt:

Diese Blatt dient zum Bekanntmachen mit der Programmiersprache C++. Dazu werden Funktionen, Kontrollstrukturen, Zeiger, Arrays und elementare Ein-/Ausgabe Aspekte betrachtet.

Aufgabe 1 (P) Funktionen

In dieser Aufgabe soll die Programmierung von Funktionen und die Integration von Kontrollstrukturen geübt werden. Dafür wird eine Funktion zur Berechnung der Fakultät als Beispiel verwendet.

- Schreiben Sie eine Applikation, die über eine Funktion `Factorial` verfügt, die die Fakultät einer vom Benutzer eingegebenen Zahl berechnet und ausgibt. Erstellen Sie hierzu lediglich eine Datei `Factorial.cpp` ohne header-Datei mit nur zwei Funktionen, einer `main`-Funktion und der oben genannten. Verwenden Sie für die Berechnung der Fakultät die Rekursion. Definieren Sie zuerst die `main`-Funktion, dann die Funktion `Factorial` und verzichten Sie auf eine explizite Deklaration der Funktionen.
- Drehen Sie nun die Definition der beiden Funktionen um, so dass zuerst `Factorial`, dann `main` definiert wird. Versuchen Sie die Quelldatei zu kompilieren.
- Fügen Sie vor den Funktionsdefinitionen nun die Deklarationen ein, also die Funktionsprototypen. Kompilieren Sie erneut die Quelldatei.
- Erstellen Sie nun eine header-Datei `Factorial.h` und setzen Sie dort die Funktionsprototypen ein. Löschen Sie diese aus der `.cpp`-Datei und fügen Sie stattdessen ein `#include`-Statement ein, das die header-Datei einbindet.

Aufgabe 2 (P) Kontrollstrukturen und I/O

Schreiben Sie ein Programm, das den Wochentag für ein vom Benutzer eingegebenes Datum ausgibt. Erstellen Sie dafür eine Datei `Weekday.cpp` mit den entsprechenden Funktionen. Falls nötig, erstellen Sie ebenfalls einen header-Datei. Für die Berechnung des Wochentages sollen die Regeln des gregorianischen Kalenders gelten:

- Die Zeitrechnung beginnt drei Tage nach Christi Geburt, am 1. Januar 0001. Dies ist ein Samstag.
- Schaltjahre werden folgendermaßen berechnet: Bis 1599 ist jedes vierte Jahr ein Schaltjahr, wie es im julianischen Kalender festgelegt wurde. Ab 1600 ist jedes vierte Jahr ein Schaltjahr, es sei denn, es handelt sich um ein Jahrhundert, das nicht durch 400 teilbar ist. Also ist 1600 oder 2000 ein Schaltjahr, aber 1700 oder 1900 nicht. Natürlich sind aber z.B. 1904, 1724 usw. wiederum schon Schaltjahre.

- Der zusätzliche Tag bei Schaltjahren ist der 29. Februar.
- Auf den 4. Oktober 1582 folgte gleich der 15. Oktober 1582, wobei die Abfolge der Wochentage dabei unverändert blieb.

Diese Regeln wurden von Papst Gregor in einer päpstlichen Bulle am 24. Februar 1582 erlassen. Versuchen Sie, für Wochentage und Monatsnamen enumerations zu verwenden. Sollte es dabei zu Konvertierungsschwierigkeiten kommen, versuchen Sie ein type-casting anzuwenden, wie Sie es aus Java kennen.

- a) An welchem Wochentag hat Papst Gregor die Bulle zur Einführung des gregorianischen Kalenders erlassen?
- b) An welchem Wochentag wurde der Bau der Berliner Mauer begonnen¹?
- c) An welchem Tag fiel die Berliner Mauer²?
- d) An welchem Wochentag hielt John F. Kennedy seine berühmte Rede in Berlin³?
- e) An welchem Wochentag wurde Kennedy erschossen⁴?
- f) An welchem Wochentag wurde der FC Bayern München gegründet⁵?
- g) Was für ein Wochentag ist Sylvester 2008?
- h) An welchem Wochentag war Ihre letzte Vordiplomsprüfung?

Erstellen Sie ein Programm das es ermöglicht interaktiv Daten von der Kommandozeile einzugeben, darauf hin soll das Programm den dazu gehörigen Wochentag ausgeben.

Aufgabe 3 (P) Strukturen (structs), Zeiger (pointer) und Felder (arrays)

Ein Zeiger vom Typ T wird durch T* ausgedrückt. Ein einfaches Beispiel hierfür ist:

```
int no = 23;
int* p = &no; // p holds the address of no
int no2 = *p; // no2 == 23
```

Dieses einfache Programm verdeutlicht den Unterschied zwischen einem Objekt von einem bestimmten Typ T und einem Pointer auf das Objekt. Der Pointer gibt die Speicheradresse an, in der das Objekt dieses Datentyps liegt. Eine Struktur bezeichnet die Zusammenfassung von Daten beliebigen Types:

```
struct PhonebookEntry {
string lastName;
string firstName;
int phoneNumber;
};
```

¹13. August 1961

²9. November 1989

³26. Juni 1963

⁴22. November 1963

⁵27. Februar 1900

Mit Hilfe der eben eingeführten Struktur soll nun eine Art Telefonbuch realisiert werden, in dem jedem Namen (Nachname und Vorname) jeweils eine Rufnummer zugeordnet wird. Zusätzlich zu den trivialen Funktionen

```
void insertEntry(PhonebookEntry a); //Referenz oder Pointer?
```

```
void deleteEntry(PhonebookEntry a); //Referenz oder Pointer?
```

soll eine Sortierfunktion implementiert werden, die das gesamte Telefonbuch nach Namen sortiert. Achten Sie hierbei auf korrekte und effiziente Verwendung von Pointern und Referenzen. In der C++-StandardLibrary (STL) befindet sich ein sort-Algorithmus, der selbstverständlich verwendet werden darf. Näheres dazu finden Sie auf www.codeproject.com/vcpp/stl/stdsort.asp