



C++ - Modul 00

Namensräume, Klassen, Mitgliedsfunktionen, stdio-Streams, Initialisierungslisten, static, const, und einige andere grundlegende Dinge

Zusammenfassung:

Dieses Dokument enthält die Übungen des Moduls 00 von C++-Modulen.

Fassung: 9

Inhalt

I	Einführung	2
II	Allgemeine Regeln	3
III	Übung 00: Megaphon	5
IV	Übung 01: Mein geniales Telefonbuch	6
V	Übung 02: Der Job Ihrer Träume	8
VI	Einreichung und Peer-Evaluierung	10

Kapitel I

Einleitung

C++ ist eine allgemeine Programmiersprache, die von Bjarne Stroustrup als Weiterentwicklung der Programmiersprache C oder "C with Classes" (Quelle: [Wikipedia](#)) entwickelt wurde.

Das Ziel dieser Module ist es, Sie in die **objektorientierte Programmierung** einzuführen. Dies wird der Ausgangspunkt für Ihre C++-Reise sein. Viele Sprachen werden empfohlen, um OOP zu lernen. Wir haben uns für C++ entschieden, da es von Ihrem alten Freund C abgeleitet ist. Da es sich um eine komplexe Sprache handelt, und um die Dinge einfach zu halten, wird Ihr Code dem Standard C++98 entsprechen.

Wir sind uns bewusst, dass modernes C++ in vielerlei Hinsicht anders ist. Wenn Sie also ein kompetenter C++-Entwickler werden wollen, liegt es an Ihnen, nach den 42 Common Core weiterzugehen!

Sie werden Schritt für Schritt neue Konzepte entdecken. Die Übungen werden nach und nach immer komplexer.

Kapitel II

Allgemeine

Vorschriften

Kompilieren

- Kompilieren Sie Ihren Code mit C++ und den Flags -Wall -Wextra -Werror
- Ihr Code sollte sich trotzdem kompilieren lassen, wenn Sie das Flag -std=c++98 hinzufügen

Formatierungs- und Benennungskonventionen

- Die Übungsverzeichnisse werden folgendermaßen benannt: ex00, ex01, ... , exn
- Benennen Sie Ihre Dateien, Klassen, Funktionen, Mitgliedsfunktionen und Attribute wie in den Richtlinien gefordert.
- Schreiben Sie Klassennamen im Format **UpperCamelCase**. Dateien, die Klassencode enthalten, werden immer nach dem Klassennamen benannt. Zum Beispiel: Klassenname.hpp/Klassenname.h, Klassenname.cpp, oder Klassenname.tpp. Wenn Sie also eine Header-Datei haben, die die Definition einer Klasse "BrickWall" enthält, die für eine Ziegelmauer steht, lautet ihr Name BrickWall.hpp.
- Wenn nicht anders angegeben, müssen alle Ausgabemeldungen mit einem Zeilenumbruch abgeschlossen und auf der Standardausgabe ausgegeben werden.
- *Auf Wiedersehen Norminette!* In den C++-Modulen ist kein Kodierungsstil vorgeschrieben. Sie können Ihrem Lieblingsstil folgen. Aber denken Sie daran, dass ein Code, den Ihre Mitbewerber nicht verstehen, auch nicht benotet werden kann. Geben Sie Ihr Bestes, um einen sauberen und lesbaren Code zu schreiben.

Erlaubt/Verboten

Sie programmieren nicht mehr in C. Zeit für C++! Deshalb:

- Sie dürfen fast alles aus der Standardbibliothek verwenden. Anstatt sich also an das zu halten, was Sie bereits kennen, wäre es klug, so viel wie möglich die C++-ähnlichen Versionen der C-Funktionen zu verwenden, an die Sie gewöhnt sind.
- Sie können jedoch keine andere externe Bibliothek verwenden. Das bedeutet, dass C++11 (und abgeleitete Formen) und Boost-Bibliotheken verboten sind. Die

folgenden Funktionen sind ebenfalls verboten: `*printf()`, `*alloc()` und `free()`. Wenn Sie sie verwenden, wird Ihre Note 0 sein und das war's.

- Beachten Sie, dass, sofern nicht ausdrücklich anders angegeben, der using-Namensraum `<ns_name>` und Freundschaftswörter sind verboten. Andernfalls wird Ihre Note -42 sein.
- **Du darfst die STL nur in den Modulen 08 und 09 verwenden.** Das bedeutet: keine **Container** (Vektor/Liste/Map/usw.) und keine **Algorithmen** (alles, was den `<algorithm>`-Header erfordert) bis dahin. Andernfalls wird Ihre Note -42 sein.

Einige Designanforderungen

- Speicherlecks treten auch in C++ auf. Wenn Sie Speicher zuweisen (mit der Funktion `new` Schlüsselwort), müssen Sie **Speicherlecks** vermeiden.
- Von Modul 02 bis Modul 09 muss der Unterricht in der **orthodoxen kanonischen Form** gestaltet werden, **es sei denn, es ist ausdrücklich etwas anderes angegeben.**
- Jede Funktionsimplementierung in einer Header-Datei (mit Ausnahme von Funktionsschablonen) bedeutet 0 für die Übung.
- Sie sollten in der Lage sein, jeden Ihrer Header unabhängig von anderen zu verwenden. Daher müssen sie alle Abhängigkeiten einschließen, die sie benötigen. Allerdings müssen Sie das Problem der doppelten Einbindung vermeiden, indem Sie **Include-Guards** hinzufügen. Andernfalls wird Ihre Note 0 sein.

Lies mich

- Sie können bei Bedarf zusätzliche Dateien hinzufügen (z. B. um Ihren Code aufzuteilen). Da diese Aufgaben nicht von einem Programm überprüft werden, können Sie dies gerne tun, solange Sie die vorgeschriebenen Dateien einreichen.
- Manchmal sehen die Richtlinien einer Übung kurz aus, aber die Beispiele können Anforderungen aufzeigen, die nicht ausdrücklich in den Anweisungen stehen.
- Lesen Sie jedes Modul vollständig durch, bevor Sie beginnen! Wirklich, tun Sie es.
- Bei Odin, bei Thor! Benutze deinen Verstand!!!




Sie werden eine Menge Klassen implementieren müssen. Das kann mühsam sein, es sei denn, Sie können in Ihrem Lieblings-Texteditor Skripte schreiben.



Sie haben einen gewissen Spielraum bei der Durchführung der Übungen. Halten Sie sich jedoch an die vorgeschriebenen Regeln und seien Sie nicht faul. Sie würden eine Menge nützlicher Informationen verpassen! Zögern Sie nicht, sich über theoretische Konzepte zu informieren.

Kapitel III

Übung 00: Megaphon

	Übung : 00
Megaphon	
Turn-in-Verzeichnis :	
Erreichende Dateien: Makefile,	
megaphone.cpp	
Verbotene Funktionen : Keine	

Um sicher zu gehen, dass alle wach sind, schreiben Sie ein Programm, das sich wie folgt verhält:


```
$>./Megaphon "shhhhh... Ich glaube, die Schüler schlafen..."
SHHHHH... ICH GLAUBE, DIE SCHÜLER SCHLAFEN...
$>./megaphone Verdammt " ! " "Tut mir leid, Schüler, ich dachte, das Ding wäre
aus." DAMNIT ! TUT MIR LEID STUDENTEN, ICH DACHTE, DAS DING WÄRE AUS.
$>./megaphone
* LAUTE UND UNERTRÄGLICHE RÜCKKOPPLUNGSGERÄUSCHE *
$>
```



Lösen Sie die Aufgaben in der Form von C++.

Kapitel IV

Übung 01: Mein geniales Telefonbuch

	Übung : 01
Mein geniales Telefonbuch	
Turn-in-Verzeichnis : <i>ex01/</i>	
Einzureichende Dateien: Makefile, *.cpp, *.{h, hpp}	
Verbotene Funktionen : Keine	

Willkommen in den 80er Jahren und ihrer unglaublichen Technologie! Schreiben Sie ein Programm, das sich wie eine beschissene, tolle Telefonbuchsoftware verhält.

Sie müssen zwei Klassen implementieren:

- **Telefonbuch**

- Sie verfügt über eine Reihe von Kontakten.
- Es können maximal **8 Kontakte** gespeichert werden. Wenn der Benutzer versucht, einen 9. Kontakt hinzuzufügen, wird der älteste Kontakt durch den neuen ersetzt.
- Bitte beachten Sie, dass eine dynamische Zuweisung nicht zulässig ist.

- **Kontakt**

- Steht für einen Telefonbuchkontakt.

In Ihrem Code muss das Telefonbuch als eine Instanz der PhoneBook-Klasse instanziiert werden. Dasselbe gilt für die Kontakte. Jeder von ihnen muss als eine Instanz der Klasse **Contact** instanziiert werden. Es steht Ihnen frei, die Klassen so zu gestalten, wie Sie wollen, aber denken Sie daran, dass alles, was immer innerhalb einer Klasse verwendet wird, privat ist, und dass alles, was außerhalb einer Klasse verwendet werden kann, öffentlich ist.



Vergessen Sie nicht, sich die Intranet-Videos anzusehen.

Beim Programmstart ist das Telefonbuch leer und der Benutzer wird aufgefordert, einen von drei Befehlen einzugeben. Das Programm akzeptiert nur die Befehle ADD, SEARCH und EXIT.

- **ADD:** einen neuen Kontakt speichern
 - Wenn der Benutzer diesen Befehl eingibt, wird er aufgefordert, die Informationen des neuen Kontakts in ein Feld nach dem anderen einzugeben. Sobald alle Felder ausgefüllt sind, fügen Sie den Kontakt zum Telefonbuch hinzu.
 - Die Kontaktfelder sind: Vorname, Nachname, Spitzname, Telefonnummer und das dunkelste Geheimnis. Ein gespeicherter Kontakt kann keine leeren Felder haben.
- **SUCHEN:** einen bestimmten Kontakt anzeigen
 - Zeigen Sie die gespeicherten Kontakte als eine Liste mit **4 Spalten** an: Index, Vorname, Nachname und Spitzname.
 - Jede Spalte muss **10 Zeichen** lang sein. Sie werden durch ein Pipe-Zeichen (|) voneinander getrennt. Der Text muss rechtsbündig sein. Ist der Text länger als die Spalte, muss er abgeschnitten und das letzte darstellbare Zeichen durch einen Punkt (.) ersetzt werden.
 - Fordern Sie den Benutzer dann erneut auf, den Index des anzuzeigenden Eintrags einzugeben. Wenn der Index außerhalb des Bereichs liegt oder falsch ist, definieren Sie ein entsprechendes Verhalten. Andernfalls zeigen Sie die Kontaktinformationen an, ein Feld pro Zeile.
- **EXIT**
 - Das Programm wird beendet und die Kontakte sind für immer verloren!
- **Alle anderen Eingaben werden verworfen.**

Sobald ein Befehl korrekt ausgeführt wurde, wartet das Programm auf einen weiteren. Es hält an, wenn der Benutzer EXIT eingibt.


Vergeben Sie einen aussagekräftigen Namen für Ihre ausführbare Datei.



<http://www.cplusplus.com/reference/string/string/> und natürlich
<http://www.cplusplus.com/reference/iomanip/>

Kapitel V

Übung 02: Der Job Ihrer Träume

	Übung : 02
Der Job Ihrer Träume	
Turn-in-Verzeichnis : <i>ex02/</i>	
Einzureichende Dateien: <i>Makefile, Konto.cpp, Konto.hpp, tests.cpp</i>	
Verbotene Funktionen : Keine	



Account.hpp, tests.cpp und die Protokolldatei stehen auf der Intranetseite des Moduls zum Download bereit.

Heute ist Ihr erster Tag bei *GlobalBanksters United*. Nachdem Sie die Einstellungstests erfolgreich bestanden haben (dank einiger *Microsoft* Office-Tricks, die Ihnen ein Freund gezeigt hat), wurden Sie in das Entwicklungsteam aufgenommen. Sie wissen auch, dass der Personalverantwortliche erstaunt war, wie schnell Sie *Adobe Reader* installiert haben. Dieses kleine Extra hat den Unterschied ausgemacht und Ihnen geholfen, all Ihre Gegner (aka die anderen Bewerber) zu besiegen: Sie haben es geschafft!

Wie auch immer, Ihr Vorgesetzter hat Ihnen soeben eine Aufgabe erteilt, die Sie erledigen müssen. Ihre erste Aufgabe besteht darin, eine verlorene Datei wiederherzustellen. Es ist etwas schief gelaufen und eine Quelldatei wurde versehentlich gelöscht. Leider wissen Ihre Kollegen nicht, was Git ist, und verwenden USB-Sticks, um Code auszutauschen. An diesem Punkt wäre es sinnvoll, den Ort sofort zu verlassen. Sie beschließen jedoch zu bleiben. Herausforderung angenommen!

Ihre Entwicklerkollegen geben Ihnen einen Haufen von Dateien. Beim Kompilieren von *tests.cpp* stellt sich heraus, dass die fehlende Datei *Account.cpp* ist. Glücklicherweise wurde die Header-Datei *Account.hpp* gespeichert. Außerdem gibt es eine Protokolldatei. Vielleicht können Sie sie verwenden, um zu verstehen, wie die Klasse **Account** implementiert wurde.

Sie beginnen damit, die Datei Account.cpp neu zu erstellen. In nur wenigen Minuten programmieren Sie ein paar Zeilen reines C++. Nach ein paar fehlgeschlagenen Kompilierungen besteht Ihr Programm die Tests. Seine Ausgabe stimmt perfekt mit der in der Protokolldatei gespeicherten überein (**abgesehen von den Zeitstempeln**, die sich natürlich unterscheiden, da die in der Protokolldatei gespeicherten Tests vor Ihrer Einstellung durchgeführt wurden).

Verdammt, du bist beeindruckend!



Die Reihenfolge, in der die Destruktoren aufgerufen werden, kann je nach Compiler/Betriebssystem unterschiedlich sein. Ihre Destruktoren können also in umgekehrter Reihenfolge aufgerufen werden.



Sie können dieses Modul bestehen, ohne die Übung 02 zu bearbeiten.

Kapitel VI

Einreichung und Peer-Evaluierung

Reichen Sie Ihre Arbeit wie gewohnt in Ihrem Git-Repository ein. Nur die Arbeit in Ihrem Repository wird während der Verteidigung bewertet. Zögern Sie nicht, die Namen Ihrer Dateien zu überprüfen, um sicherzustellen, dass sie korrekt sind.



????????????? XXXXXXXXXXXX = \$3\$f15bc138aca1e76ec6f4cfd0797ec037