

Modulare Programmierung und Bibliotheken

Übungsaufgaben (Alexander Droste)

Aufgabe 1

- a) Erzeugen sie die Dateien **klient.c**, **anbieter.c** sowie **anbieter.h**. Implementieren sie drei unterschiedliche einfache Funktionen in **anbieter.c**. Machen sie alle Funktionen innerhalb von **anbieter.c** durch Funktionsprototypen (Funktionsdeklarationen) im Anschluss an ihre **includes** verfügbar. Testen sie, ob sich in jeder Funktion auf alle drei Funktionen zugreifen lässt.
- b) Deklarieren sie zwei dieser Funktionen im **Header** **anbieter.h** und machen sie diese in **client.c** durch eine **include**-Anweisung verfügbar. Greifen sie nun mit Hilfe des **extern**-Modifiers in **client.c** auch auf die Funktion zu, die nicht in der Schnittstelle von **anbieter.h** enthalten ist. Schützen sie anschließend diese Funktion durch **static** in **anbieter.c** so, dass diese in **client.c** nicht mehr verwendet werden kann.
- c) Wiederholen sie b) aber verwenden sie diesmal Variablen statt Funktionen.
- d) Implementieren sie zwei weitere Funktionen in **anbieter.c** die für die Eingabe von zwei Werten das gleiche Ergebnis bei unterschiedlicher Implementation liefern. Schützen sie diese durch **static**. Implementieren sie eine dritte Funktion, welche auf Abfrage einer boolschen Variable eine der beiden Rechenfunktion verwendet und machen sie diese in der Schnittstelle verfügbar. Verwenden sie die Funktion in **anbieter.c**. Ändern sie den Wert der boolschen Variable und rufen sie die Funktion erneut auf. Verifizieren sie, dass die verschiedenen Funktionen verwendet werden.

Aufgabe 2

- a) Suchen sie in ihrem Dateisystem nach den Ordnern die
- 1) Libraries enthalten welche vom Betriebssystem benötigt werden
 - 2) Libraries/Header von installierten Programmen oder selbst hinzugefügte enthalten

[schauen sie evtl. zur Hilfe in die Präsentationsfolien oder auf Wikipedia unter [Filesystem Hierarchy Standard](#) um Anhaltspunkte zu bekommen, wo sich diese befinden können]

- b) Installieren sie mit Hilfe eines Package-Managers eine Library, die sich noch nicht auf ihrem System befindet. Nutzen sie testweise einige Funktionen der Library und compilieren sie den Code um sicherzustellen, dass die Library korrekt verlinkt ist. Geben sie beim Kompilieren die entsprechenden Flags an.
- c) Erstellen sie selbst eine dynamische Library und machen sie die Schnittstelle durch einen Header verfügbar, damit durch **include** in ihren Quelltextdateien auf diese zugegriffen werden könnte. Überlegen sie, wo sich Header und Librarydatei befinden müssen, damit beim **include** kein vollqualifizierter Pfad angegeben werden muss.