



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik/Numerik

Fachbereich C
Mathematik und Naturwissenschaften,
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de

WWW www.math.uni-wuppertal.de/~buhl

DATUM 24. November 2014

generische Programmierung

WS 2014/2015 – Übungsblatt 5

Ausgabe: 24. November 2014

Abgabe bis 1. Dezember 2014 an: <mailto:125319@uni-wuppertal.de>

Aufgabe 1. *Name mangling*

Im Artikel „Calling conventions for different C++ compilers and operating systems“ (http://www.agner.org/optimize/calling_conventions.pdf) wird das „name mangling“ der gängigen C++-Compiler beschrieben.

Wozu wird es benötigt? Schreiben Sie ein Programm mit vierfach überladener Funktion, erzeugen Sie dessen Objektdatei und sehen Sie sich die entstandenen „mangled names“ an (Hinweis: benutze `nm`).

Warum kann in C++ der Typ des Funktionsergebnisses nicht zur Unterscheidung weiterer überladener Funktionsvarianten benutzt werden?

Welche Abkürzungen werden im GNU3-Mangling für die C++-Operatoren (Tabelle 17 des obigen Artikels) benutzt?

Aufgabe 2. *überladene C++-Funktionen*

Erzeugen Sie mit Hilfe einer Template-Funktion mehrere überladene Funktionen und benutzen Sie sie in einem Testrahmenprogramm.

Betrachten Sie mit Hilfe von `nm`, welche Inkarnationen genau erzeugt wurden (benutzen Sie `c++filt` zum Demanglen).

Wie können Sie die Inkarnation weiterer Varianten der Template-Funktion erzwingen?

Aufgabe 3. *ar*

[http://man.cx/ar\(1\)/de](http://man.cx/ar(1)/de)

Mit Hilfe des Unix/Linux-Tools `ar` können Sie Sammlungen von Objektdateien (*.o) erzeugen und modifizieren:

```
ar rc libperson.a person.o
```

erzeugt zum Beispiel die statische Bibliothek `libperson.a` mit der Objektdatei `person.o` als Inhalt.

Wozu und wie wird `libperson.a` benutzt?

Schreiben Sie kurze Benutzungsanleitungen (Fallstudien) über den Umgang mit statischen Bibliotheken: Hinzufügen einer neuen Objektdatei, Ersetzen einer aktualisierten Objektdatei, Extraktion einer Objektdatei, ...

[http://de.wikipedia.org/wiki/Ar_\(Unix\)](http://de.wikipedia.org/wiki/Ar_(Unix))

beschreibt im Abschnitt Dateiformat den Aufbau von `*.a`-Archiven. Welche Einschränkungen gibt es für statische Bibliotheken wegen dieses recht primitiven Dateiformats?

Wo werden statische Bibliotheken in aktuellen Linux/Unix-Betriebssystemen noch benutzt?

Warum hat sich an den meisten Stellen das `*.so`-Format durchgesetzt?

Benutzen Sie `ar`, um die Objektdatei-Bestandteile der Bibliothek `/usr/lib/libm.a` (oder einer anderen auf Ihrem System vorhandenen statischen Bibliothek) zu untersuchen.

Schreiben Sie eine Templatefunktion `my_max(T1, T2)` und testen Sie sie.

Wie erzeugen Sie verschiedene Objektdateien mit jeweils eigenen unterschiedlichen Instanzen der Template-Funktion `my_max(T1, T2)`?

Wie fügen Sie diese zu einer statischen Bibliothek zusammen?

Aufgabe 4. *shared library*

Warum werden gemäß

<http://www.yolinux.com/TUTORIALS/LibraryArchives-StaticAndDynamic.html>

Bibliotheken benutzt? Wie unterscheidet sich die Erzeugung einer statischen von der einer shared Bibliothek? Erläutern Sie jeweils die einzelnen Schritte mit eigenen Worten.

Aufgabe 5. *Versionsnummern*

Lesen Sie

<http://www.yolinux.com/TUTORIALS/LibraryArchives-StaticAndDynamic.html>

den Abschnitt „Library Versions“ und erläutern Sie den Zweck von Bibliotheks-Versionsnummern (insbesondere im Hinblick auf die am Ende von Abschnitt 1.5.3 genannten „schmutzigen“ Tricks von Systemadministratoren).

Wie könnte ein sauberes Modulsystem den unsauberen Workarounds vorbeugen?