



Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2010/2011 – Übungsblatt 3

8. November 2010

Abgabe: bis 15. November 2010 an c.seepold@uni-wuppertal.de

Aufgabe 1. *Object file*

Lesen Sie: <http://docs.hp.com/en/B2355-90655/ch02s03.html>

Was ist ein Object-File? Weshalb wird es benötigt? Was enthält es zu diesem Zweck, was nicht?

Wofür steht das Akronym ELF? Wie unterscheiden sich „executable“, „relocatable“ und „shared object“ ELF-Dateien voneinander?

Wie kann man die Symbole einer Objektdatei, wie die benutzten shared Bibliotheken ermitteln?

Aufgabe 2. *BidirectionalIterator*

Diskutieren Sie den BidirectionalIterator der STL als abstrakten Datentyp: Welche Operationen sind mit welchen Eigenschaften vorhanden, ...

Warum benötigt man ihn für den generischen Algorithmus `reverse()`?

Was steckt hinter dem Requirement EqualityComparable? Welcher generische Algorithmus benötigt einen dieses Requirement erfüllenden Templateparameter?

Aufgabe 3. *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 4. *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?