BERGISCHE UNIVERSITÄT
WUPPERTAL
GAUSS-STRASSE 20
42119 WUPPERTAL
TELEFAX (0202) 439-2901
TELEFON (0202) 439-0
WWW www.uni-wuppertal.de

Fachbereich C

MATHEMATIK UND NATURWISSENSCHAFTEN
Fachgruppe Mathematik und Informatik

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

E-Mail: Hans-Juergen.Buhl@math.uni-wuppertal.de
WWW: http://www.math.uni-wuppertal.de/∼buhl

Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2008 /2009 – Übungsblatt 9

7. Januar 2009
Abgabe: bis 14. Januar 2009 an
gcaltavu@studs.math.uni-wuppertal.de

Aufgabe 1. *Compile time assert*

Benutzen Sie hier und auch für alle kommenden Übungsaufgaben `conceptg++`.

Demonstrieren Sie die neue C++-Zusicherung `static_assert()` in einem Beispielprogramm, das nur compiliert werden kann, wenn Ihre C++-Sprachumgebung einen 4 Byte langen `int`-Typ gesitzt. Benutzen Sie dabei die in C++ zur Übersetzungszeit ausgewertete Funktion `sizeof()`.

Wann wird die `sizeof()`-Funktion in C99 ausgewertet?

Aufgabe 2. *Vorbedingungen in Templates*

Demonstrieren Sie die Benutzung von `static_assert()`, um die Template-Metafunktion `fact` (Aufgabe 1 von Übungsblatt 7) vor dem Aufruf mit einem negativen Templateparameter-Wert zu schützen. Testen Sie!

Warum nennt man `fact` in diesem Zusammenhang eine Metafunktion und nicht einfach eine Funktion?

Aufgabe 3. *eingeschränkte Generizität*

Modifizieren Sie die Lösung `geomMittel2` (Aufgabe 1 von Übungsblatt 2) zu einem eingeschränkt generischen Template (Ersetzen von `typename T` durch einen neuen eingeschränkten Typ-Laufbereich (welchen?) und testen Sie den neuen generischen Algorithmus.

Begründen Sie Ihre Wahl des Laufbereichs.

Aufgabe 4. *eingeschränkte Generizität und requires-Bedingungen*

Modifizieren Sie die Lösung `geomMittel` (Aufgabe 3 von Übungsblatt 5) zu einem eingeschränkt generischen Template (Ersetzen von `class T` durch einen neuen eingeschränkten Typ-Laufbereich (welchen?)) und testen Sie den neuen generischen Algorithmus.

Fügen Sie dann geeignete `requires`-Bedingungen hinzu, um `class InputIterator` geeignet einzuschränken.

Testen Sie! Welche Vorteile hat eine solche eingeschränkte Generizität für den Autor generischer Algorithmen, welche für den Benutzer von generischen Algorithmen?