



Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2010/2011 – Übungsblatt 8

13. Dezember 2010

Abgabe: bis 20. Dezember 2010 an c.seepold@uni-wuppertal.de

Aufgabe 1. *Template binary*

Testen Sie:

```
template <unsigned long N>
struct binary
{
    static unsigned const value
        = binary<N/10>::value * 2 // prepend higher bits
        + N%10; // to lowest bit
};

template <> // specialization
struct binary<0> // terminates recursion
{
    static unsigned const value = 0;
};
```

Was wird hier berechnet? Lassen Sie eine Tabelle berechneter Werte ausdrucken. Welche ähnlichen Anwendungen von Templates erscheinen Ihnen nützlich?

Aufgabe 2. *Metaprogrammierung*

Lesen Sie

<http://divyepakoor.blogspot.com/2008/07/walking-through-your-first-template.html>

und testen Sie das Fibonacci-Beispiel.

Welche Einsatzgebiete sieht der Autor für Metaprogrammierung?

Welche Einsatzgebiete sehen Sie?

Aufgabe 3. *boost/concept_check.hpp*

Lesen Sie den Inhalt der Datei „boost/concept_check.hpp“.

Was unterscheidet das Konzept `Comparable` vom Konzept `SGIComparable`? Wie unterscheiden sich `RandomAccessContainer` und `Mutable_RandomAccessContainer` voneinander?

Erstellen Sie eine Übersicht zu den garantiert vorhandenen Operationen aller Container gemäß der „Container Checking Classes“.

Aufgabe 4. *CONCEPT_CHECK_REQUIRES()*

Ergänzen Sie die generischen Funktionen `my_min(-)`, `arith_average(-,-)` und `geomMittel2(-,-)` der letzten Übungsblätter um die Überprüfung geeigneter gewählter Konzepte.

Provozieren Sie Konzeptverletzungen bei der Template-Instanziierung: Welche Fehlermeldungen werden erzeugt?