



Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2010/2011 – Übungsblatt 10

10. Januar 2011

Abgabe: bis 17. Januar 2011 an c.seepold@uni-wuppertal.de

Aufgabe 1. *Vorbedingungen in Templates*

Demonstrieren Sie die Benutzung von `static_assert()`, um die Template-Metafunktion `fact` (Aufgabe 3 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 2. *Vorbedingungen in Templates: Fortsetzung*

Demonstrieren Sie die Benutzung von `static_assert()`, um die Template-Metafunktion `template<int n, int m> struct cpower::result` gegen unsinnige Parameter abzusichern.

Aufgabe 3. *Vorbedingungen in Templates: Fortsetzung 2*

Demonstrieren Sie die Benutzung von `static_assert()`, um die Template-Metafunktion `template <unsigned long N> struct binary::value` gegen einen unsinnigen Parameter abzusichern.

Aufgabe 4. Lesen Sie

<http://accu.org/index.php/journals/424>

und referieren Sie die Anwendung der Metaprogrammierung für das N-Queens-Problem.

Testen Sie das Programm selbst aus. Was ist das Hauptproblem des Einsatzes von Compiletime-Metaprogrammen?