



Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2008 /2009 – Übungsblatt 1

29. Oktober 2008

Abgabe: bis 5. November 2008 an
gcaltavu@studs.math.uni-wuppertal.de

Aufgabe 1. *TR1 und TR2*

Was ist im Zusammenhang mit C++ unter TR1 und was unter TR2 zu verstehen?

Was liefert das folgende Programmsegment Neues?

```
#include <iostream>
#include <typeinfo>
#include <tr1/utility>
using std::pair; using std::make_pair;
using std::tr1::get; using std::tr1::tuple_element;
using std::tr1::tuple_size;
using std::cout;

template <class Ty>
void show(const Ty& pr)
{
    cout << "size:  " << tuple_size<Ty>::value << '\n';
    cout << "first type:  "
        << typeid(typename tuple_element<0, Ty>::type).name() << '\n';
    cout << "second type:  "
        << typeid(typename tuple_element<1, Ty>::type).name() << '\n';
    cout << "first:  " << get<0>(pr) << '\n';
    cout << "second:  " << get<1>(pr) << '\n' << '\n';
}

int main()
{
```

```
show(make_pair(1, 2));
show(make_pair(3.0, 1.1f));
return 0;
}
```

Wie werden also die TR1-Ergänzungen nutzbar? Welche weiteren Neuerungen im Container-Umfeld sind in der TR1 enthalten (http://en.wikipedia.org/wiki/Technical_Report_1)?

Aufgabe 2. *C++0x*

Welche Verbesserungen bezüglich der template-Nutzung sind im neuen C++-Standard <http://www.research.att.com/~bs/rules.pdf> geplant?

Aufgabe 3. *STL-Iteratoren*

Lesen Sie:

<http://www.mmdb.iai.uni-bonn.de/lehre/proprak0203/vortraege/rouven.ppt>

Wie unterscheiden sich die verschiedenen Iterator-Kategorien (welche sind das?) voneinander?

Aufgabe 4. *generische Programmierung*

Was versteht man nach

<http://theory.gsi.de/~vanhees/faq/gravitation/node70.html>

unter generischer Programmierung? Welche Vor-, welche Nachteile sind mit ihr verbunden?