



# Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2010/2011 – Übungsblatt 1

25. Oktober 2010

Abgabe: bis 2. November 2010 an [c.seepold@uni-wuppertal.de](mailto:c.seepold@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/C++\\_Technical\\_Report\\_1](http://en.wikipedia.org/wiki/C++_Technical_Report_1))?

Beachten Sie auch [http://www.aristeia.com/EC3E/TR1\\_info.html](http://www.aristeia.com/EC3E/TR1_info.html).

**Aufgabe 2.** *C++0x/1x*

Welche Veränderungen bezüglich der template-Nutzung sind im neuen C++-Standard

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2010/n3126.pdf>  
nach <http://en.wikipedia.org/wiki/C++0x> geplant?

**Aufgabe 3.** *STL-Iteratoren*

Wie unterscheiden sich die verschiedenen STL-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?