



Generische Programmierung (Spezielle Kapitel der praktischen Informatik)

WS 2008 /2009 – Übungsblatt 6

3. Dezember 2008

Abgabe: bis 10. Dezember 2008 an
gcaltavu@studs.math.uni-wuppertal.de

Aufgabe 1. *non-type template-Parameter*

Warum funktioniert:

```
#include <iostream>
using namespace std;

template <bool k> void print ()
{
    if(k==true) // oder auch: if(k)
        cout << "true" << endl;
    else
        cout << "false" << endl;
    return;
}

int main()
{
    print<7>();
    print<0>();

    return 0;
}
```

Wie viele und welche Inkarnationen der Templatefunktion `print()` werden automatisch erzeugt?

Welche Regeln gelten für die Typ-Transformation von non-type Funktionstemplate-Parametern über diejenigen der type Funktionstemplate-Parameter hinaus?

Ergänzen Sie in `main()` Aufrufe von
`print<8>(); print<9>(); ...`

Was erwarten Sie? Überprüfen Sie Ihre Erwartungen durch Benutzung
von `nm`.

Aufgabe 2. *Template-Funktion mit array-Parameter*

Vergleichen Sie

```
#include <iostream>
#include <numeric>

template <typename Type>
Type sum(Type *tp, size_t n)
{
    return std::accumulate(tp, tp+n, Type());
}

int main()
{
    int x[10];
    for(int l = 0; l < 10; l++)
        x[l] = l;

    std::cout << sum(x, 10) << std::endl;

    return(0);
}
```

mit

```
#include <iostream>
#include <numeric>

template <typename Type, size_t n>
Type sum(const Type (&tp)[n])
{
    return std::accumulate(tp, tp+n, Type());
}

int main()
{
    int x[10];
    for(int l = 0; l < 10; l++)
        x[l] = l;

    std::cout << sum(x) << std::endl;
}
```

```
    return(0);  
}
```

Welche Vor- und Nachteile bietet die erste Variante gegenüber der zweiten? Schreiben Sie eine Template-Funktion `unsigned int get_dimension()`, die die Anzahl der Komponenten eines übergebenen Arrays als Funktionsergebnis liefert.

Aufgabe 3. *Draft Proposal: Dynamic Libraries in C++*

Wie sollten gemäß

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2003/n1496.html>

shared Objekte (so's) in C++ systemunabhängig standardisiert werden?

Warum wird dieser Erweiterungsvorschlag nach

<http://herbsutter.wordpress.com/2007/02/07/iso-c0x-complete-public-review-draft-in-october-2007/>

nicht in C++0x realisiert werden?

Aufgabe 4. *Draft Proposal: Modules in C++*

Welche Vorteile wären mit Modulen in C++ verbunden:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n2073.pdf>

Wie soll die Syntax aussehen? Welche Vorteile brächten Module gegenüber der momentanen Praxis (welche ist das)?

Aus welchen Gründen wurden Module noch nicht in C++0x aufgenommen (vgl.

<http://herbsutter.wordpress.com/2007/02/07/iso-c0x-complete-public-review-draft-in-october-2007/>)?