

Übungen zur Vorlesung  
"Objektorientiertes Programmieren im Wissenschaftlichen Rechnen"

Dr. O. Ippisch, Dr. C. Engwer

Abgabe am 13. 07. 2010 in der Vorlesung

---

ÜBUNG 1 FIBONNACI ZAHLEN

Der Mathematiker Leonardo Fibonacci untersuchte 1202 die Kaninchenpopulation. Er stellte das folgende Gedankenexperiment auf.



*Ein neugeborenes Hasenpaar wird in einen umzäunten Garten gesetzt. Jedes Hasenpaar erzeugt während seines Lebens jeden Monat ein weiteres Paar. Ein neugeborenes Paar wird nach einem Monat fruchtbar und bekommt somit nach zwei Monaten seine ersten Nachkommen. Es soll angenommen werden, daß die Hasen nie sterben. Wieviele Hasenpaare sind nach einem Jahr in diesem Garten?*

Die Anzahl der Hasen in dem Gehege wird durch die Fibonacci-Folge beschrieben:

$$\begin{aligned}F_0 &= 1 \\F_1 &= 1 \\F_n &= F_{n-1} + F_{n-2}\end{aligned}$$

- Programmieren sie die Folge als Template-Meta Programm.
- `Fibonacci<n>::value` soll das n-te Folgenglied liefern.
- Programmieren sie zusätzlich eine Funktion `int fibonacci(int n)`, welche das n-te Folgenglied dynamisch berechnet.
- Messen und plotten<sup>†</sup> sie die Laufzeit der Berechnung im dynamischen und im statischen Fall für wachsendes n.

8 Punkte

ÜBUNG 2 STATISCHE TESTS MIT TMP

Man möchte oft zur Compilezeit schon Eigenschaften von Templateparameters überprüfen, sei es, um eine Spezialisierung der Implementierung zu verwendenden, oder um einen Fehler zu melden, wenn bestimmte Anforderungen verletzt werden.

Schauen sie sich folgendes Template Metaprogramm an:

---

<sup>†</sup>hierzu kann man z.B. gnuplot (<http://gnuplot.org>) verwenden

```

template <class A, class B>
class Check
{
    typedef char Small;
    struct Big{char dummy[2];};
    static Small test(B*);
    static Big test(...);
public:
    enum {
        IsDerived = sizeof(test(static_cast<A*>(0))) == sizeof(Small)
    };
};

```

- Was tut folgender Code:

```

class X {};
class Y : public X {};

int main()
{
    std::cout << Check<Y, X>::IsDerived << std::endl;
    std::cout << Check<int, std::string>::IsDerived << std::endl;
    return 0;
}

```

- Erklären sie das oben gezeigte Check Konstrukt.

6 Punkte

Wie immer gilt: Kommentieren Sie Ihr Programm. Erklären Sie was Sie tuen.