



Einführung in die Informatik und Programmierung (Informatik I)

WS2000/2001 – Übungsblatt 10

10. Januar 2001
Bearbeitungstermin: 3. KW

Aufgabe 1. *Funktion long Fakultae(long), Forts., 3 Punkte*

Ändern Sie Ihre Funktion `long Fakultae(long)` (Übungsblatt 6, Aufgabe 4) durch Verwendung der in der Vorlesung besprochenen Methode (Verhinderung des Überschreitens des Wertes `LONG_MAX` bzw. des Unterschreitens des Wertes `LONG_MIN`) so ab, dass im „Fehlerfall“ eine Ausnahmebedingung (Exception) erzeugt wird.

Aufgabe 2. *rationalNumbers, Forts., 6 Punkte*

Erzeugen Sie analog geeignete Exceptions in den Operationen

```
rationalNumber operator* (const rationalNumber& z, const rationalNumber& n)
```

```
rationalNumber operator/ (const rationalNumber& z, const rationalNumber& n)
```

Ihrer Klasse `rationalNumber` (Übungsblatt 8, Aufgabe 4).

Wie können Sie diese Operatoren so verbessern, daß in möglichst wenigen Fällen eine solche Exception ausgelöst wird?

Aufgabe 3. *rationalNumbers, Forts., 4 Punkte*

Erzeugen Sie ebenfalls geeignete Exceptions in den Operationen

```
rationalNumber operator+ (const rationalNumber& z, const rationalNumber& n)
```

und

```
rationalNumber operator- (const rationalNumber& z, const rationalNumber& n)
```

Ihrer Klasse `rationalNumber` (Übungsblatt 9, Aufgabe 6).

Aufgabe 4. *Zahlkonversionen, 4 Punkte*

Schreiben Sie eine Funktion, die eine beliebige nichtnegative LONG-Zahl in Ihre Darstellung in einer beliebigen Basis b ($b \geq 2$) umrechnet und testen Sie geeignet.

Aufgabe 5. *Tabellen-Ausdruck, 3 Punkte*

Testen Sie die beiden Versionen

```
////////////////////////////////////
// Datei:   table.cc
// Version: 1.0
// Zweck:   Funktionstabelle drucken
// Autor:   Hans-Juergen Buhl
// Datum:   20.09.1998
////////////////////////////////////

#include <iostream>
#include <cmath>

using namespace std;

typedef double (*FKT)(double);

void printtable(const double start_x,
               const double delta_x,
               const int nr_rows,
               FKT f)
{
    double x(start_x);
    if (nr_rows > 0)
        for (int i = 0; i < nr_rows; i++) {
            cout << x << "      " << f(x) << endl;
            x = x + delta_x;
        }
};

int main()
{
    printtable( 0.0, 0.1, 5, sin);
}

und

////////////////////////////////////
// Datei:   table2.cc
// Version: 2.0
// Zweck:   Funktionstabelle drucken
// Autor:   Hans-Juergen Buhl
// Datum:   20.09.1998
```

```

////////////////////////////////////

#include <iostream>
#include <cmath>

using namespace std;

typedef double (*FKT)(double);

class Table {

    double start_x;
    double delta_x;
    int nr_rows;

    FKT fkt;

public:

    Table(double start, double delta, int nr, FKT f):
        start_x(start), delta_x(delta), nr_rows(nr), fkt(f) {};

    void print()
    {
        double x(start_x);
        if (nr_rows > 0)
            for (int i = 0; i < nr_rows; i++) {
                cout << x << "    " << fkt(x) << endl;
                x = x + delta_x;
            }
    };
};

int main()
{
    Table SinTable(0.0, 0.1, 5, sin);
    SinTable.print();
}

```

einer Lösung des Tabellen-Ausdruckproblems.

Diskutieren Sie jeweils Vor- und Nachteile. Für welche Version entschieden Sie sich unter welchen Umständen?

Welche Verbesserungen erscheinen Ihnen in beiden Versionen angebracht, um den systematischen Fehler der x-Stellen zu minimieren?