BERGISCHE UNIVERSITÄT
GESAMTHOCHSCHULE WUPPERTAL
GAUSS-STRASSE 20
42097 WUPPERTAL
(Korrespondenzanschrift)
42119 WUPPERTAL
(Lieferanschrift)
TELEX 8 592 262 bughw
TELEFAX (0202) 439-2901
TELEFON (0202) 439-1

Fachbereich 7

MATHEMATIK

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik / Numerik

e-mail: Juergen.Buhl@math.uni-wuppertal.de

Programming by Contract

WS 2003/2004 – Übungsblatt 3

27. November 2003
Ausgabe: 17. November 2003

Aufgabe 1. *Kindklassen*

Können Kindklassen einzelne Attribute beziehungsweise Methoden der Elternklasse nicht besitzen? Wenn ja: wie kann das realisiert werden?

Aufgabe 2. *Parkplatzproblem*

Zeigen Sie, dass das Parkplatzproblem genau eine Lösung der am Ende von Kapitel 3.1 spezifizierten Art für alle Eingabegrößen besitzt, die die Vorbedingungen erfüllen. Warum sollte der Auftraggeber der Problemlösung der Softwarelösung `Parkplatzproblem` mit den Vorbedingungen einverstanden sein? Wie könnte eine Softwarelösung aussehen, die ohne den obigen Beweis die Einhaltung des Contracts für sich in Anspruch nimmt?

Aufgabe 3. *Klasse Day*

Ergänzen Sie die Klasse Day

```
////////////////////////////////////////////
// Datei:   enum_Day5.cc
// Version: 5.0
// Zweck:   enum-Klasse, >>, cerr-Warnung
// Autor:   Hans-Juergen Buhl
// Datum:   23. Nov. 99
////////////////////////////////////////////

#include        <iostream>
#include        <sstream>
#include        <string>

using namespace std;

class Day{
private:
```

```
        enum DayType {_Montag, _Dienstag, _Mittwoch, _Donnerstag,
                      _Freitag, _Samstag, _Sonntag};

// nicht als member zugelassen; nur Konstruktoren dürfen als
//    Initialisierer von Membern in Klassen dienen!!!
//
// static const string DayTable[] = {"Montag", "Dienstag", "Mittwoch",
//                      "Donnerstag", "Freitag", "Samstag", "Sonntag"};


        static const string DayTable[7];

        DayType t;

        Day(const DayType& dt): t(dt) {};

public:

        static const Day Montag;
        static const Day Dienstag;
        static const Day Mittwoch;
        static const Day Donnerstag;
        static const Day Freitag;
        static const Day Samstag;
        static const Day Sonntag;

        Day(const Day& d = Montag): t(d.t) {};

        Day& operator++();
        const Day operator++(int);

        friend istream& operator>>(istream&, Day&);
        friend ostream& operator<<(ostream&, const Day&);

};

const Day Day::Montag(_Montag);
const Day Day::Dienstag(_Dienstag);
const Day Day::Mittwoch(_Mittwoch);
const Day Day::Donnerstag(_Donnerstag);
const Day Day::Freitag(_Freitag);
const Day Day::Samstag(_Samstag);
const Day Day::Sonntag(_Sonntag);

const string Day::DayTable[] = {"Montag", "Dienstag", "Mittwoch",
                                "Donnerstag",
                                "Freitag", "Samstag", "Sonntag"};
dvips Uebung3
Day& Day::operator++()
```

```
{
        (*this).t = (_Sonntag == t) ? _Montag : DayType((*this).t + 1);
        return *this;
}

const Day Day::operator++(int)
{
        Day old_value(*this);
        ++(*this);
        return old_value;
}

istream& operator>>(istream& is, Day& d)
{
        string s;
        is >> s;
        for (int i = 0; i < 7; i++)
                if (s == Day::DayTable[i]) {
                        d.t =  Day::DayType(i);
                        return is;
                }

        is.clear(ios_base::badbit);
        return is;
}

ostream& operator<<(ostream& os, const Day& d)
{
        os << Day::DayTable[int(d.t)];
        return os;
}

int main()
{
        Day d1;
        Day d2(Day::Sonntag);
        Day d3;

        cout << d1 << endl; cout << d2 << endl;

        d1 = Day::Montag;
        for (int i = 0; i < 15; i++)
                cout << ++d1 << endl;
        cout << endl;

        d1 = Day::Montag;
        for (int i = 0; i < 15; i++)
                cout << d1++ << endl;
        cout << endl;
```

```
        for (int i = 0; i < 5; i++) {
           cout << "Bitte einen Wochentag eingeben: ";
           {
                string s;
                getline(cin, s);

istringstream ss(s);
ss >> d2;

                if (ss.bad()) {
                   ss.clear();
                        cerr << "Eingabefehler!" << endl;

                };

// hier könnten andere Eingabemöglichkeiten bearbeitet
// werden

                { string shelp;
                ss >> shelp;
                if (shelp.length()>0)
cerr << "Zusätzlicher Zeileninput ignoriert!" << endl;
                }
           }
           cout << endl << endl << " ---" << d2 << "---" << endl;
        };
cout << "Bitte einen Wochentag eingeben: ";
        cin >> d2;
if (cin.bad()){
cerr << "Eingabefehler!" << endl;
cin.clear();
        }else
cout << "Eingabe: " << d2 << endl;

}
```

um Methoden `istWochenendtag()`, Addition einer ganzen Zahl zu einem Day-Exemplar, Subtraktion zweier Day-Exemplare und testen Sie. Welche Bedeutung könnte der genannten Addition bzw. Subtraktion zugrunde liegen? Welche Vorteile hat die Klasse `Day` gegenüber einer reinen `enum`-Lösung mit eigenen Operatoren?

Aufgabe 4. *newmat10*

Besorgen Sie sich die Bibliothek `newmat10` aus dem Internet:

http://www.robertnz.net/download

Lesen Sie die Installationsanweisung, installieren Sie sie auf einem Rechner des wmit-Clusters und bringen Sie dann mit Hilfe dieser Bibliothek ein Beispielprogramm zum Laufen.