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

Algorithmen und Datenstrukturen (Informatik III)

WS1999/2000 – Übungsblatt 5

Abgabetermin: 8. Dezember 1999

Aufgabe 1. *Der C++ Standard*

Suchen sie im C++ Standard (Draft Dezember 1996)
http://www.cygnus.com/misc/wp/dec96pub/
nach der Spezifikation des `enum`-Typs.
Was ist neu für Sie? Diskutieren Sie je einen Einsatzzweck für jede dieser neuen Varianten des `enum`-Typs.

Aufgabe 2. *Aufzählungstypen als Klasse*

Schreiben Sie nach dem Muster der Vorlesung

```
////////////////////////////////////////////////
// Datei:   enum_Day3.cc
// Version: 3.0
// Zweck:   enum-Klasse, >> mit Zurückschreiben
// Autor:   Hans-Juergen Buhl
// Datum:   23. Nov. 99
////////////////////////////////////////////////

#include        <iostream>
#include        <string>

using namespace std;

class Day{
private:
        enum DayType {_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"};

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.putback(' ');
        for (string::const_reverse_iterator i = ((const string)s).rbegin();
             i < ((const string)s).rend(); i++)
                        is.putback(*i);

        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);

        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 < 15; i++) {
                cout << "Bitte einen Wochentag eingeben: ";
                cin >> d2;
                if (cin.bad()) {
                        cin.clear();
                        cout << "Eingabefehler!" << endl;
                        { string s;
                          cin >> s;
```

```
                        cout << "Fehleingabe: " << s << endl;
                    };
            };
            cout << endl << endl << " ---" << d2 << "---" << endl;
    };

}
```

eine Klasse `Schachfiguren` mit den folgenden möglichen Werten:

`Koenig, Dame, Laeufer, Springer, Turm, Bauer`

Beachten Sie dabei, daß bei der Eingabe beziehungsweise Ausgabe statt `Koenig` die Form `König` ... gewählt werden sollte.

Testen Sie Ihre Klasse „vollständig“.

Aufgabe 3. `sstream`

Mit den Programmzeilen

```
string s;
getline(cin, s);
```

können Sie eine ganze Eingabezeile in den String `s` einlesen. Um mit in Strings stehenden Eingabezeilen analog wie mit `cin` arbeiten zu können, besteht die Möglichkeit, String-Streams (Objekte der Klasse `istringstream`) zu benutzen:

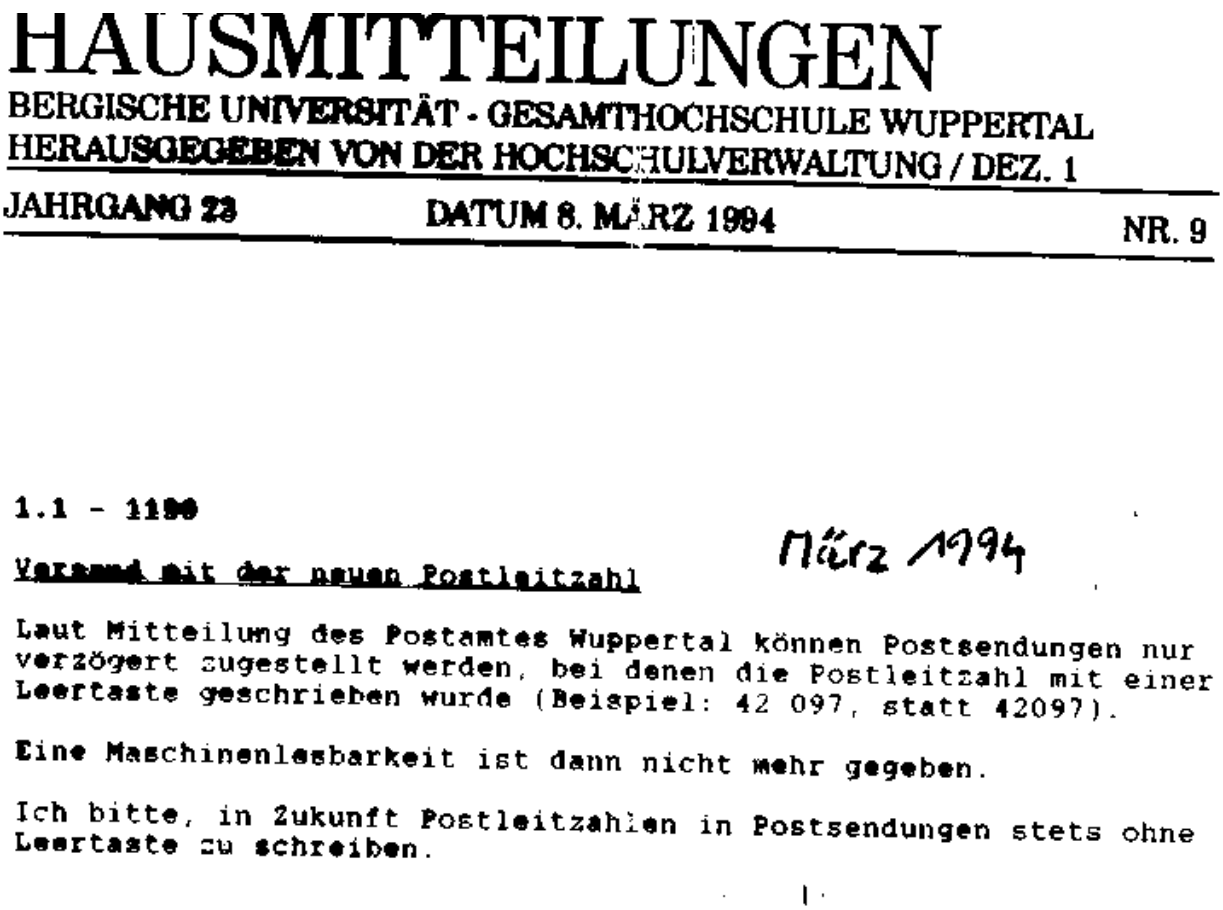
```
#include <sstream>

for (int i = 0; i < 3; i++) {
    cout << "Bitte einen Wochentag eingeben: ";
    {
         string s;
         getline(cin, s);
         s.append(" ");        // work around
         istringstream ss(s);
         ss >> d2;

         if (ss.bad()) {
             ss.clear();
             cout << "Eingabefehler!" << endl;
            { string s;
              ss >> s;
              cout << "Fehleingabe: " << s << endl;
            };
         };
    }
```

Modifizieren Sie Ihr Testrahmenprogramm `main()` von Aufgabe 2 entsprechend. Ergänzen sie dessen Ausgabe durch die zusätzliche Ausgabe einer Warnung, wenn nach dem Prompt `Bitte einen Wochentag eingeben:` mehr als ein Wort eingegeben wurde.

Neben `istringstream` gibt es auch den `ostringstream`, bei dem das zugehörige Stringattribut bei Bedarf automatisch verlängert wird. Dieses Stringattribut ist schließlich durch den Observator `str()` abfragbar.
Diskutieren Sie Einsatzmöglichkeiten beim Programm-Nutzer-Dialog und testen Sie.

Aufgabe 4. *Softwaregüte: neue Postleitzahlen*

Diskutieren Sie die Hintergründe der in der Anlage beschriebenen Probleme nach Einführung der fünfstelligen Postleitzahlen.



HAUSMITTEILUNGEN
BERGISCHE UNIVERSITÄT - GESAMTHOCHSCHULE WUPPERTAL
HERAUSGEGEBEN VON DER HOCHSCHULVERWALTUNG / DEZ. 1
JAHRGANG 23 DATUM 8. MÄRZ 1994 NR. 9

1.1 - 1190

März 1994

Versand mit der neuen Postleitzahl

Laut Mitteilung des Postamtes Wuppertal können Postsendungen nur verzögert zugestellt werden, bei denen die Postleitzahl mit einer Leertaste geschrieben wurde (Beispiel: 42 097, statt 42097).

Eine Maschinenlesbarkeit ist dann nicht mehr gegeben.

Ich bitte, in Zukunft Postleitzahlen in Postsendungen stets ohne Leertaste zu schreiben.