

```

if (!liste->IsEmpty())
{
    POSITION pos;
    for( pos = liste->GetHeadPosition(); pos != NULL; )
    {
        CPoi nt *p=(CPoi nt*)liste->GetNext( pos );
        pDC->Li neTo(*p);
    }
}

void CFc0Vi ew::OnLButtonDown(UI NT nFI ags, CPoi nt poi nt)
{
    CCl ientDC dc(thi s);

    // char S[100];
    // sprintf(S, "(%d,%d)", poi nt.x, poi nt.y);
    // dc.TextOut(poi nt.x, poi nt.y, S);

    if (!ButtonDown)
    {
        ButtonDown = TRUE;
        ol dpoi nt=poi nt;
        li ste->AddHead(new CPoi nt(poi nt));
    }
}

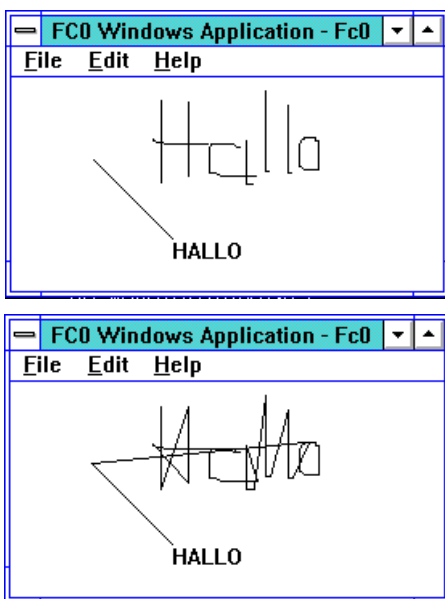
void CFc0Vi ew::OnLButtonUp(UI NT nFI ags, CPoi nt poi nt)
{
    if (ButtonDown)
    {
        ButtonDown = FALSE;
    }
}

void CFc0Vi ew::OnMouseMove(UI NT nFI ags, CPoi nt poi nt)
{
    if (ButtonDown)
    {
        CCl ientDC dc(thi s);
        dc.MoveTo(ol dpoi nt);
        dc.Li neTo(poi nt);
        ol dpoi nt=poi nt;
        li ste->AddHead(new CPoi nt(poi nt));
    }
}

```

### Zusammenfassung

Es ist nicht allzuviel des Code, den man für ein einfaches Fenster schreiben muß; man ist aber auch noch weit davon entfernt, die Zusammenhänge wirklich zu verstehen.



□

## C++-Beiträge in den PC-NEWS

**PC-NEWS 20**, S.22..25: Zeigt den Übergang von Strukturen zu Klassen an Hand eines Beispiels mit einer Struktur eines Gehaltsempfängers, bestehend aus einem Namen (String) und einem Gehalt.

**PC-NEWS 21**, S.21..23: Ableitung, Vererbung, virtuelle Funktionen, abstrakte Basisklassen

**PC-NEWS 22**, S.41: Zur Schreibweise in C, S.41..42: Friends, S.42..44: Überladen, S.44: Referenzvariable, S.45: Neues in C++, S.45..48: OOP mit C++, S.49: Wie lernt man am besten C++? S.49..51: C++ im Unterricht, Ein Beitrag zur Sprachendiskussion.

**PC-NEWS 23**, S.26..30: Materialien für den C-Unterricht, S.59..62: Entwicklung einer Klasse von Parametersubstitution über Parameterübergabe, Polymorphie, S.63..64: Erweiterungen von C in C++.

**PC-NEWS 25**, S.33..36: Eine kleine Grafikbibliothek, zeigt insbesondere zum Abschluß den Begriff Polymorphie.

**PC-NEWS 30**: S.47..64: Eine IO-Klasse für hardwarenahes Programmieren.

**PC-NEWS 35**: S.40..44. ADIM-Skripten im Vergleich, darunter auch C++-Skriptum, sowie Fehlerkorrektur dazu.

**PC-NEWS 35**: S.52..53: const und static in C und C++.

**PC-NEWS 35**: S.54..68: Hauptspeicherzugriffe mit eigenen Klassen.

### Weitere Hinweise für C-Programmierung

ADIM-Bände 40 (C), 50 (C++)

**PC-NEWS-31**, Von PASCAL zu C

Der schnelle Weg zum hohen C, PCN-SON-003

**Real programmers** programs never work right the first time. But if you throw them on the machine they can be patched into working in only a few 30-hours debugging sessions.

**Real programmers** don't use Fortran. Fortran is for wimpy engineers who wear white socks, pipe stress freaks, and crystallography weenies. They get excited over finite state analysis and nuclear reactor simulation.

**Real programmers** don't use COBOL. COBOL is for wimpy application programmers.

**Real programmers** never work 9 to 5. If any **real programmers** are around at 9 am, it's because they were up all night.

**Real programmers** don't write in BASIC. Actually, no programmers write in BASIC, after the age of 12.

**Real programmers** don't document. Documentation is for simps who can't read the listings or the object deck.

**Real programmers** don't write in Pascal, or Bliss, or Ada, or any of those pinko computer science languages. Strong typing is for people with weak memories.

**Real programmers** know better than the users what they need.

**Real programmers** think structured programming is a communist plot.

**Real programmers** don't use schedules. Schedules are for manager's toadies. Real programmers like to keep their manager in suspense.

**Real programmers** think better when playing adventure.

**Real programmers** don't use PL/I. PL/I is for insecure momma's boys who can't choose between COBOL and Fortran.

**Real programmers** don't use APL, unless the whole program can be written on one line.