

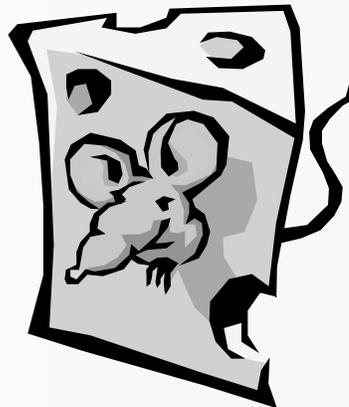
2. MFC Microsoft Foundation Classes

Vorlesung "Application Frameworks und Componentware"
Universität Trier

Sommersemester 2000

Frameworks

- Vorgefertigter Anwendungsrahmen
 - Vorgeplante Lücken ausfüllen
 - Funktionalität ggf. erweitern
 - Ableitungen
 - Neue Klassen
- Spezifisch für bestimmte Anwendungsdomäne
 - Domänenexperte waren Modellierer/Entwickler
 - Etablierte Konzepte
- Werkzeugunterstützung beim Ausfüllen der Lücken



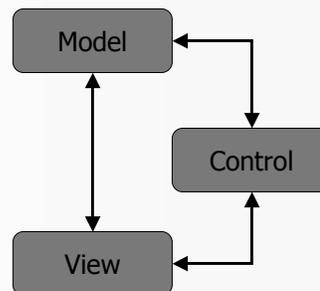
MFC

- Primäre Anwendungsdomäne: GUIs
 - Einheitliches Look and Feel
- Ursprünglich OO-Zugang WIN-API: Klassensammlung
- Mittlerweile auch Application Framework
 - Anwendungsgrundstruktur
 - GUI-Philosophie
- Stetig gewachsen: Version 6 ca. 200 Klassen
- Weitreichende Werkzeugunterstützung
 - Diverse Wizards
 - Interface Builder
- Literatur: Jeff Prosise, *Programming Windows with MFC*, 2. Auflage, MS Press, 1999

© 2000, Peter Sturm, Trier

Document/View-Architektur

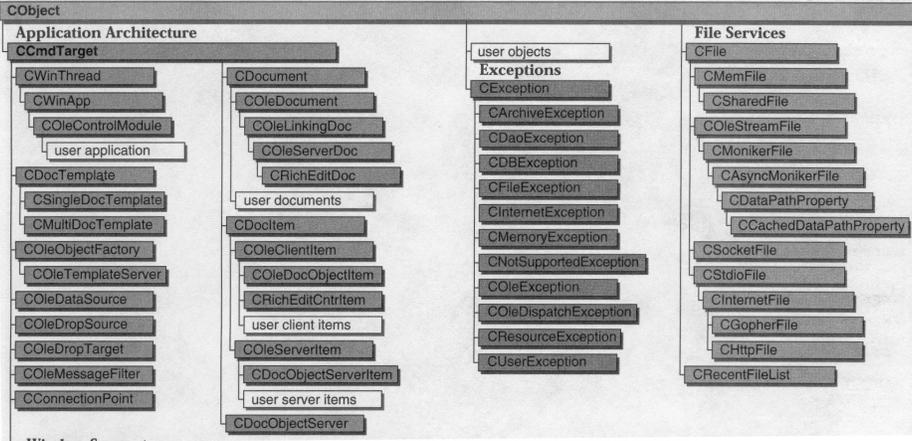
- basiert MVC-Konzept aus Smalltalk
- Trennung von
 - Inhalt (Semantik, Model)
 - Darstellung (Syntax, View)
 - Interaktion (Control)
- Vor- und Nachteile?
- Control in MFC nicht separiert
 - „Jede“ Klasse kann übernehmen
- Zwei Ansätze
 - Single Document Interface (SDI)
 - Multiple Document Interface (MDI)



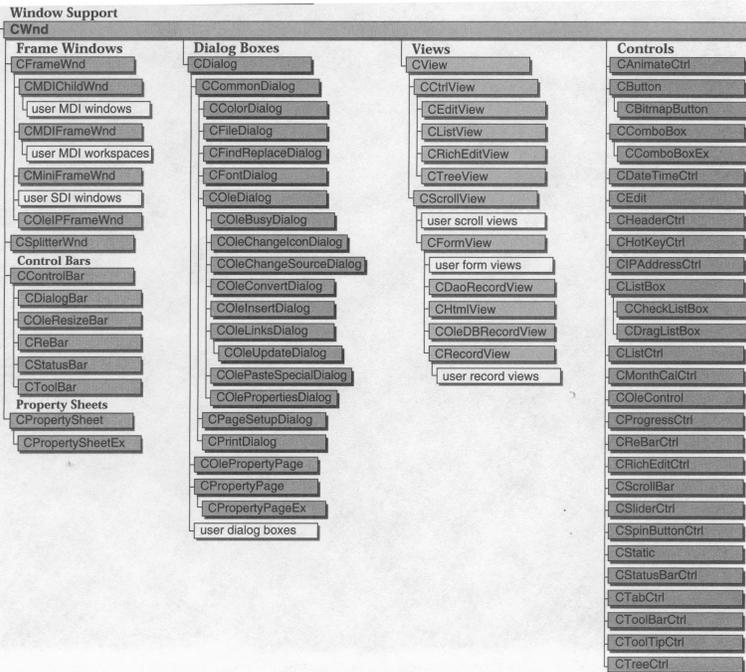
© 2000, Peter Sturm, Trier

Klassenübersicht

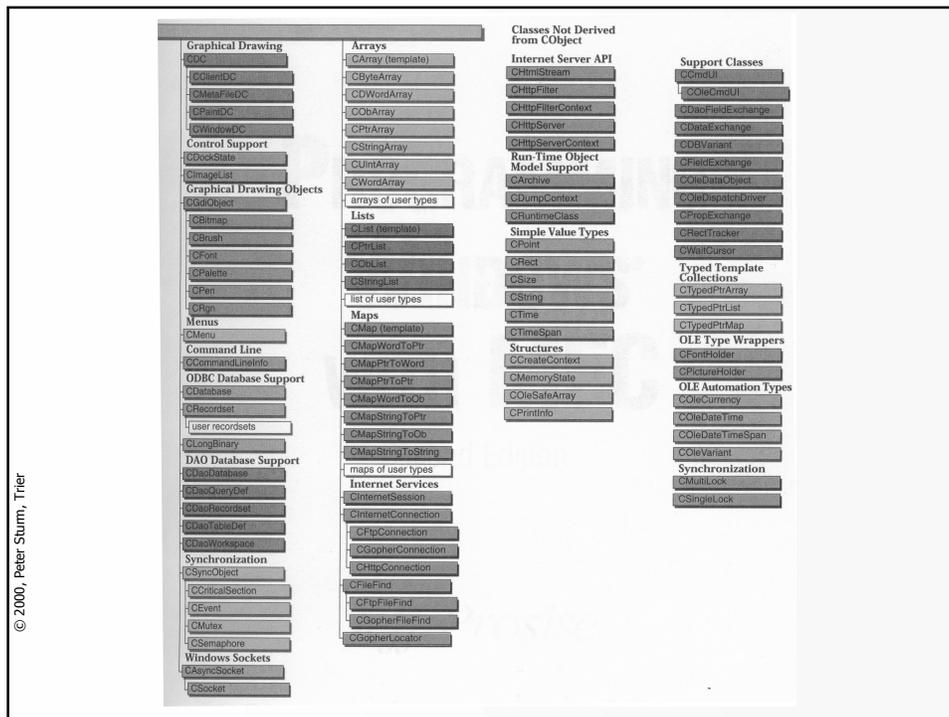
Microsoft Foundation Class Library Version 6.0



© 2000, Peter Sturm, Trier

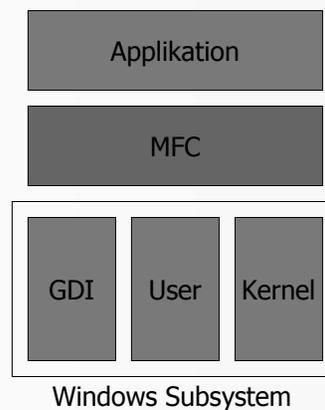


© 2000, Peter Sturm, Trier



Abhängigkeiten

- MFC setzen auf WIN32-API auf
 - GDI
 - Zeichenbefehle
 - Fonts
 - Farbverwaltung
 - User
 - Fensterverwaltung
 - Dialoge, Buttons, ...
 - Ereignisse
 - Kernel
 - Speicherverwaltung
 - Threads
 - Synchronisation
 - Dateien
 - ...



MFC: Bereich GDI

Device Context

- Device Context (DC)
 - Konzentration graphikspezifischer Zeichenattribute
- Attribute
 - Textfarbe
 - Hintergrundfarbe
 - Hintergrundmodus
 - Mapping Modus
 - Zeichenmodus
 - Aktuelle Position (Turtle Graphics :-)
 - Aktueller Stift (Pen), Pinsel (Brush), ...
 - Bitmaps, Palettes, Regions, ...
 - Aktueller Font



Zeichenmodus

- `CPaintDC dc(this)`
`dc.SetROP2(rop)`
- Mögliche Rasteroperationen rop, u.a.:
 - `R2_NOP`: dest = dest
 - `R2_NOT`: dest = not dest
 - `R2_BLACK`: dest = BLACK
 - `R2_COPYPEN`: dest = src
 - `R2_MERGEPENNOT`: dest = (NOT dest) or src
 - `R2_MERGEPEN`: dest = dest OR src
 - `R2_MASKPEN`: dest = dest AND src
 - ...
 - `R2_XORPEN`: dest = dest XOR src
- Was macht man bei Farben?

© 2000, Peter Sturm, Trier

Mapping Modus

- Umwandlung Logische Koordination in Gerätekoordinaten
- Modi
 - `MM_TEXT`: Logische Einheit = 1 Pixel
 - `MM_LOMETRIC`: Logische Einheit = 0.1 mm
 - `MM_HIMETRIC`: 0.01 mm
 - `MM_LOENGLISH`: 0.01 inch
 - `MM_HIENGLISH`: 0.001 inch
 - `MM_TWIPS`: 1/1440 inch
 - `MM_ISOTROPIC`: User Defined (X- und Y-Skalierung gleich)
 - `MM_ANISOTROPIC`: User Defined (Skalierungen beliebig)
- Richtung der X- und Y-Achsen

© 2000, Peter Sturm, Trier

Zeichenfunktionen

- Grundfunktionen
 - Aktuelle Position verändern (**MoveTo**)
 - Linien (**LineTo**)
 - Polygonzüge (**Polyline**, **PolylineTo**)
 - Kreisbögen (**Arc**, **ArcTo**)
 - Bezierkurven (**PolyBezier**, **PolyBezierTo**)
 - Kombinationen aus Linien und Bezierkurven (**PolyDraw**)
- Geschlossene Objekte
 - Chord (Schnitt Ellipse und Halbeebene)
 - Ellipse
 - Tortenstück (Pie)
 - Polygon
 - Rechteck (mit und ohne runde Ecken)

© 2000, Peter Sturm, Trier

Stifte

- Stiftdicke
- Stiftstil
 - Durchgehender Stift (**PS_SOLID**)
 - Gestrichelt (**PS_DASH**)
 - Gepunktet (**PS_DOT**)
 - Keine Linie (**PS_DOT**)
 - Varianten: **PS_DASHDOT**, **PS_DASHDOTDOT**
 - Spezial: **PS_INSIDEFRAME**
 - Linien bleiben innerhalb der Objekt-BoundingBox
- Stiftfarbe

© 2000, Peter Sturm, Trier

Pinsel

- Ausfüllen von Flächen
- Modi
 - Einheitliche Farbe
 - Muster (HatchBrush):
 - Linien
 - Kariert
 - Schraffiert
 - ...
- Ursprung
 - Wichtig beim Tiling



© 2000, Peter Sturm, Trier

Textfunktionen

- Unterstützung von ASCII- und UNICODE-Text
- Diverse Ausgabefunktionen
 - **DrawText**: Ausgabe innerhalb einer BoundingBox
 - **TextOut**: Ausgabe ab einer Position
 - **ExtTextOut**: Textausgabe inkl. Hintergrund füllen und variable Zeichenabstände
 - **TabbedTextOut**: Berücksichtigung von Tabulatoren
- Zusätzliche Funktionen:
 - Länge eines String bei gegebenem Font
 - Abfragen von Fontgrößen (Höhe, Extent, ...)
- Fontverwaltung

© 2000, Peter Sturm, Trier

Messages

- Rückmeldung von „Graphik“-Ereignissen an Anwendung
- Umfangreiche Sammlung
 - Fensterverwaltung
 - Öffnen, Schließen, Iconfy
 - Überdeckung und Sichtbarwerden von Fensterbereichen
 - Mausereignisse
 - Drücken und Loslassen von Maustasten
 - Maus betritt Fenster
 - Maus verläßt Fenster
 - Mausrad
 - Tastatur

© 2000, Peter Sturm, Trier

Quiz: Realisierung Messages

- Nahezu jede Klasse kann Messages empfangen und verarbeiten

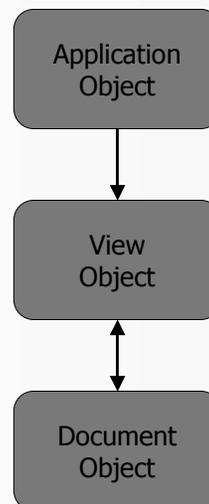
© 2000, Peter Sturm, Trier

MFC: Bereich User

Single Document Interface

Grundarchitektur

- Application Object
 - Allgemeine Verwaltungsaufgabe
- Document Object
 - Verwaltet Dokumentzustand
 - MVC-Modeler, -Control
- View Object
 - Darstellung



CDocument

- Grundfunktionen
 - Zugriff auf zugeordnete `CView`-Objekte
 - Zugriff auf Pfadnamen des Dokuments
 - Titel
 - Modifikationsstatus (`isModified`, `SetModifiedFlag`)
 - Aktualisierung (`UpdateAllViews`)
- Überschreibbare Funktionen
 - Erzeugung eines neuen Dokuments (`OnNewDocument`)
 - Öffnen eines Dokuments (`OnOpenDocument`)
 - Löschen eines Dokumentzustands (`DeleteContents`)
 - Serialisierung (`Serialize`)

© 2000, Peter Sturm, Trier

CView

- Überschreibbare Grundfunktionen
 - Initialisierung (`OnInitialUpdate`)
 - Sicht darstellen (`OnDraw`)
 - Dokumentzustand verändert (`OnUpdate`)

© 2000, Peter Sturm, Trier