

Course 2DC7, C1: Intro, Bauanleitung mit C++/MFC7.0

Copyright © by V. Miszalok, last update: 02-04-2002

- ↓ [Ein leeres Fenster](#)
- ↓ [TextOut: Hallo Welt](#)
- ↓ [Fenstergröße anzeigen und SetTextColor](#)
- ↓ [links, rechts, oben, unten](#)
- ↓ [MoveTo, LineTo, Rectangle, Ellipse](#)
- ↓ [Zeichenstift pen mit zufälligen Farben zeichnet Stern mit zufälligen Strichlängen](#)
- ↓ [Roter Brush und weißer Pen zeichnen ein Polygon](#)
- ↓ [Weitere Aufgaben](#)

Ein leeres Fenster

Microsoft Visual Studio.NET starten

File - New - Project - Project Types: Visual C++ Projects, Templates: MFC Application

Name: Intro1

Location: C:\temp

Button OK unten Mitte klicken

Es meldet sich der MFC Application Wizard - Intro1 mit der Seite Overview, die uninteressant ist.

Links unter Overview auf Application Type klicken.

Schritt1: single document einschalten

Schritt 2: Document/View architecture support Checkbox ausschalten

Schritt 3: Finish VS.NET legt nun unter C:.\temp ein Directory Intro1 an und schreibt dorthin jede Menge Files, die insgesamt das Projekt Intro1 bilden.

Klicken Sie Debug in der Menü-Zeile oberhalb des Hauptfensters

Es öffnet sich ein Untermenü. Klicken Sie auf Start Without Debugging Ctrl F5

Es erscheint eine Message Box: These project configuration(s) are out of date: Intro1 - Debug Win 32. Would you like to build them ?

Sie antworten mit Yes

Das Rumpfprogramm wird jetzt automatisch übersetzt, gelinkt und gestartet, das kann beim ersten Mal bis zwei Minuten dauern. Beobachten Sie solange das Output-Fenster am unteren Rand von VisualStudio.NET.

Unser Programm startet automatisch als eigenes Fenster, das aus drei Teilen besteht:

1. Hauptfenster = MainFrame mit Titelzeile: Intro1
2. drei funktionierende Buttons für Minimize, Maximize, Close
3. ein schmaler Rahmen, dessen 4 Seiten beweglich sind und dessen 4 Ecken beweglich sind.

Verkleinern/Vergrößern Sie den Rahmen.

Minimieren Sie VisualC++, damit Sie ein Gefühl dafür bekommen, dass Intro1.exe ein eigenständiges, autonomes Windows-Programm ist.

Starten Sie den Explorer. Gehen sie nach C:\temp\Intro1\Debug.

Doppelklicken Sie auf Intro1.exe. Sie können so beliebig viele Instanzen von Unbenannt- Intro1 starten (Diese müssen aber alle einzeln wieder beendet werden, bevor das Programm geändert wird.). Minimieren Sie den Explorer.

Beenden Sie alle Instanzen von Intro1 bis auf eine einzige.

Erproben Sie in Intro1 die drei Menüs File Edit View Help

Es funktioniert File Exit. Das Menü Edit funktioniert, enthält aber keine ausführbaren Elemente.

Das Menü View erlaubt das Ein/Ausschalten von Toolbar und Status Bar.

Das Menü Help funktioniert und zeigt den About-Dialog: About Intro1

Diese gesamte Funktionalität ist vorprogrammiert und steht jetzt zu Verfügung als Basis für Ihr erstes MFC-Programm.

Wichtig: Immer zuerst alle Instanzen von Intro1 beenden, bevor neuer Code eingegeben und übersetzt wird !

TextOut: Hallo Welt

Klicken Sie im Hauptmenu von VS.NET auf den Menüpunkt `View` und dann auf den Unterpunkt `Class View` `Ctrl+Shift+C`.

Sie sehen im rechten Bereich des VS.NET-Hauptfensters das Unterfenster `Class View - Intro1`.

An dessen unteren Rand die Registerkarte `Klassen` klicken.

Darin sehen Sie die Zeile `+ Intro1`, klicken Sie das Pluszeichen.

Sie sehen u.a. `+ CChildView`, klicken Sie das Pluszeichen vor `CChildView`.

Sie sehen u.a. `OnPaint(void)`, doppelklicken Sie diesen Funktionsnamen.

Im blauen Titelposten von VS.NET steht `Intro1 - Microsoft Visual C++ [design] - ChildView.cpp`.

Suchen Sie die Funktion `void CChildView::OnPaint(void)`.

Innerhalb der geschweiften Klammern, unterhalb der Zeile `CPaintDC dc(this);`

Anstelle der beiden Kommentarzeilen, die vorne mit `// TODO:` beginnen, schreiben Sie eine neue Zeile: `dc.TextOut(10, 10, "Hello world, here is Intro1 !");`

Die Funktion muß; nun so aussehen:

```
void CChildView::OnPaint(void)
{
    CPaintDC dc(this); //device context for painting
    dc.TextOut(10,10, "Hello world, here is Intro1 !");
}
```

Klicken Sie `Debug` in der Menü-Zeile oberhalb des Hauptfensters

Es öffnet sich ein Untermenü. Klicken Sie auf `Start Without Debugging Ctrl F5`.

Es meldet sich die bekannte Message Box, die Sie mit `Yes` verlassen. Das Programm wird wieder (jetzt viel schneller) automatisch übersetzt, gelinkt und gestartet.

Fenstergröße anzeigen und SetTextColor

Version2: Beenden Sie Ihr Programm `Intro1`.

Schreiben Sie sieben weitere Zeilen in die Funktion `void CChildView::OnPaint():`

```
CRect r; //ein Rechteck namens r vom Typ CRect
CString sometext; //eine Zeichenkette namens sometext vom Typ CString
GetClientRect( r ); //Frage an das Betriebssystem nach der Fenstergröße
sometext.Format( "width=%d, height=%d", r.right, r.bottom ); //2 Zahlen
dc.TextOut( 10, 30, sometext ); //in eine Zeichenkette konvertieren
dc.SetTextColor(RGB(255,0,0)); //rote Buchstaben
dc.TextOut(10,50, "Change the size of your window !" );
```

Klicken Sie `Debug` und dann `Start Without Debugging Ctrl F5`.

Wichtiger Hinweis: Wenn Sie sich vertippt haben, dann übersetzt der Compiler nichts, sondern meldet sich mit einer Message Box: `There were build errors. Continue ?`. Sie quittieren mit No. Am unteren Rand von VS.NET erscheint ein Fenster mit Fehlermeldungen. Scrollen Sie innerhalb dieser Fehlerliste nach oben zum ersten `error`. Wenn dort kein `error` sondern eine `warning` steht, dann ignorieren Sie die `warning` und suchen tiefer den ersten `error`. Doppelklicken Sie die Zeile mit dem ersten `error`. Der Cursor springt nun automatisch in Ihren Code auf die Zeile, wo der Fehler entdeckt wurde. Suchen Sie dort den Fehler und beseitigen Sie ihn. Ignorieren Sie alle weiteren Fehler (das sind meistens Folgefehler des ersten Fehlers) und übersetzen Sie neu. Vorsicht: Der Fehler kann auch in der Zeile vor der Cursorzeile sitzen (z.B. wenn Sie dort das abschließende Semikolon vergessen haben). Diesen Vorgang wiederholen Sie so oft, bis keine Fehlermeldung mehr erscheint und Ihr übersetztes und gelinktes Programm automatisch startet.

links, rechts, oben, unten

Version3: Beenden Sie Ihr Programm Intro1.

Schreiben Sie acht weitere Zeilen in die Funktion `void CChildView::OnPaint()`:

```
CPoint p; //ein Punkt namens p vom Typ CPoint
p.x = r.right / 2; // Fenstermitte x
p.y = r.bottom / 2; // Fenstermitte y
dc.SetTextColor( RGB( 0, 0, 255 ) ); //blaue Buchstaben
dc.TextOut( 0 , p.y , "left" );
dc.TextOut( r.right-50, p.y , "right" );
dc.TextOut( p.x , 0 , "top" );
dc.TextOut( p.x , r.bottom-20, "bottom" );
```

Klicken Sie Debug und dann Start Without Debugging Ctrl F5.

MoveTo, LineTo, Rectangle, Ellipse

Version4: Beenden Sie Ihr Programm Intro1.

Schreiben Sie 8 weitere Befehle in die Funktion `void CChildView::OnPaint()`:

```
dc.MoveTo( 0 , 0 );
dc.LineTo( r.right, r.bottom );
dc.MoveTo( r.right, 0 );
dc.LineTo( 0 , r.bottom );
int w5 = r.right / 5; //ein Fünftel der Breite
int h5 = r.bottom / 5; //ein Fünftel der Höhe
dc.Rectangle( w5, h5, 4*w5, 4*h5 );
dc.Ellipse ( w5, h5, 4*w5, 4*h5 );
```

Klicken Sie Debug und dann Start Without Debugging Ctrl F5.

Zeichenstift pen mit zufälligen Farben zeichnet Stern mit zufälligen Strichlängen

Version5: Beenden Sie Ihr Programm Intro1.

Schreiben Sie ganz oben in den Anfang von CChildView.cpp hinter die 3 Befehle, die mit #include beginnen folgende Zeile beginnend ganz links am Seitenrand:

```
#include "math.h" //damit sin(), cos() erkannt werden
```

Schreiben Sie vor die void CChildView::OnPaint() Funktion folgende Zeile beginnend ganz links am Seitenrand:

```
#define nn 120
```

Schreiben Sie folgende weiteren Befehle in die Funktion void CChildView::OnPaint():

```
int i;
CPen pen; //ein Zeichenstift namens pen vom Typ CPen
CPoint splash[ nn ]; //ein Array der Länge nn namens splash für x/y-Koordinaten
double arcus = 2. * 3.14159 / nn; // ein kleines Segment des Einheitskreises
double radius_x = 1.5 * w5; // Ellipsenradius waagrecht
double radius_y = 1.5 * h5; // Ellipsenradius senkrecht
for ( i = 0; i < nn; i++ )
{
    COLORREF multicolor = RGB ( rand()%255, rand()%255, rand()%255 );
    pen.CreatePen( PS_SOLID, 20, multicolor ); //Stiftdicke 20, zufällig bunt
    dc.SelectObject( pen ); //nimm den Stift in die Hand
    double factor = (double)rand() / (double)RAND_MAX; //irgendwas zwischen 0.0
und 1.0
    if ( factor < 0.25 ) factor = 0.25; //aber nicht kleiner als 1/4
    double cosinus = radius_x * factor * cos( i * arcus );
    double sinus = radius_y * factor * sin( i * arcus );
    dc.MoveTo( p ); //Mittelpunkt
    dc.LineTo( p.x + (int)cosinus, p.y + (int)sinus );//Endpunkt
    pen.DeleteObject(); //wirf den Stift weg
    splash[i].x = p.x + int( cosinus * 0.8 ); //x-Koordinate speichern
    splash[i].y = p.y + int( sinus * 0.8 ); //y-Koordinate speichern
}
}
```

Klicken Sie Debug und dann Start Without Debugging Ctrl F5.

Erproben Sie das Programm durch Ziehen am Fensterrand.

Roter Brush und weißer Pen zeichnen ein Polygon

Version6: Beenden Sie Ihr Programm Intro1.

Schreiben Sie folgende weiteren Befehle in die Funktion void CChildView::OnPaint() unterhalb die schon vorhandenen Befehle:

```
dc.SelectStockObject( WHITE_PEN ); //ein Stift der Dicke 1 namens WHITE_PEN
CBrush brush; //ein Pinsel namens brush vom Typ CBrush
brush.CreateSolidBrush( RGB( 255,0,0 ) ); //tauche ihn in rote Farbe
dc.SelectObject( brush ); //nimm ihn in die Hand
dc.Polygon( splash, nn ); //male ein Vieleck mit nn Ecken
brush.DeleteObject(); //wirf den Pinsel weg
dc.SetTextColor( RGB( 0,0,255 ) );
dc.TextOut( p.x-30, p.y-8, "Splash !" );
```

Klicken Sie Debug und dann Start Without Debugging Ctrl F5.

Erproben Sie das Programm durch Ziehen am Fensterrand.

Weitere Aufgaben

Lesen Sie die Erklärungen zu jeder Zeile im HTML-Text `C1_Code_Comment`.

Klicken Sie auf `Help` in der Menüleiste von VS.NET. Klicken Sie auf das Untermenü `Index`.

Gehen Sie in das Feld `Filtered by:` und wählen Sie dort `Visual C++`. Dann geben Sie im Feld `Look for:` folgende Schlüsselwörter ein: `#define`, `CPoint`, `rand`, `Polygon`, `Ellipse`, `Rectangle` etc.

Sie bekommen dann eine Auswahl von Stichwörtern, mit dem gleichen Anfang. Wenn es komplizierte alternative Textstellen gibt, erscheint ein zusätzliche Auswahl im Outputfenster am unteren Rand von VS.NET. Lesen Sie die Hilfetexte. Die Hilfetexte überdecken Ihren Code. Wenn Sie mit dem lesen fertig sind, entfernen Sie die Hilfetexte mit deren X-Button in deren rechter oberer Fensterecke.

Beenden Sie VS.NET, starten sie den Explorer, löschen Sie die gesamte Directory `C:\temp\Intro1`

Starten Sie VS.NET wieder und erzeugen dasselbe Programm so oft, bis Sie das Programm ohne Anleitung und schnell von Null an erstellen, verändern und bedienen können.

Benutzen Sie Drag&Drop beim Schreiben von Code = verschieben und kopieren mit Maus mit/ohne Strg-Taste.

Erfinden und erproben Sie neue Varianten des Programms (in Form von neuen Projekten `Intro2`, `Intro3` usw. nach obigem Muster), z.B. bunte Striche waagrecht parallel, senkrecht parallel, bunte Rechtecke und Ellipsen an zufälligen Stellen etc.