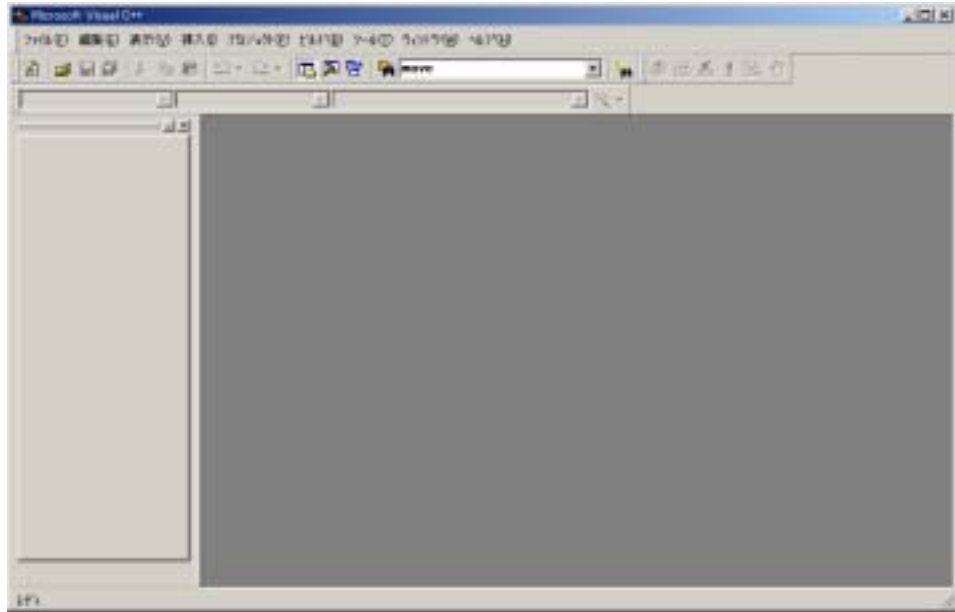


C 言語による Windows プログラミング

§ 1 . 画面レイアウトをダイアログベースで作成する

1 . 空のウィンドウの作成

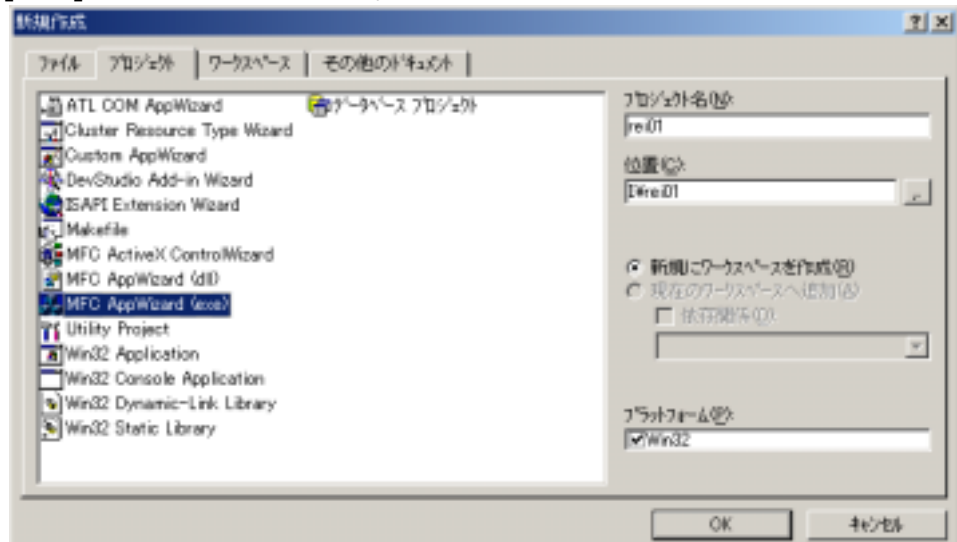
- 1) [スタート] - [プログラム] - [Microsoft Visual Studio 6.0] - [Microsoft Visual C++ 6.0] を選択して、VC++を起動する。



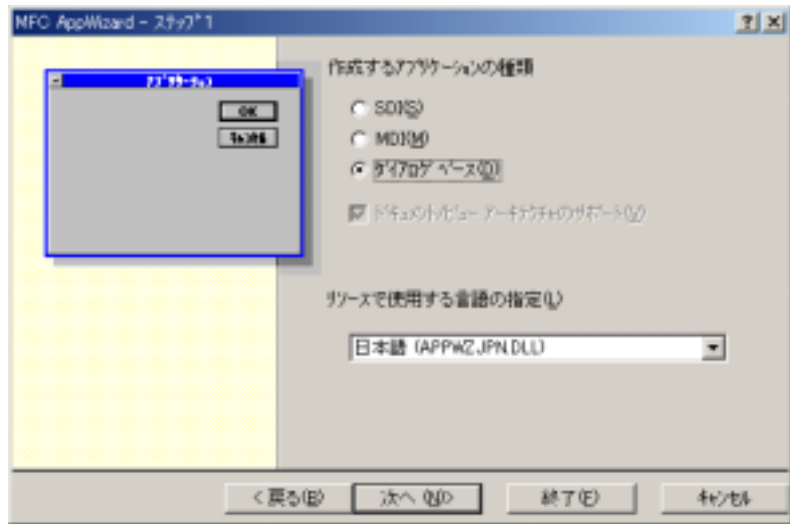
- 2)メニューから [ファイル] - [新規作成] を選択する。

「新規作成」ダイアログボックスが表示される。

- ・ MFC AppWizard(exe)を選択する。
- ・ 「プロジェクト名」欄に **rei00** と入力する。
- ・ 「位置」欄でプロジェクトのフォルダ位置を指定する。(例 . D:¥rei00)
- ・ [OK] ボタンをクリックする。



3)MFC AppWizard が起動するので、[作成するアプリケーションの種類] からダイアログベースを選択し、[終了] ボタンをクリックする。



4)新規プロジェクト情報が表示されるので、内容を確認し [OK] ボタンをクリックする。

5)rei00 と云うプロジェクトが作成されるので、[ビルド] メニューより [実行 rei00.exe] を選択しプログラムを実行する。

2 . プログラムを作成する。(画像出力)

1)画面の左側にある領域 (ワークスペース) の下側にある [File] タブをクリックする。

- ・表示された [rei00 ファイル] をダブルクリックする。
- ・さらにその下の [Source Files] をダブルクリックする。
- ・さらにその下の [rei00Dlg.cpp] をダブルクリックする。

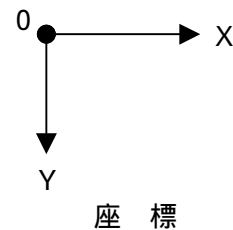
2)画面の右側にソースファイルが表示される。

3)ソースファイルの OnPaint 関数の中に実際のプログラムを追加する。

(次ページの赤字部分を挿入する。)

手順

- ・ペン、ブラシの作成
 - ペン - 線の色などを決める
 - ブラシ - 図形を塗りつぶす色などを決める
- ・描画関数
 - 直線 `MoveTo(x1, y1); LineTo(x2, y2)`
 - 四角形 `Rectangle(x1, y1, x2, y2)`
 - 楕円 `Ellipse(x1, y1, x2, y2)`
 - 文字 `TextOut(x1, y1, "文字列");`



3 . 保存

1) メニューから [ファイル] - [上書き保存] を選択する。

```

void CRei00Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 描画用のデバイス コンテキスト

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // クライアントの矩形領域内の中央
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // アイコンを描画します。
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CPaintDC dc(this);
        CPen penGreen(PS_SOLID, 1, RGB(0, 255, 0)); //緑ペンの作成
        CPen penBlue(PS_SOLID, 1, RGB(0, 0, 255)); //青ペンの作成
        CBrush brushBlue(RGB(0, 0, 255)); //青ブラシの作成
        CPen* oldPen=dc.SelectObject(&penBlue); //旧ペンの保存と青ペンの選択
        CBrush* oldBrush=dc.SelectObject(&brushBlue); //旧ブラシの保存と青ペンの選択

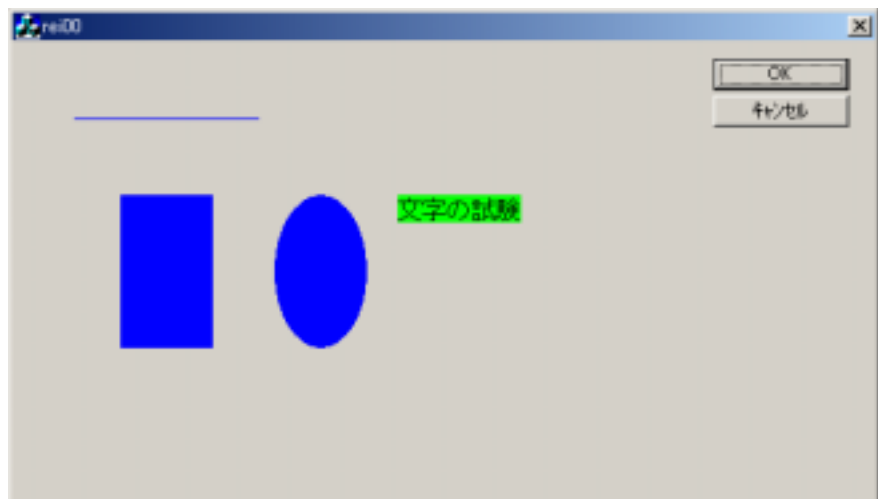
        dc.MoveTo(40, 50); //基点(40,50)の指定
        dc.LineTo(160, 50); //((160,50)まで直線を描く
        dc.Rectangle(70, 100, 130, 200); //((70,100)-(130,200)の四角を描く
        dc.Ellipse(170, 100, 230, 200); //((170,100)-(230,200)に内接する楕円を描く
        dc.SetBkColor(RGB(0, 255, 0)); //文字の背景色を緑色に指定
        dc.TextOut(250, 100, "文字の試験"); //基点(250,100)から文字描画

        dc.SelectObject(oldPen); //旧ペンに戻す
        dc.SelectObject(oldBrush); //旧ブラシ戻す

        CDialog::OnPaint();
    }
}

```

実行結果：



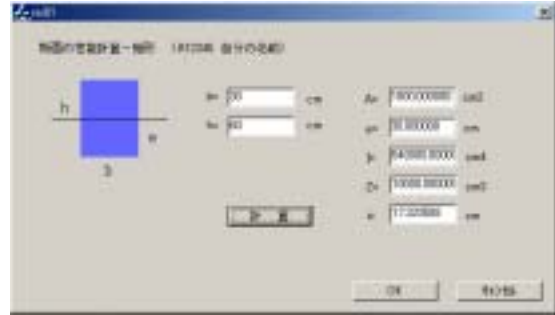
§ 2 . 矩形断面の性能計算プログラム

1 . 空のウィンドウの作成

- 1)プロジェクト名を **rei01** とする。
(前章 1 . を参照)

2 . 矩形断面の描画

- 1)ソースファイルの **OnPaint** 関数の中に次のプログラムを追加する。(赤文字の所)



```
void CRei00Dlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 描画用のデバイス コンテキスト

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // クライアントの矩形領域内の中央
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // アイコンを描画します。
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CPaintDC dc(this);
        CPen penGreen(PS_SOLID, 1, RGB(0, 255, 0)); // 緑ペンの作成
        CPen penBlue(PS_SOLID, 1, RGB(0, 0, 255)); // 青ペンの作成
        CBrush brushBlue(RGB(0, 0, 255)); // 青ブラシの作成
        CPen* oldPen=dc.SelectObject(&penBlue); // 旧ペンの保存と青ペ～
        CBrush* oldBrush=dc.SelectObject(&brushBlue); // 旧ブラシの保存と青～

        dc.Rectangle(70, 60, 130, 140);
        dc.SelectObject(oldPen);
        dc.SelectObject(oldBrush);
        dc.MoveTo(40, 100); dc.LineTo(160, 100);
        dc.SetBkColor(RGB(210, 210, 210));
        dc.TextOut(50, 80, "h");
        dc.TextOut(95, 145, "b");
        dc.TextOut(140, 110, "e");

        dc.SelectObject(oldPen); // 旧ペンに戻す
        dc.SelectObject(oldBrush); // 旧ブラシ戻す

        CDialog::OnPaint();
    }
}
```

3. コントロールの作成

1) エディットボックスの配置

- ・[コントロール] ツールバーの [エディットボックス] ボタンをクリックし、ダイアログボックス上の配置位置でクリックする。
順番は、幅(b) 高さ(h) 断面積(A) 距離(e)
断面二次モーメント(I) 断面係数(Z) 断面二次半径(i)とする。

2) スタティックテキストの配置

- ・[コントロール] ツールバーの [スタティックテキスト] ボタンをクリックし、ダイアログボックス上の配置位置でクリックする。
- ・配置したら、それぞれの文字を入力する。(選択された状態でキーボードから文字を打ち込むとプロパティが表示されキャプションで入力出来る)

3) ボタンの配置

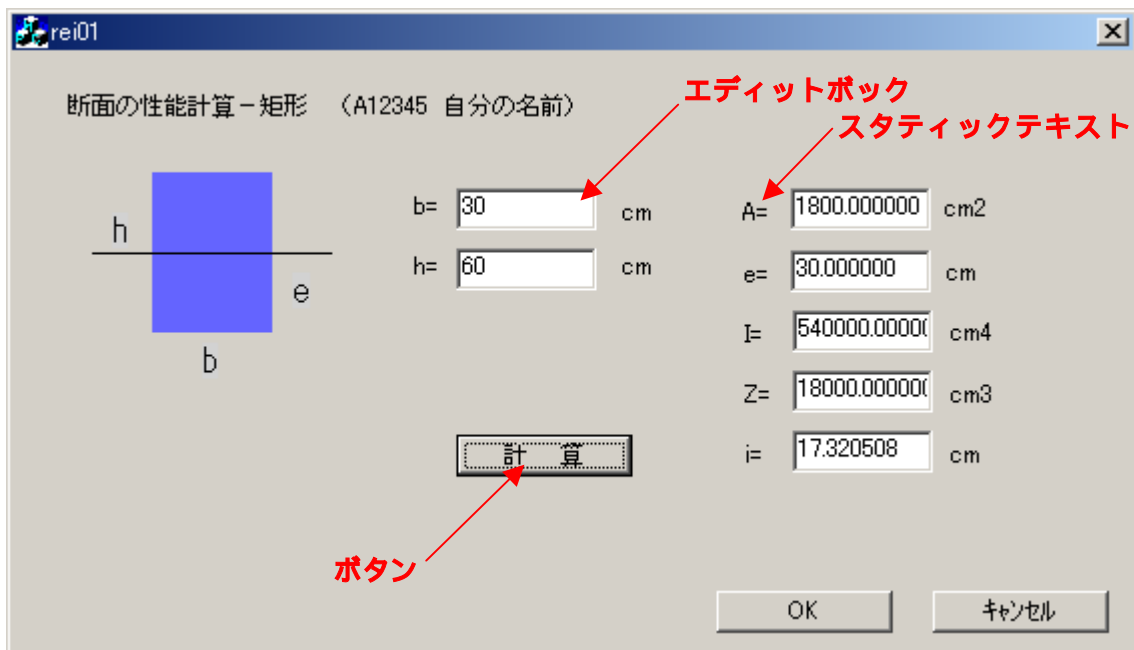
- ・[コントロール] ツールバーの [ボタン] ボタンをクリックし、ダイアログボックス上の配置位置でクリックする。
- ・配置したら、"計算" と入力する。(選択された状態でキーボードから文字を打ち込むとプロパティが表示されキャプションで入力出来る)

エディットボックス
スタティックテキスト

ボタン



配置された各コントロールは、その上で右クリックするとメニューが表示される。プロパティを選択する事により、キャプションなどから設定した文字を変更できる。



4 . 計算式の作成、値の表示

1)ダイアログボックス上の [計算ボタン] をダブルクリックする。

[メンバ関数の追加] ダイアログが開くので [OK] ボタンをクリックする。

2)ソースファイルが OnButton1 関数の位置で開くので、関数の中に次のプログラムを追加する。(赤文字の所)

```
void CRei01Dlg::OnButton1()
{
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
    double b, h, A, e, I, Z, i;
    CString s;
    CEdit *p;

    // 入力
    // b=
    p=(CEdit *)GetDlgItem(IDC_EDIT1);
    p->GetWindowText(s);
    b=atof(s);
    // h=
    p=(CEdit *)GetDlgItem(IDC_EDIT2);
    p->GetWindowText(s);
    h=atof(s);

    // 計算
    A=b*h;           //断面積
    e=h/2.0;        //距離
    I=b*h*h*h/12.0; //断面二次モーメント
    Z=I/e;          //断面係数
    i=sqrt(I/A);    //断面二次半径

    // 出力
    // A
    p=(CEdit *)GetDlgItem(IDC_EDIT3);
    s.Format("%f", A);
    p->SetWindowText(s);
    // e
    p=(CEdit *)GetDlgItem(IDC_EDIT4);
    s.Format("%f", e);
    p->SetWindowText(s);
    // I
    p=(CEdit *)GetDlgItem(IDC_EDIT5);
    s.Format("%f", I);
    p->SetWindowText(s);
    // Z
    p=(CEdit *)GetDlgItem(IDC_EDIT6);
    s.Format("%f", Z);
    p->SetWindowText(s);
    // i
    p=(CEdit *)GetDlgItem(IDC_EDIT7);
    s.Format("%f", i);
    p->SetWindowText(s);
}
```

3)インクルードファイルの追加

- ・ソースファイル(rei01Dlg.cpp)の先頭付近に以下のインクルードファイルを追加する。
(赤文字の所)

```
// rei01Dlg.cpp : インプリメンテーション ファイル
//

#include "stdafx.h"
#include "rei01.h"
#include "rei01Dlg.h"
#include <math.h>
#include <stdlib.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

5 . プログラムの実行

- ・[ビルド]メニューより[実行 rei01.exe]を選択しプログラムを実行する。
- ・実際の実行ファイルは、rei01 フォルダの中の Debug フォルダに rei01.exe として作成される。

§ 3 . 演習

1 . 円断面の性能計算プログラムを作成する。

(ヒント)

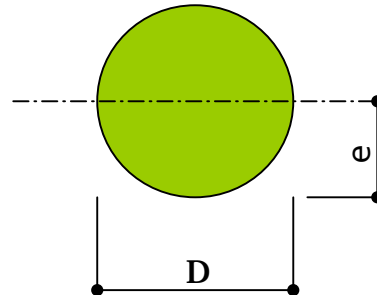
$$A = D^2 / 4$$

$$e = D / 2$$

$$I = D^4 / 64$$

$$Z = D^3 / 32$$

$$i = \sqrt{I / A}$$



2 . BOX断面の性能計算プログラムを作成する。

(ヒント)

$$A = b h - (b - 2 t)(h - 2 t)$$

$$e = h / 2$$

$$I = \{ b h^3 - (b - 2 t)(h - 2 t)^3 \} / 12$$

$$Z = I / e$$

$$i = \sqrt{I / A}$$

