

# NCAS システム開発マニュアル

## 目次

1. クラス一覧
2. クラスのメンバー変数およびメンバー関数
3. ノードの基本的な親子構成
4. NCAS のデータ構造
5. 参照型 Object とインスタンス Object
6. メッセージの伝播と SendMsg
7. MSG\_DO\_JOB と CJobDoer

## 1. クラス一覧

NCAS システムにおける全クラスのリストとその役割を以下に記述している。クラス名の次に括弧に囲まれているのがその基本クラスを示している。

... in header ELEM.H

クラス CViecFCList (CObject)

リストのクラスである。CviewFCの子を維持するためなどに使われる。

クラス CViecFC (CObject)

NCAS システム構築のために様々なクラスが定義される。これはその定義されるすべての基本クラスになっている。

クラス CMSUnknown (CMathSymb)

参照型のノードでまだ参照ノードとリンクされていないインスタンスと一時的にリンクされるためのクラスである。

クラス CMSIndep (CMathSymb)

独立変数を表現するクラス

クラス CMSPhys (CMathSymb)

物理量変数を表現するクラス

クラス CMSDep (CMSPhys)

従属変数

クラス CMSIndex (CMathSymb)

インデックスあるいは従属変数の添え字に使われる変数

クラス CMSNumber (CMathSymb)

数値の定数

クラス CMSReal (CMSNumber)

実数

クラス CMSInt (CMSNumber)

整数タイプ 1

クラス CMSInteger (CMSNumber)

整数タイプ 2

クラス CMSConst (CMathSymb)

変数定数

クラス CMSOperator (CMathSymb)

数学演算子

クラス CMSOPlus (CMSOperator)

加算 (プラス)

クラス CMSOMinus (CMSOperator)

減算 (マイナス)

クラス CMSONeg (CMSOperator)

負 (ネガティブ)

クラス CMSOEqual (CMSOperator)

等号 (=)

クラス CMSOMult (CMSOperator)

掛算 (×)

クラス CMSODiv (CMSOperator)

割り算 (／)

クラス CMSODiff (CMSOperator)

偏微分演算子

クラス CMSOAbs (CMSOperator)

絶対値演算子

クラス CGElem (CViecFC)

隠蔽機能を備えたコンテナ

クラス CPElem (CGElem)

入出力パイプを持つコンテナ

クラス CIElem (CPElem)

代用の要素

クラス CSymbInst (CPElem)

参照ノードのリンクあるいはインスタンス

クラス COperatorFolder (CPElem)

演算子の定義群を格納するコンテナ

クラス `CEquInst (CIElem)`

式 (等号) のインスタンス

クラス `CMSMatrix (CMathSymb)`

連立方程式における係数行列

クラス `CSuitCase (CObject)`

メッセージ送信時に伝播するスーツケースとして使われるコンテナ

クラス `CJobDoer (CObject)`

`MSG_DO_JOB` メッセージが送信された時にある特定の仕事をを行うクラスの基本クラス

クラス `CElemView (CViecFC)`

ウィンドウの基本クラス

クラス `CGDoc (CDocument)`

ドキュメントビューアの NCAS システム構築のドキュメントクラス

クラス `CGMainDoc (CGDoc)`

`CGDoc` の派生クラスでアプリケーション一つに一つしか存在しない

クラス `CGTemplate (CMultiDocTemplate)`

NCAS システムのドキュメントテンプレート

クラス `CGView (CScrollView)`

ドキュメントビューアの NCAS システム構築のドキュメントクラス

... in header `PDE_MSH.H`

クラス `CMSMeshSize (CMSInteger)`

メッシュサイズを意味する整数

クラス `CPDEBoundDefs (CPDEBase)`

境界定義

クラス `CPointOnAxis (CSymbInst)`

軸上の点をあらわすクラスで、ここでは境界の位置を表すためである。

クラス `CPointOnAxisOfMesh (CPointOnAxis)`

物理量の定義位置

クラス `CHalfAndDir (CViecFC)`

境界の位置を表すためのもので、グリッド上にあるかグリッドの間にあるかを表すクラス。

クラス `COnePointOnAxis (CMSInteger)`

境界の位置を表すためのもので、何番目のメッシュにあるかを表すクラス

クラス `COnePointOnAxisOfMesh (COnePointOnAxis)`

境界の位置を表すためのもので、何番目のメッシュにあるかを表すクラス  
クラス `CSurfOnAxisOfMesh (CPointOnAxisOfMesh)`  
境界を表現するもので、境界壁の向きを表す。  
クラス `CRangeOnAxisOfMesh (CPointOnAxisOfMesh)`  
境界を表現するもので、境界壁の範囲を表す。  
クラス `CRangeOnAxis (CPointOnAxis)`  
軸の範囲を表すクラス  
クラス `CPDEBoundDef (CPDEBase)`  
境界の定義のクラス  
クラス `CPDEMeshMap (CPDEBase)`  
メッシュマップで、メッシュの絵を描画するクラス。  
クラス `CPDEMeshInfo (CPDEBase)`  
メッシュ情報のコンテナ  
クラス `CBoundLine (CSymbInst)`  
メッシュマップ上の境界線を描画するクラス

... in header `PDE_BAS.H`

クラス `CPDEBase (CPElem)`  
偏微分方程式 (PDE) 関係のクラスの基本クラス  
クラス `CPDEIElem (CIElem)`  
偏微分方程式 (PDE) 関係のクラスの代表クラス  
クラス `CPDESet (CPDEBase)`  
支配偏微分方程式、境界条件群から構成される一セットのコンテナ  
クラス `CPDESpaceVar (CPDEBase)`  
空間変数  
クラス `CPDETimeVar (CPDESpaceVar)`  
時間変数  
クラス `CPDESpaceVars (CPDEBase)`  
空間変数群のコンテナ  
クラス `CPDEProcess (CPDEBase)`  
ある特定のプロセスを表すボックスである。数式処理プロセスや離散化プロセスやプログラム生成プロセスなどがある。  
クラス `CPDEPrcManip (CPDEProcess)`  
数式処理プロセス  
クラス `CPDEPrcDisc (CPDEProcess)`  
偏微分方程式の離散化プロセス

クラス CPDEObject (CPDEBase)  
プロセスの結果を格納するコンテナ

クラス CPDEPosDef (CPDEBase)  
物理変数の1メッシュの定義場所を表すクラス。

クラス CMSDepVar (CMSDep)  
従属変数

クラス CPDEDepInfo (CPDEBase)  
従属変数関連の情報のコンテナ

クラス CPDEParaInfo (CPDEBase)  
パラメータのコンテナ

クラス CPDEFileFold (CPDEParaInfo)  
ファイル定義のコンテナ

クラス CPDEDepPDE (CPDEBase)  
とくべきある特定の物理量変数に関する情報のコンテナ

クラス CPDEBoundCond (CSymbInst)  
境界条件のクラス

クラス CPDEBoundConds (CPDEBase)  
強権条件群を格納するコンテナ

クラス CPDETarget (CSymbInst)  
ソルバーセットのメンバーで解くターゲットの変数

クラス CPDESolvSet (CPDESet)  
ソルバーセット

クラス CPDERunSpread (CPDEBase)  
すべてのソルバーを展開したときの結果のコンテナ

クラス CPDEAbstProcVis (CPDEBase)  
ソルバーの計算順序を可視化するクラス

クラス CPDEObtainedVars (CPDEBase)  
可視化中の途中結果の変数を格納するコンテナ

クラス CPDEInvokedProcs (CPDEBase)  
計算ステップごとに施される (実行される) ソルバー

クラス CPDELineTerm(CPDEBase)  
改行マーク  
横に並ぶオブジェクトをこのクラスまでにして次の行に移る。

クラス CPDEMTxDefs (CPDEBase)  
係数行列群を格納するコンテナ

クラス CPDEBrkPntDefs (CPDEMTxDefs)

ブレイクポイント (チェックポイント) 群の定義

クラス CPDEConstDef (CMSIndex)

定数群のコンテナ

クラス CMSIndepVar (CMSIndep)

独立変数クラス

クラス CMSIndepConst (CMSIndep)

独立変数の定数

... in header PAD.H

クラス CPad (CPElem)

PAD 図の基本クラス、PAD 図関係の他のクラスはすべてこのクラスを継承しなければならない。PAD 図関係の共通した機能を持っている。

クラス CPadInst (CIElem)

PAD 図の代表クラス、その部分の PAD の図が別のウィンドウに表示させる際に元の窓の位置に代表して現れるものである。

クラス CPDSubst (CPad)

代入ステートメント (代入文) 例えば  $A=B$ 。

クラス CPDBlock (CPad)

ブロック、複数のステートメント

クラス CPDProg (CPad)

プログラム

クラス CPDProc (CPDProg)

サブルーチン

クラス CPDIf (CPad)

IF 文

クラス CPDFor (CPDProg)

FOR ループ

クラス CPDSdLay (CPad)

子メンバーを横に並べるコンテナ

クラス CPDVar (CPad)

変数宣言のコンテナ

クラス CPDArgs (CPDVar)

サブルーチンの引数のコンテナ

クラス CPDArgs2 (CPDArgs)

サブルーチンの引数のコンテナ

クラス CPDOutput (CPad)

数値データをファイルや標準出力デバイスに出力する

クラス CPDForSub (CPad)

FOR ループの様々な条件のコンテナ

クラス CPDComm (CPad)

コメント (解釈)

クラス CPDBreak (CPad)

ループを抜けるブレイクステートメント

クラス CPDCall (CPad)

サブルーチンをコールするステートメント

... in header JD\_PRG.H

クラス CJDGatherProc (CJDAAlgManip)

サブルーチンのすべてを集めるクラス

クラス CJDGatherDepVar (CJDAAlgManip)

従属変数 (解くべき変数) を集めるクラス

クラス CJDGatherVar (CJDAAlgManip)

変数を集めるクラス

クラス CJDCreateProgInC (CJDAAlgManip)

C プログラムを生成するクラス

... in header JD\_UTL.H

クラス CTargetDep

ソルバーセットのメンバーで、ターゲットの変数を表すクラス

クラス CJDAAlgManip (CJobDoer)

記号処理するクラス

クラス CJDCheckIfEqual (CJDAAlgManip)

等号 (式) のノードかどうかをチェックするクラス

... in header JD\_PRC.H

クラス CJDCreateMainProc (CJDAAlgManip)

メインサブルーチンのクラス

クラス CJDCreateProc (CJDAAlgManip)

ソルバーサブルーチンを生成するクラス

... in header JD\_DSC.H

クラス CJDDescretizer (CJDAAlgManip)

離散化するクラス

クラス CJDDescDeriv (CJDDescrretizer)

偏微分の部分だけを離散化するクラス

クラス CJDDescUpwndFstOrd (CJDDescrretizer)

風上差分で離散化するクラス

クラス CJDShiftToPos (CJDDescrretizer)

添え字の場所をシフトさせるクラス

クラス CJDMapToFit (CJDDescrretizer)

実際に存在しない添え字 (1 / 2) を平均化して直す

クラス CJDCheckDepend (CJDDescrretizer)

変数の依存関係をチェックするクラス

クラス CJDCheckIfIncluded (CJDAlgManip)

IF 文が含まれているかをチェックするクラス

... in header JD\_ALG.H

クラス CJDSimpEq (CJDDescrretizer)

数式を簡単化するクラス

クラス CJDExpEq (CJDDescrretizer)

クラス CJDExpEq2 (CJDDescrretizer)

数式を展開するクラス

クラス CJDBindEq (CJDDescrretizer)

数式の結合を行うクラス

クラス CJDExpEqNum (CJDDescrretizer)

定数の数値を展開するクラス

クラス CJDBindEqNum (CJDDescrretizer)

定数の数値を結合するクラス

クラス CJDFormEq (CJDDescrretizer)

行列係数を作成するクラス

... in header VAR\_WTC.H

クラス CPDE3DimVar (CPDEBase)

3次元変数の可視化ボックスのコンテナ

クラス CPDE3DimVarSP (CPDEBase)

3次元変数の内容の描画クラス

クラス CPDE3DimVarCP (CPDEBase)

3次元変数の可視化を制御するコントロールパネル



クラス CPDE2DimVar (CPDEBase)  
2次元変数の可視化ボックスのコンテナ  
クラス CPDE2DimVarSP (CPDEBase)  
2次元変数の内容の描画するクラス  
クラス CPDE2DimVarCP (CPDEBase)  
2次元変数の可視化に制御するコントロールボックス  
クラス CPDE1DimVar (CPDEBase)  
1次元変数の可視化ボックスのコンテナ  
クラス CPDE1DimVarSP (CPDEBase)  
1次元変数の内容を描画するクラス  
クラス CPDE1DimVarCP (CPDEBase)  
1次元変数の可視化を制御するコントロールボックス

... in header PDE\_UTL.H

クラス CMSInteger (CMSConst)  
整数の定数  
クラス CMSReal (CMSConst)  
実数の定数  
クラス CMSRealSub (CMSReal)  
実数の定数  
クラス CMSUnknown2 (CMSUnknown)  
定義されていない参照型  
クラス CMSFract (CMSNumber)  
分数型の数値  
クラス CMSRealEx (CMSNumber)  
実数型の数値  
クラス CMSPDEMtx (CMSMatrix)  
連立方程式の係数行列  
クラス CMSPDEVct (CMSPDEMtx)  
2次元変数つまりベクトル  
クラス CPDEBtn (CViecFC)  
ボタン  
クラス CEFile (CViecFC)  
結果の出力ファイル

... in header PDE\_PRG.H

クラス CPDEObjPrg (CPDEObject)  
計算プログラムを格納するコンテナ

クラス CPDEBrkObj (CPDEBase)  
ブレイクオブジェクト  
プログラム中にセットするチェックポイントに使われる

クラス CPDBrkPnt (CPDSubst)  
プログラムの動きを止める実際のブレイクポイントオブジェクト

クラス CPDFFile (CPad)  
結果出力ファイル

クラス CPDESngVarBox (CPDEBase)  
変数定義のボックス

クラス CPDESngVar (CViecFC)  
変数

... in header PDE\_ADV.H

クラス CPDEInfo (CPDEBase)  
PDE 問題の全入力情報を格納するボックス

クラス CPDERange (CPDEBase)  
PDE 偏微分方程式の支配する範囲 (メッシュ上の範囲)

クラス CPDEProblem (CPDEBase)  
問題全体を格納するクラス  
これがアプリケーションの一番トップに置かれるものである。

## 2. クラスのメンバー変数およびメンバー関数

... in ELEM.H

クラス CViecFC base-> CObject  
一番基本のクラス

CEText \*m\_petName;  
ノード名を意味する文字列型

CEText \*m\_petFont;  
ノード名を表示するためのフォント名

BYTE m\_nDot, m\_nCharSet, m\_nItalic;  
ノード名を表示するための可視化情報、ドット数、文字の種類、イタリックかなど

BYTE m\_bFindFirst;

リンクを張ろうとする際に名前をマッチさせる必要がある。  
マッチしたノードを検索するときに優先的にチェックするフラグに  
なる。これによりリンクの検索を加速することができる。

**BOOL m\_bSelected;**

ノードがセレクトされているかのフラグ

**BOOL m\_bFocus;**

ノードがフォーカスされているかのフラグ

**BOOL m\_bDrag;**

ノードがドラッグされているかのフラグ

**BOOL m\_bDragAllow;**

ノードがドラックされることが許されるかのフラグ

**CRect m\_extent;**

ノードの可視化あるいは表示範囲を意味する長方形

**CPoint m\_org;**

ノードの表示位置（表示される座標）

**CViecFC \*m\_parent;**

ノードの親を指すポインタ

**CViecFC \*m\_focus;**

子メンバーの中どれがフォーカスされているかを指すポインタ

**CViecFCList m\_child;**

子メンバーのリストを格納しているメンバー変数

**BOOL m\_bEncap;**

子メンバーが可視化状態にあるかを示すフラグ

**CPoint t\_point;**

ユーザがマウスポインタでノードアクセスするときそのポインタ  
の座標を維持する位置的変数

**CFile \*m\_pFile;**

ノードを保存するときのファイルオブジェクト

**CArchive \*m\_pArc;**

ノードを保存するときのアーカイブ

**static int s\_nDot;**

ノード名を表示するフォントの初期ドット数

**enum dirMode { dirNot=0, dirUp=-1, dirDown=1 };**

アクセス方向、メッセージ送信方向を意味する定数

**enum modeFile { modeRead=1, modeWrite };**

ファイルアクセスのモード（読み込み、書き込み）

**int OnFocus(CPoint &p);**  
与えられている座標 p がノードの表示範囲内に含まれているかをチェックするメンバー関数

**int OnRun();**  
ノードにある特定の仕事を実行させるメンバー関数（未使用）

**int OnAccess(CPoint \*p = NULL);**  
ユーザがノードにマウスポイントでアクセスする際に呼ばれるメンバー関数

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**  
ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数である。

**int OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags);**  
ノードがフォーカスされている状態でキーが放されたときに呼ばれるメンバー関数

**int OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);**  
ノードがフォーカスされている状態である文字が押されたあとに呼ばれるメンバー関数

**int OnDrag(const CPoint& old, const CPoint& last);**  
ノードがマウスポインタでドラッグされているときに呼ばれるメンバー関数

**int OnDrop(const CPoint& last);**  
ノードがドラッグされたあとにドロップされる。ドロップされたときにこのメンバー関数が呼ばれる。

**void OnDraw(CDC \*pdc, const CPoint& p);**  
自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**void OnDrawBody(CDC \*pdc, const CPoint& p);**  
ノード自身（ノードだけ子メンバーを含まない）を描画するメンバー関数

**void OnDrawDrag(CDC \*pdc, const CPoint& p);**  
ノードがドラッグされているときにノード関係の描画のために呼ばれるメンバー関数である。

**void OnDrawFocus(CDC \*pdc, const CPoint& p);**  
フォーカスされている状態のノードがフォーカスされていることを描画するために呼ばれるメンバー関数である。

**int OnCommand(UINT id);**  
ユーザがノードのアクセスメニューを選択した時にその選択された  
コマンドとともに呼ばれるメンバー関数である。

**BOOL IsInside(CViecFC \*pel);**  
pel というノードがこのノードの中に存在しているかをチェックす  
るメンバー関数

**const CRect& GetExtent();**  
ノードの表示範囲を取得するメンバー関数

**const CPoint& GetOrg();**  
ノードの表示開始位置 (座標) を取得するメンバー関数

**CRect GetRect();**  
ノードをフィットする長方形を計算しその長方形を返すメンバー関  
数

**void GetRectG(CRect& rr);**  
GetRect と同じ役割をするメンバー関数ではあるが、これは全ウイ  
ンドウに対する長方形の座標を計算する。

**CSize GetSize();**  
ノードが表示されるときに占める領域 (サイズ) を返すメンバー関  
数

**CViecFC\* GetFocus();**  
ノードがフォーカスされるかをチェックするメンバー関数。

**void SetFocus(BOOL b);**  
ノードフォーカスされた状態にセットするメンバー関数

**CViecFC\* GetParent();**  
ノードの直結の親ノードを取得するメンバー関数。

**BOOL SubstChild(CViecFC \*pFrom, CViecFC \*pTo, dirMode  
dir=dirUp);**  
ノードの子メンバーを置換するメンバー関数

**int CalcExt(dirMode d=dirUp);**  
ノードの表示される範囲 (長方形) を計算するメンバー関数。これ  
によってノードの画面上の占める長方形の領域が計算し、m\_extent  
に保存するメンバー関数である。

**void Serialize(CArchive& ar);**  
ノードとそれ以下の子メンバーを保存するメンバー関数

**void SerializeBody(CArchive& ar);**  
Serialize 関数のヘルパ関数で自分の分だけの保存を行うメンバー関

数。

**BOOL OnDelete(CViecFC \*);**

ノードが削除される時呼ばれるメンバー関数

**BOOL SendMsg(int, CObject\*);**

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。るノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**CViecFC\* GetUnknownElem(CArchive& ar, const char \*);**

未定義の要素を検索するメンバー関数

**CElemView \*GetNewView(PETCHAR );**

ディスプレイ上に新しいウインドウを作成する

**POSITION AddTail(CViecFC \*);**

新たに子メンバーを最後に追加するメンバー関数

**POSITION AddHead(CViecFC \*);**

新たに子メンバーを最初に追加するメンバー関数

**CViecFC \*Copy();**

ノードのコピーを作成するメンバー関数

**void CopyBody(CViecFC \*);**

子メンバーを除いて自分のノードだけをコピーするメンバー関数

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**CRect ToCalcExt(CSymbInst\*, dirMode d=dirUp);**

参照型のメンバー関数でリンクされた状態でインスタンスの表示範囲を計算する。

**void ToDraw(CSymbInst \*pInst, CDC \*pdc, const CPoint& shift);**

参照型のメンバー関数でリンクされた状態でインスタンスの描画を行うメンバー関数である。インスタンスが現れる姿を自分で描画しないで自分がリンクしているノードに負かせて描画してもらうのである。

**PETCHAR GetDrawName();**  
 実際に描画させる名前の文字列

**void OnDrawChild(CDC \*pdc, const CPoint& p);**  
 子メンバーを表示させるメンバー関数

**void SetName(PETCHAR name, BYTE nDot, BYTE nCharSet);**  
 名前をセットする関数である。名前の文字列と共に表示するときのサイズ（ドット数）、Character set をセットする。

**void SetName(PETCHAR name);**  
 名前の文字列だけを設定するメンバー関数である。

**PETCHAR GetName();**  
 ノードの名前の文字列を取得するメンバー関数である。

**PETCHAR GetFont();**  
 ノードの名前を描画する際に用いられるフォント名の文字列を取得するメンバー関数である。

**size\_t GetNameLen();**  
 ノード名の文字列の長さを返すメンバー関数である。

**size\_t GetFontLen();**  
 ノード名を表示するときフォント名の文字列の長さを返すメンバー関数である。

**CViecFC \* GetChild(int no);**  
 No 番目の子メンバーのポインタを取得するメンバー関数である。

**CViecFC \*GetElemByName(const char \*, dirMode dir=dirUp, CViecFC\* pfrom=NULL);**  
 名前の文字列でもってエレメントを検索するメンバー関数である。

**BOOL GetFindFirst();**  
 ノード名を検索する際に優先的に検索するかのフラグを返すメンバー関数である。

**void SetFindFirst(BOOL b);**  
 ノード名を検索する際に優先的に検索するかのフラグをセットするメンバー関数である。

**void SetFocusChild(CViecFC \*fc);**  
 ノードの子メンバーである f c をフォーカス状態にセットするメンバー関数である。

**void CopyChild(CViecFC \*);**  
 ノードにぶら下がっているすべての子のコピーを作成する。

**int GetCount();**

子メンバーの数を取得するメンバー関数である。

**POSITION GetHeadPosition();**

子メンバーの最初の位置を取得するメンバー関数である。

**BOOL SendMsgToChild(int, CObject\*);**

子メンバーにある特定メッセージを送信するメンバー関数である。

**CViecFC \*GetNext(POSITION&);**

子メンバーを網羅する際に次の子メンバーのポインタを取得するメンバー関数である。

**POSITION FindIndex(int ind);**

ind 番目の子メンバーのポジションを取得するメンバー関数である。

**CViecFC \*GetAt(POSITION pos);**

pos のポジションにある子メンバーのポインタを取得するメンバー関数である。

**void SerialChild(CArchive& ar);**

子メンバーをシリアルライズ (格納、保存) するメンバー関数である。

**CViecFC \*RemoveHead();**

最初の子メンバーを削除する

**CViecFC \*RemoveTail();**

最後の子メンバーを削除する

**void RemoveAll();**

全ての子メンバーを削除する

**int Release();**

?

**void InvalidateRect(const CPoint&);**

ノードの描画領域を無効化し強制的に再描画させるメンバー関数である。

**void InvalidateRectBd(const CPoint& point, const CRect& rb = CRect(0,0,0,0), int wd=2);**

ノードの描画領域の枠の部分だけ (全領域でなく) を無効化し強制的に再描画させる。

**CDC\* GetDC();**

MS-Windows の描画に用いられるデバイスコンテキスト (DC) を取得するメンバー関数である。

**int ReleaseDC(CDC\*);**

GetDC で取得したデバイスコンテキスト (DC) を開放するメンバー関数である。



`int SetFont(CDC *pdc, int dot, PETCHAR name, BYTE nCharSet, BYTE nItalic=TRUE, int wd=0);`

ノード名表示に用いられるフォント名をセットするメンバー関数である。

`int SetPen(CDC *pdc, int nPenStyle, int nWidth, COLORREF crColor);`

MS-Windows の描画に用いられるオブジェクトである Pen をセットするメンバー関数である。

`int SetBrush(CDC *pdc, COLORREF crColor);`

MS-Windows の描画に用いられるオブジェクトである Brush をセットするメンバー関数である。

`void SetOldGdiObject(CDC *pdc);`

MS-Windows の描画に用いられるオブジェクトをデフォルト状態に戻すメンバー関数である。

`CViecfC* GetView();`

ノードが描画されているウィンドウ (ビュー) のポインタを取得する関数である。

`CViecfC* GetAncestor();`

一番上のノード (ルートノード) を取得する関数である。

`int SendCmdQ(int cmd);`

Window のメッセージキューを通してメッセージを送信する関数である。

`BOOL PostCmdQ(int cmd);`

Windows のメッセージキューを通してメッセージをポストする関数である。

`CEText *GetNewTChar(PETCHAR pch, CEText *old);`

新しい文字列を作成する関数である。返り値がその文字列型へのポインタになる。

`BOOL FocusToView(CViecfC *pel);`

ある特定のビューをフォーカスされた状態にセットする。

`void DrawFold3D(CDC *pdc, const CPoint& p, int mode = 0);`

ノードの長方形枠を 3 次元ボタンに描画する関数である。

`int GetId();`

ノードの ID 番号を取得する関数である。

`void SetId(int id);`

ノードの ID 番号を設定する関数である。

**int SetExtent(const CRect& r);**  
ノードの描画領域をセットする関数である。

**int SetOrg(const CPoint& p);**  
ノードの描画位置を設定する関数である。

**void SetParent(CViecFC \*);**  
ノードの親ノードを設定する関数である。

**void SetDragAllow(BOOL);**  
ノードのドラッグが許されるかを設定する関数である。

**BOOL GetDragAllow();**  
ノードのドラッグが許されるかを取得する関数である。

**CPoint GetMenuPoint(CPoint \*);**  
ポップアップメニューの表示場所を計算する関数である。

**CMenu \*GetPopupMenu(BOOL fg=TRUE);**  
新しいポップアップメニューを作成しそのポインタを返す関数である。

**BOOL TrackPopupMenu(CMenu\*, CPoint p0);**  
ポップアップメニューを表示させユーザがメニューを選択するかエスケープが押されたかまで表示し続ける。

**CArchive \*OpenFile(const char\*, modeFile);**  
ファイルを開き、ノードの内容を保存する準備する関数である。

**void CloseFile();**  
**OpenFile** で開いたファイルを閉じる関数である。

**void DrawArrow(CDC \*pdc, const CPoint& p1, const CPoint& p2);**  
矢印付きの線を描画する関数である。

**CRect GetTextExtent(CDC \*pdc, const char \*pch, size\_t nSize);**  
描画する文字列の寸法を取得する関数である。

**void UnionExtent(CRect& dest, CRect& src);**  
2つの長方形をユニオン（加算）する関数である。

**void TextOut(CDC \*pdc, int x, int y, const char \*pch, size\_t len);**  
文字列を描画する関数である。

**void CalcExtPost(dirMode dir, const CRect& extent);**  
描画領域を計算するヘルパ関数である。

**void SelectStockObject(CDC \*pdc, int nIndex);**  
描画オブジェクトを作成しないで、ストックオブジェクトを使用するメンバー関数である。

**void SetTextColor(CDC \*pdc, DWORD col);**

ノード名を描画する際に使われる色を設定する関数である。

```
void EtchingRect(CRect& rct, CRect& eth);
```

?

```
CViecFC *GetBaseElem(CRuntimeClass *);
```

基本クラスを取得する関数である。

```
CViecFC *GetChildElem(CRuntimeClass * pRC);
```

子メンバー中 pRC のクラスのものを選択しその子のポインタを返すメンバー関数である。

クラス CMathSymb base->CViecFC

```
BYTE m_dimension[4]; // 0,1,2,3
```

次元数を保持するメンバー変数である。

```
SetDimension(BYTE dim0, BYTE dim1, BYTE dim2, BYTE dim3);
```

次元数をセットするメンバー関数である。

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extent に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

ノードを描画するメンバー関数である。

```
CRect ToCalcExt(CSymbInst*, dirMode d=dirUp);
```

Reference Object 固有のもの、リンクされているノードのためにその描画される領域を計算するメンバー関数である。

```
void ToDraw(CSymbInst *pInst, CDC *pdc, const CPoint& shift);
```

参照型のメンバー関数でリンクされた状態でインスタンスの描画を行うメンバー関数である。インスタンスが現れる姿を自分で描画しないで自分がリンクしているノードに負かせて描画してもらうのである。

```
int GetnDot(int level, int dot);
```

数式描画のときに、ドット数を計算するメンバー関数で、文字列を描画する際に使われる。

```
int GetSpc(int level);
```

数式ノードを描画する際にノード間のスペースを計算するメンバー関数である。

```
int OnCommand(UINT id);
```

ノードにコマンド（主にポップアップメニューから選択された命令）

が送られたときに呼ばれるメンバー関数である。

クラス CMSNumber base->CMathSymb

BOOL IsNeg();

この数値が負か否かを取得するメンバー関数である。

クラス CMSInteger base->CMSNumber

int m\_value;

整数型の実際の数値が格納されるメンバー変数である。

クラス CGElem base->CViecFC

enum mode { modeOpen=1, modeClose=2 };

ノードに属する子メンバーを見せるか (open) 見せないか (close) を表現する定数 (列挙) である。

mode m\_mode;

ノードに属する子メンバーを見せるか (open) 見せないか (close) の情報を保存するメンバー変数である。

CRect m\_RName;

ノード名だけを描画する場合の占める領域を維持するメンバー変数である。

UINT m\_nFormat;

?

BOOL m\_bCreated;

ノードが誕生してからはそのノードの OnCreated メンバー関数が呼ばれたかを意味するメンバー変数である。

CIElem \*m\_pInst;

ノードが同時に他のウィンドウに代表して描画されている場合の代表ノードへのポインタをキープするメンバー変数である。

CRect m\_oldRect;

古い描画領域を維持するメンバー変数である。

CIElem \*GetNewInst();

新しい代表ノードを取得するメンバー関数である。

UINT GetFormat();

フォーマットを取得するメンバー関数である。

void SetFormat(UINT d);

フォーマットを設定するメンバー関数である。

BOOL IsInst();

ノードが代表ノードかどうかを取得するメンバー関数である。

void SetInst(CIElem \*pI);

ノードの代表ノードを設定するメンバー関数である。

**mode GetMode();**

ノードの現在のモードを取得するメンバー関数である。

**CIElem \*GetInstBody();**

ノードの代表ノードへのポインタを取得するメンバー関数である。

**BOOL ToDb1Body();**

ノードを新たに作成したウィンドウに描画させ、元のウィンドウには代表ノードを作成して代表して描画するという一連の作業を行うメンバー関数である。

**void SetOldRct(const CRect& r);**

古い描画領域を設定するメンバー関数である。

**const CRect& GetRName();**

ノード名だけを描画する場合の占める領域を取得するメンバー関数である。

**BOOL AppendDefaultMenu(CMenu \*);**

デフォルトメニューを作成するメンバー関数である。

クラス **CPElem** base->**CGElem**

**CViecFCList m\_rpInput, m\_rpOutput;**

入出力を持つプロセスノードの場合にその入力先およびその出力先のリストを維持するメンバー変数である。

**BOOL m\_bZoneFg;**

マウスが動いてポインタがウィンドウ上に描画されているノードのどのゾーンにあるかを示すメンバー変数である。

**int GetInputCount();**

入力先の数を取得するメンバー関数である。

**CViecFC \*GetInputChild(int n);**

n 番目の入力先を取得するメンバー関数である。

**CRelationPipe \*FindInputRel(CViecFC \*pS);**

pS という入力先を探しそのパイプへのポインタを返すメンバー関数である。

**int GetOutputCount();**

出力先の数を取得するメンバー関数である。

**CViecFC \*GetOutputChild(int n);**

n 番目の出力先を取得するメンバー関数である。

**CRelationPipe \*FindOutputRel(CViecFC \*pS);**

pS という出力先を探しそのパイプへのポインタを返すメンバー関数である。

クラス **CIElem base-> CPElem**

他のノードを代表して表示するノードである。

**CGElem \*m\_pBody;**

自ノードが代表している実態のノードのボディへのポインタを維持するメンバー変数である。

**BOOL DblInst(CGElem \*);**

代表ノードをダブるメンバー関数である。

**BOOL PackInst();**

ダブられた (代表) ノードを合体するメンバー関数である。

**CGElem \*GetBody();**

代表する大元のノードへのポインタを取得するメンバー関数である。

クラス **CSymbInst base-> CPElem**

**CViecFC \*m\_pSymb;**

この **CSymbInst** クラスはインスタンス型のクラスで、このメンバー変数は参照している **Reference** ノードへのポインタを維持している。

**CEText \*m\_petSymbName;**

参照ノードのノード名を維持するメンバー変数である。

**int m\_level;**

数式の際に一番上の等号 (=) からいくつめのレベルかを表すメンバー変数である。

**int m\_int;**

**General purpose** の整数変数である。

**double m\_double;**

**General purpose** の実数変数である。

**CIElem \*GetNewInst();**

新しい代表ノードを作成しその代表ノードへのポインタを返すメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに (マウス右ボタンが押されたとき) 呼ばれるメンバー関数である。

**CSymbInst \*GetBaseSymbInst(CRuntimeClass \*);**

?

**CViecFC \*GetSymbBody();**

リファレンスオブジェクト（リンクされるノード）の場合、ノードがどのノードにリンクされているかそのノードを返すメンバー関数である。

**PETCHAR GetSymbName();**

シンボル名を取得する関数である。

**size\_t GetSymbNmLen();**

シンボル名の文字列の長さを取得する関数である。

**BOOL ChangeInst(const char\*);**

シンボル名をセットし直します。

クラス **CSuitCase** base-> **CObject**

**SendMsg** 関数によってある特定のメッセージがノード中を伝播するとき、そのメッセージとともに伝播するコンテナクラスで、処理結果を集める役割をするものである。

**int m\_int;**

単独の整数の結果を入れる変数である。

**double m\_double;**

単独の実数の結果を入れる変数である。

**int m\_intAry[10];**

10 個まで複数個の整数結果を入れる整数配列である。

**int m\_doubleAry[10];**

10 個まで複数個の実数結果を入れる実数配列である。

**CObject \*m\_case, \*m\_case2;**

**Cobject** へのポインタが処理結果になる場合それぞれのポインタを入れるポインタ変数である。

**CViecFCList \*m\_oblist;**

リスト (**CViecFCList**) のポインタが結果の場合に、そのポインタを入れるポインタ変数である。

**CRuntimeClass\* m\_runtimeClass;**

**CruntimeClass** 型のポインタが結果になる場合、そのポインタを入れる変数である。

**void \*m\_void;**

**void** 型へのポインタが結果の場合、そのポインタを入れる変数である。

クラス **CJobDoer** base->**CObject**

**MSG\_DO\_JOB** メッセージがノード中を伝播するとき、そのメッセージとともに伝播する実際に何かの特定の仕事をやるクラスで

**BOOL Do(CObject \*);**

実際に仕事をするメンバー関数である。この関数をオーバーライドしこの関数内に特定の仕事を定義することで実装できる。

クラス **CElemView** base->CViecFC

ウィンドウクラスである。オブジェクトが現れている窓（ウィンドウ）のクラスである。ウィンドウにあるべきあらゆる機能（描画機能、サイジング機能、メッセージセンデリング機能、。。。）及びリソース（ペン、ブラシ、フォント、。。。）を持つ。

**CGdiObject \*pOldPen, \*pOldBrush, \*pOldFont;**

設定前（デフォルト）のペン、ブラシ、フォントを入れる変数である。

**CScrollView\* m\_pView;**

MFC のスクロールオブジェクトで実際に描画機能やスクロール機能を実装しているクラスである。

**BOOL m\_bResizeFrame;**

フレーム（ウィンドウのこと）がサイズ変更できるかどうかのフラグである。

**void Initialize(CScrollView\* pvw);**

ウィンドウの初期化ルーチンである。

... in PDE\_MSH.H

クラス **CMSMeshSize** base->CMSCInteger

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**

メッシュのシンボル（IMAX, JMAX, ...）とその実際のサイズを維持するクラスである。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEBoundDefs** base->CPDEBase

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。



**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPointOnAxis** base->**CSymbInst**

**int m\_axisNo; // 0:x 1:y 2:z**

?

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス **CHalfAndDir** base->**CViecFC**

このクラスは境界の定義に用いられます。境界がグリッド上にあるか、グリッドの真ん中にあるのかといった情報や、境界面（壁）がどの方向に向

いているかの情報が入っています。

**BOOL m\_bHalf**

境界がグリッド上にあるかの情報を入れる変数であり、**FALSE** だったら境界がグリッド上にあつて、**TRUE** だったらグリッドの真ん中にあることを示す。

**BOOL m\_bPosDir;**

境界面の方向を示す。**TRUE** だったら軸のプラス方向で、**FALSE** だったら軸のマイナス方向であること示す。

**void CopyBody(CViecFC \*);**

複製を作成するメンバー関数である。

**void SerializeBody(CArchive& ar);**

ノードを保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、矢印キーでこのクラスに関する情報を変えることができる。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **COnePointOnAxis base->CMSInteger**

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、矢印キーで内部情報を変える。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**

に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス `COnePointOnAxisOfMesh` base->`COnePointOnAxis`

```
int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
```

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、矢印キーでこのクラスの情報を変える。

クラス `CSurfOnAxisOfMesh` base->`CPointOnAxisOfMesh`

```
CHalfAndDir *m_pHalfDir;
```

?

```
COnePointOnAxis *m_pAt;
```

?

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス `CRangeOnAxisOfMesh` base->`CPointOnAxisOfMesh`

```
COnePointOnAxis *m_pFrom;
```

?

```
COnePointOnAxis *m_pTo;
```

?

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス `CRangeOnAxis` base->`CPointOnAxis`

```
COnePointOnAxis *m_pFrom;
```

?

```
COnePointOnAxis *m_pTo;
```

?

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス `CPDEBoundDef` base->`CPDEBase`

```

CSurfOnAxisOfMesh *m_surf;
    ?
CRangeOnAxisOfMesh *m_range;
    ?
CRangeOnAxisOfMesh *m_range2;
    ?
BOOL CreateBoundDef1D(const char*, int, BOOL, BOOL);
    ?
BOOL CreateBoundDef(const char*, int, BOOL, BOOL, const char*,
int, int);
    ?
BOOL CreateBoundDef3D(const char*, int, BOOL, BOOL, const
char*, int, int, const char*, int, int);
    ?
void OnDraw(CDC *pdc, const CPoint& p);
    自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関
    数である。
int CalcExt(dirMode dir=dirUp);
    ノードの表示される範囲（長方形）を計算するメンバー関数。これ
    によってノードの画面上の占める長方形の領域が計算し、m_extent
    に保存するメンバー関数である。
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
    ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行
    うメンバー関数である。引数自ノードだけか子ノードまで初期化す
    るかを示す列挙型の変数である。
BOOL SendMsg(int, CObject*);
    あるノードにメッセージを送信するメンバー関数である。2 つの引
    数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味
    する。送られる側はこの整数値でメッセージを判断する。2 つ目の
    引数はそのメッセージで用いられるオブジェクトへのポインタであ
    る。

```

クラス CPDEMeshMap base->CPDEBase

```

CPDEMeshInfo *m_meshInfo;
    ?
CRect m_extIn;
    ?

```

`int m_meshNo[3];`

?

`BOOL SendMsg(int msg, CObject* el);`

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

`int OnDrop(const CPoint& last);`

?

`void OnDraw(CDC *pdc, const CPoint& p);`

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

`int CalcExt(dirMode dir=dirUp);`

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

`BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);`

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス `CPDEMeshInfo base->CPDEBase`

`CPDESpaceVars *m_spaceVars;`

?

`CPDEMeshMap *m_meshMap;`

?

`CPDEBoundDefs *m_boundDefs;`

?

`BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);`

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

`void OnDraw(CDC *pdc, const CPoint& p);`

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

`int CalcExt(dirMode dir=dirUp);`

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL CreateMeshInfo1();**

?

**BOOL CreateMeshInfo2();**

?

**BOOL CreateMeshInfo1D();////Kawa**

?

**BOOL CreateMeshInfo3D();////Kawa**

?

クラス **CBoundLine base->CSymbInst**

**int m\_verOrHor;**

?

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

... in **PDE\_BAS.H**

クラス **CPDEBase base->CPElem**

**CPDEBase \*GetBaseObj();**

?

int CalcExtVert(dirMode dir=dirUp, int mg=0);

?

int CalcExtMidVert(dirMode dir=dirUp, int mg=0);

?

int CalcExtHTab(dirMode dir=dirUp, int tabWd=50);

?

int CalcExtLines(dirMode dir=dirUp, int wd=10, int hg=10);

?

int CalcExtHMerge(dirMode dir, const char \*nm, int mg);

?

void OnDrawName(CDC \*, const CPoint&);

ノード名の文字列をウィンドウ上に表示する関数である。

void OnDrawRel(CDC \*pdc, const CPoint& p);

?

BOOL SendMsg(int msg, CObject\* el);

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

int OnAccess(CPoint \*p = NULL);

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

int OnCommand(UINT);

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

CIElem \* GetNewInst();

?

void OnDrawClose(CDC \*, const CPoint&);

?

クラス CPDEIElem base->CIElem

BOOL PackInst();

?

クラス CPDEIndexFold base->CPDEBase

BOOL SendMsg(int msg, CObject\* el);

あるノードにメッセージを送信するメンバー関数である。2 つの引

数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDESet base->CPDEBase**

**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。



```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
int OnCommand(UINT);
```

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス CPDESpaceVar base->CPDEBase

```
CMSIndepVar *m_pIndepVar;
```

?

```
CMSIndep *m_pIndep;
```

?

```
CMSIndex *m_pIndex;
```

?

```
CMSCInteger *m_division;
```

?

```
CPDESpaceVar();
```

初期化ルーチンである。

```
BOOL CreateSpace(const char* ind, const char* dx, const char *I,  
const char *IMAX, int no);
```

1次元分の空間あるいは軸関係の変数や定義などを作成する関数で、例えば CreateSpace(“x”, “dx”, “I”, “IMAX”, 20) と呼び出すとその空間の距離に使われる変数は x で 1 メッシュ分の距離は d x で、インデックス変数は I で、その軸上の分割数は IMAX で初期値が 2 0 の空間定義を実現できる。

```
BOOL CreateSpaceConst(const char* ind, const char* dx, const char  
*I, const char *IMAX, int no);
```

?

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

```
int OnCommand(UINT);
```

ユーザがノードのアクセスメニューを選択した時にその選択された

コマンドとともに呼ばれるメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*, const CPoint&);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDETimeVar** base->**CPDESpaceVar**

**CPDEIndexFold \*m\_indexFold;**

?

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*, const CPoint&);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL CreateTimeVar(const char\* ind, const char\* dx, const char \*I, const char \*IMAX, int no);**

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

クラス **CPDESpaceVars** base->**CPDEBase**

**int m\_spaceNo;**

空間の次元数を表すメンバー変数である。

**CPDESpaceVar \*m\_spaceVar[3];**

空間変数のクラスへのポインタを入れる配列である。これはスピードを高めるためにここにも保存する。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス **CPDEProcess base->CPDEBase**

**CViecFC \*MakeCopy(CViecFC \*pelm);**

複製を作成するメンバー関数である。

**BOOL GetInputOutput(CViecFC\*& pin, CViecFC\*& pout);**

ノードへのすべてのインプットとノードからのすべてのアウトプットを集めるメンバー関数である。

**CViecFC \*CopyProc();**

?

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEPrcManip** base->CPDEProcess

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEPrcDisc** base->CPDEProcess

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEObject** base->CPDEBase

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDEPosDef** base->CPDEBase

メッシュ上での定義場所の情報を保存するクラスである。

**BYTE m\_pos[3];**

?

**BYTE m\_velocity;**

速度を意味する変数なのかを示すフラグである。

**CPDEPosDef();**

初期化するメンバー関数である。

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、矢印キーで場所の定義を変更できる。

**void CopyBody(CViecFC \*);**

自分のノードの複製を作成するメンバー関数である。

**void SerializeBody(CArchive& ar);**

自分のノードの複製を保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

クラス **CMSDepVar** base->CMSDep

**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDEDepInfo** base->CPDEBase

**CViecFC \*m\_descript;**

説明の文字列を入れる文字列型の変数である。

**CMSDepVar \*m\_var;**

?

**CPDEPosDef \*m\_pos;**

?

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL CreateDep1D(const char \*nm, const char \*var, int s1, int s2);**

1次元の

**BOOL CreateDep2D(const char \*nm, const char \*var, int s1, int s2);**

?

**BOOL CreateDep3D(const char \*nm, const char \*var, int s1, int s2);**

?

クラス CPDEParaInfo base->CPDEBase

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDEFileFold** base->**CPDEParaInfo**

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT id);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEDepPDE** base->**CPDEBase**

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**BOOL CreateDepPDE();**

?

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

クラス **CPDEBoundCond** base->CSymbInst

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDEBoundConds** base->CPDEBase

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

クラス **CPDETarget** base->CSymbInst

クラス **CPDESolvSet** base->CPDESet

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。



**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDERunSpread** base->CPDEBase

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL CreateNewTemplate(CViecFC \*pelm);**

?

クラス **CPDEAbstProcVis** base->CPDEBase

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス CPDEObtainedVars base->CPDEBase

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス CPDEInvokedProcs base->CPDEBase

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス CPDELineTerm base->CPDEBase

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス CPDEMTxDefs base->CPDEBase

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス CPDEBrkPntDefs base->CPDEMTxDefs

int OnAccess(CPoint \*pp);

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

int OnCommand(UINT);

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス CPDEConstDef base->CMSIndex

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス CMSIndepVar base->CMSIndep

BOOL SendMsg(int msg, CObject\* el);

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CMSIndepConst base->CMSIndep

double m\_value;

?

CMSIndepConst(double val = 1.0);

?

double GetValue();

?

**BOOL SendMsg(int msg, CObject\* el);**

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extent に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**void SerializeBody(CArchive& ar);**

?

... in PAD.H

クラス CPad base->CPElem

int m\_bFixed;

?

static int s\_merginX;

?

static int s\_merginY;

?

static int s\_space;

?

CIElem \*GetNewInst();

?

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT id);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int CalcExtClose();**

ノードがクローズされた状態の描画領域（長方形）の計算をするメンバー関数である。

**int CalcExtOpen(CPoint &p = CPoint(0,0), int st=0);**

ノードがオープンされた状態の描画領域（長方形）の計算をするメンバー関数である。

**void OnDrawClose(CDC \*, const CPoint&);**

ノードがクローズされた状態の描画するメンバー関数である。

**void OnDrawOpen(CDC \*, const CPoint&, int st=0);**

ノードがオープンされた状態の描画するメンバー関数である。

**void OnDrawDoubleLine(CDC \*pdc, const CPoint& p);**

?

**BOOL IsFixed() return m\_bFixed;**

?

**void SetFixedFg(BOOL b);**

?

**BOOL AppendDefaultMenu(CMenu \*);**

デフォルトのメニューを作成し追加する関数である。

クラス CPadInst base->CIElem

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス CPDSubst base->CPad

**CIElem \*GetNewInst();**

新しい代表型（CIElem）のノードを作成する関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CPDBlock base->CPad

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT id);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これ

によってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
int CalcExtClose();
```

?

クラス `CPDProg` base->`CPad`

```
void OnDrawClose(CDC *, const CPoint&);
```

?

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
int OnCommand(UINT id);
```

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス `CPDProc` base->`CPDProg`

```
int CalcExtClose();
```

?

```
void OnDrawClose(CDC *, const CPoint&);
```

?

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
int OnCommand(UINT id);
```

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CPDIf base->CPad

**int m\_inedge;**

?

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CPDFor base->CPDProg

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**



あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

```
int CalcExtOpen(CPoint &p = CPoint(0,0), int st=0);  
    ?  
int CalcExtClose();  
    ?  
void OnDrawClose(CDC *, const CPoint&);  
    ?
```

クラス CPDSdLay base->CPad

```
int CalcExt(dirMode d=dirUp);  
    ノードの表示される範囲（長方形）を計算するメンバー関数。これ  
    によってノードの画面上の占める長方形の領域が計算し、m_extent  
    に保存するメンバー関数である。  
void OnDraw(CDC *pdc, const CPoint& p);  
    自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関  
    数である。
```

クラス CPDVar base->CPad

```
int CalcExt(dirMode d=dirUp);  
    ノードの表示される範囲（長方形）を計算するメンバー関数。これ  
    によってノードの画面上の占める長方形の領域が計算し、m_extent  
    に保存するメンバー関数である。  
void OnDraw(CDC *pdc, const CPoint& p);  
    自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関  
    数である。  
int OnAccess(CPoint *p = NULL);  
    ユーザによるノードへのアクセスが起こったときに（マウス右ボタ  
    ンが押されたとき）呼ばれるメンバー関数である。
```

クラス CPDArgs base->CPDVar

```
int CalcExt(dirMode d=dirUp);  
    ノードの表示される範囲（長方形）を計算するメンバー関数。これ  
    によってノードの画面上の占める長方形の領域が計算し、m_extent  
    に保存するメンバー関数である。  
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

クラス CPDArgs2 base->CPDArgs

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

クラス CPDOutput base->CPad

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

クラス CPDForSub base->CPad

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
BOOL SendMsg(int, CObject*);
```

あるノードにメッセージを送信するメンバー関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CPDComm base->CPad

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

```
int CalcExt(dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

クラス `CPDCall` `base->CPad`

`void OnDraw(CDC *pdc, const CPoint& p);`

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。

`int CalcExt(dirMode d=dirUp);`

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

... in `JD_PRG.H`

クラス `CJDGatherProc` `base->CJDAlgManip`

`BOOL Do(CObject *);`

子ノードに生成されたサブルーチン型のノード（`CPDProc`）を検索しそのポインタを集める仕事をするメンバー関数である。

`CPDProc *m_procLst[40];`

サブルーチン型ノードへのポインタの配列であり、集められたノードへのポインタがこの配列に集められる。

`int m_procNo;`

集められたサブルーチンの数を入れる変数である。

クラス `CJDGatherDepVar` `base->CJDAlgManip`

`BOOL Do(CObject *);`

従属変数または解くべき変数を集める仕事をする関数である。

`CTargetDep targetVar[40];`

集められた従属変数がこのメンバー変数に格納される。

`int targetNo;`

集められた従属変数の数を示す変数である。

`CJDGatherDepVar();`

クラスの初期化ルーチンである。

クラス `CJDGatherVar` `base->CJDAlgManip`

変数定義ノードをあるノード中から検索しそのノードへのポインタを集めるクラスである。

`CMapStringToOb m_depVar;`

従属変数を集めるマップ型変数である。

**CMapStringToOb m\_indepVar;**

独立変数を集めるマップ型変数である。

**CMapStringToOb m\_matrix;**

マトリックス変数を集めるマップ型変数である。

**CMapStringToOb m\_para;**

パラメータを集めるマップ型変数である。

**CMapStringToOb m\_const;**

定数 (Constant) を集めるマップ型変数である。

**CMapStringToOb m\_const2;**

定数 (Constant) を集めるマップ型変数である。

**CMapStringToOb m\_brkObj;**

ブレイクオブジェクトあるいはチェックポイントを集めるマップ型変数である。

**CMapStringToOb m\_file;**

入出力ファイルを集めるマップ型変数である。

**BOOL Do(CObject \*);**

自動生成された様々な変数やシンボルを集める仕事をする Do メンバー関数である。

クラス **CJDCreateProgInC base->CJDAlgManip**

作成されたノード上のプログラムを C 言語に変換しファイルに保存するクラスである。

**CString m\_fileName;**

C 言語のファイル名を保存する変数である。

**CPDEObjPrg \*m\_mainProg;**

メインサブルーチンへのポインタを示す。

**CPDEInfo \*m\_pdeInfo;**

問題入力ノードへのポインタを入れる変数である。

**CJDGatherProc findProc;**

集められるサブルーチンノードを格納する変数である。

**CJDGatherVar findVar;**

探したい変数である。

**CJDGatherDepVar findDep;**

?

**HANDLE m\_hFile, m\_hFileMap;**

C 言語のプログラムを作成する際にそのプログラムをファイルに格納する必要がある。そのファイルハンドル及び mapped file を格納

する変数である。

**LPVOID m\_lpvFile;**

File mapped memory の先頭アドレスを示すポインタである。

**char \*m\_nextPnt, \*m\_firstPnt;**

作成する C プログラムの次に格納する文字のアドレスと行の先頭アドレスを入れる変数である。

**int m\_maxWd, m\_startCol, m\_lineNo, m\_tabWd, m\_indentNo, m\_curCol;**

プログラムレイアウトに必要なパラメータを格納する変数であり、最大桁数、はじめの桁、行番号、タブの長さ、インデントの数、現在の桁番号などがある。

**char m\_lineFeed[3];**

改行コードを格納する配列変数である。

**BOOL m\_bArraySubsc;**

?

**BOOL m\_bUseAry0;**

?

**CPDEIndexFold \*pIndexFold;**

インデックスを格納するフォルダーのポインタである。

**CMapPtrToWord timeIndexTable;**

時間インデックスである。

**CMSIndex \*pIndex;**

インデックス変数へのポインタである。

**CPDESpaceVars \*pSpaceVars;**

空間インデックス変数へのポインタである。

**CPDESpaceVar \*pSpaceVar;**

空間インデックス変数へのポインタである。

**CPDETimeVar \*pTimeVar;**

時間インデックス変数へのポインタである。

**BOOL m\_bDebug;**

デバッグバージョンかどうかを示すフラグである。

**BOOL AddTailFile(const char \*buf);**

ファイルに文字列を追加するメンバー関数である。

**BOOL LineFeed();**

改行コードを挿入するメンバー関数である。

**BOOL FillIndent();**

行の先頭にインデントを挿入するメンバー関数である。

**void SortDep();**

従属変数をソートするメンバー関数である。

**void ProgGen();**

プログラム生成を行うメンバー関数である。

**void ProgGenDebug();**

デバッグバージョンのプログラム生成を行うメンバー関数である。

**void InitDebugGen();**

デバッグバージョンのプログラム生成を行う際に特別な初期化処理を行う必要があります。その初期化ルーチンを生成するメンバー関数である。

**void DetailGen(CObject \*);**

プログラムの内容を生成するメンバー関数である。

**BOOL Do(CObject \*);**

プログラム生成という仕事全体を行うメンバー関数である。

クラス **CJDFract2Real base->CJobDoer**

**BOOL m\_isUnderDepVar;**

?

**CJDFract2Real();**

?

**BOOL Do(CObject \*);**

生成型から構成された分数型を実数型の分数型に変換するメンバー関数である。

... in JD\_UTL.H

クラス **CtargetDep**

ソルバーセットの一番最初の子でターゲットとなる従属変数の情報を格納する変数であり、自動生成されるソルバーはこの従属変数を求めるためのルーチンである。

**CMSPhys \*m\_theDep;**

解くべきターゲットの変数へのポインタを入れる変数である。

**CMSIndex \*m\_timeIndex;**

解くべきターゲットの変数の時間インデックスである。

**CMSFract m\_timeShift;**

解くべきターゲットの変数の時間のシフト (n+1, n+1/2, n+2)

**int m\_number;**

?

**BOOL Copy(CTargetDep \*);**

ノードの複製を作成するメンバー関数である。

**BOOL IsThis(CSymbInst \*);**

ターゲットが与えられたシンボルと同一かを判断するメンバー関数である。

**BOOL IsIncluded(CSymbInst \*);**

与えられうシンボルがこのノード内に含まれているかを判断するメンバー関数である。

クラス **CJDAlgManip** base->CJobDoer

**BOOL m\_bStartFg;**

仕事がスタートしたことを示すフラグである。

**int m\_cManipCnt;**

?

**CMSFract m\_shift[4];**

インデックスのシフト量を示す配列変数これが分数型の分子を意味しており、分母はいつも2になっていることを前提している。

**CTargetDep m\_target;**

ターゲットとなる特べき変数を入れる変数である。

クラス **CJDCheckIfEqual** base->CJDAlgManip

数式が等しいかどうかをチェックするクラスで実際に仕事するメンバー関数は **Do** 関数である。

**BOOL Do(CObject \*);**

数式が等しいかどうかをチェックするメンバー関数である。

**CViecFC \*m\_checked;**

処理中に用いられるフラグである。

**BOOL m\_bResult;**

等しいかどうかの結果を示すメンバー変数である。

... in JD\_PRC.H

クラス **CJDCreateMainProc** base->CJDAlgManip

メインルーチンを作成クラスである。

**CPDEMtxDefs \*pMtxDefs;**

プログラム内で用いられるすべての行列の定義を集めるクラスへのポインタである。

**CPDEAbstProcVis \*pVsBox;**

ソルバーの実行される順序を可視化するボックスのオブジェクトへのポインタである。

**POSITION pos;**

?

**CPDEProcess \*pPrc;**

ある特定のプロセスへのポインタを示すメンバー変数である。

**CPDEObject \*pObj;**

ある特定の処理結果へのポインタを示す変数である。

**CPDEDepInfo \*pDep;**

従属変数へのポインタへのポインタである。

**CMSIndex \*pTimeIndex;**

時間インデックスへのポインタを示す。

**CPDEProblem \*pPrb;**

親ノードで問題全体のノードへのポインタを示すメンバー変数である。

**CPDESolvSet \*pSet;**

ソルバーセットへのポインタを示すメンバー変数である。

**CJDCheckDepend check;**

依存関係をチェックするクラスである。

**CSymbInst \*pV, \*pN, \*pN2;**

?

**CMSFract \*pF;**

?

**CMapPtrToPtr checkedAlready;**

?

**void \*plvd;**

?

**CPDEObtainedVars \*pVars;**

?

**CPDEInvokedProcs \*pVPrc;**

?

**int i, oldDepNo;**

古い時間ステップの従属変数の数を示す。

**CViecFC \*pITx;**

?

**CPDEObject \*pPrg;**



生成されるプログラムオブジェクトへのポインタを示す変数である。

**CPDERunSpread \*pRun;**

プログラム生成プロセスクラスそのものでソルバーセットごとに子ノードを持ち、それぞれのノード

**CPDProc \*pMain, \*pSubst, \*pOutput, \*pInit;**

メインプログラム、代入サブプログラム、出力サブプログラム、初期化サブプログラムへのポインタを示す変数である。

**void MakeSubstProc(CViecFC \*);**

代入サブプログラムを生成するメンバー関数である。

**void MakeOutputProc(CViecFC \*);**

計算結果を出力（格納）するサブプログラムを生成するメンバー関数である。

**void MakeInitProc(CViecFC \*);**

初期化サブプログラムを生成するメンバー関数である。

**BOOL Do(CObject \*);**

プログラム生成全体の処理を行うメンバー関数である。

**enum SOLVER\_TYPE**

ソルバーの種類を種別する列挙型の定義である。

**SOLVER\_EXPLICIT,**

簡単な陽解法

**SOLVER\_FULL\_MATRIX\_GAUSS,**

フルな行列を用いてガウス消去法で解く方法

**SOLVER\_BAND\_MATRIX\_GAUSS,**

バンド行列を用いてガウス消去法で解く方法

**SOLVER\_BAND\_MATRIX\_ADI,**

バンド行列を用いて ADI で解く方法

**SOLVER\_FULL\_MATRIX\_LU,**

フル行列を用いて LU 分解で解く方法

**SOLVER\_SOR,**

SOR 法

クラス **CJDCreateProc base->CJDAlgManip**

サブプログラムを生成するクラスである。

**BOOL Do(CObject \*);**

サブプログラム全体を生成するメンバー関数である。

**CPDESolvSet \*pSlv;**  
ソルバーセットへのポインタを格納する変数である。

**CPDETarget \*pTg;**  
ターゲット変数へのポインタを格納する変数である。

**CPDEBoundCond \*pBcs[20];**  
境界条件へのポインタを格納する配列変数である。

**CPDEDepPDE \*pPde;**  
現象を支配する偏微分方程式クラスへのポインタを格納する変数である。

**CPDERange \*pRg;**  
支配方程式が効いている範囲クラスへのポインタを格納する変数である。

**CSymbInst \*pEq;**  
数式へのポインタ

**int dim;**  
次元数を示す変数である。

**int m\_nLeftWing, m\_nRightWing, m\_nShift[3];**  
求まる行列の形を表す左ウィングと右ウィングとそのシフト量を示す整数変数である。

**int rangeBegin[3], rangeEnd[3];**  
支配方程式を解く物理量の範囲を示す整数変数である。

**CPDProc \*pSlvProc;**  
ソルバーサブプログラムへのポインタを格納する変数である。

**CViecFC \*m\_output;**  
結果出力サブプログラムへのポインタを格納する変数である。

**CString m\_name;**  
名前の文字列である。

**int m\_varNo, m\_midPos;**  
変数の数、離散化された支配方程式の何番目の項が行列バンドの真ん中にとるかの数値である。

**CSymbInst \*m\_pVarLst[20];**  
変数のリスト

**CString m\_nameLst[20];**  
名前のリスト

**CPDVar \*m\_pVars;**  
変数群

**CJDCreateMainProc \*m\_pMainProcGen;**  
 メインサブプログラムの生成クラスへのポインタ

**CMSPDEMTx \*m\_pAm, \*m\_pBv, \*m\_pFv;**  
 連立方程式  $A*B=F$  のそれぞれ

**CString sAmtx, sBvct, sFvct;**  
 連立方程式  $A*B=F$  のそれぞれのベクトルの名前

**SOLVER\_TYPE m\_eSolverType;**  
 ソルバーで用いられる解法の種類

**CString sMax[3], sMtxISz, sMtxJSz, sLeftWg, sRightWg, sDiagCol;**  
 連立方程式の行列のサイズ、形、性質を示す変数の名前である。

**CPDEConstDef \*pMxISz, \*pMxJSz, \*pLeftWg, \*pRightWg, \*pDiagCol;**  
 連立方程式の行列のサイズ、形、性質を示す変数である。

**BOOL FindStartCond(CSymbInst\* pInst);**  
 ?

**BOOL CreateMtxSlvSet(CViecFC \*pPrnt, CSymbInst \*pEq);**  
 行列群の定義オブジェクトを作成するメンバー関数である。

**BOOL CreateFullMtxSlvNum(CViecFC \*pPrnt, CSymbInst \*pEq);**

**BOOL CalcMtxNoEqFullMtx(CViecFC \*pPrnt, const char \*nm);**  
 ?

**BOOL CalcSkipWdFullMtx();**  
 ?

**void InitFullMatrix();**  
 フル行列の初期化ルーチンを作成するメンバー関数である。

**BOOL CreateMtxSlv(CViecFC \*pPrnt, CViecFC \*pL, CViecFC \*pR);**  
 ?

**BOOL FullMatrixGaussSlv(CViecFC \*);**  
 ガウス消去法でフルマトリックスを解くサブプログラムを作成するメンバー関数である。

**BOOL CreateBandMtxSlvNum(CViecFC \*pPrnt, CSymbInst \*pEq);**  
 ?

**BOOL CalcMtxNoEqBandMtx(CViecFC \*pPrnt, const char \*nm);**  
 ?

**BOOL CalcSkipWdBandMtx();**  
 ?

```
void InitBandMatrix();
```

?

```
BOOL CreateBandMtxSlv(CViecFC *pPrnt, CSymbInst *pL);
```

ガウス消去法でバンドマトリックスを解くサブプログラムを作成するメンバー関数である。

```
BOOL BandMatrixGaussSlv(CViecFC *);
```

ガウス消去法でバンドマトリックスを解くサブプログラムを作成するメンバー関数である。

... in JD\_DSC.H

クラス CJDDescretizer base->CJDAlgManip

離散化を行うクラス

```
CPDEBoundDef *m_pBoundDef;
```

境界定義オブジェクトへのポインタである。

```
BOOL Do(CObject *);
```

離散化全体を行うメンバー関数である。

```
BOOL FindStartCond(CSymbInst* pInst);
```

前準備するメンバー関数である。

```
BOOL Copy(CJDDescretizer *);
```

入力の式の複製を作成するメンバー関数である。

```
AF_ENUM AnalyseForm(CSymbInst *pInst, CMapStringToOb  
&pParam);
```

入力された数式の形を解析しどの形を取っているかを種別しその種類を返すメンバー関数である。

```
EXP_ENUM AnalyseExpForm(CSymbInst *pInst,  
CMapStringToOb& );
```

入力された数式の形を解析しどの形を取っているかを種別しその種類を返すメンバー関数である。

```
SIMP_ENUM AnalyseSimpForm(CSymbInst *pInst,  
CMapStringToOb& );
```

入力された数式の形を解析しどの形を取っているかを種別しその種類を返すメンバー関数である。

```
BIND_ENUM AnalyseBindForm(CSymbInst *pInst,  
CMapStringToOb& );
```

入力された数式の形を解析しどの形を取っているかを種別しその種類を返すメンバー関数である。

```

    FACTOR_FORM AnalyseFactForm(CViecFC *pInst, CViecFC*,
    CMapStringToOb&);
        入力された数式の形を解析しどの形を取っているかを種別しその種
        類を返すメンバー関数である。
    BOOL MakeDepDisc(CSymbInst *, CViecFC *);
        ?
    BOOL DiscDerivC1(CSymbInst* pInst, CMapStringToOb& para);
        ?
    BOOL DiscDerivC2(CSymbInst* pInst, CMapStringToOb& para);
        ?
    BOOL DiscDerivTime1(CSymbInst* pInst, CMapStringToOb&
para);
        ?
    BOOL ShiftVar(CSymbInst* pInst);
        ?
    BOOL DiscUpwndFstOrd(CSymbInst* pInst, CMapStringToOb&
para);
        ?
    BOOL ShiftVarToPos(CSymbInst* pInst);
        ?
    BOOL ExpPlusInPlus(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL ExpPlusInMult(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL ExpMultInMult(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL ExpNegInMult(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL ExpPlusInNume(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL ExpPlusInNeg(CSymbInst*& pInst, CMapStringToOb&

```

```

para);
    ?
    BOOL ExpNegInNeg(CSymbInst*& pInst, CMapStringToOb& para);
    ?
    BOOL ExpDepInMult(CSymbInst*& pInst, CMapStringToOb&
para);
    ?
    BOOL ExpDepInNegMult(CSymbInst*& pInst, CMapStringToOb&
para);
    ?
    BOOL ExpDepInNumeMult(CSymbInst*& pInst,
CMapStringToOb& para);
    ?
    BOOL ExpDepInNegNumeMult(CSymbInst*& pInst,
CMapStringToOb& para);
    ?
    BOOL BindPlusAbleForm1(CSymbInst*& pInst, CMapStringToOb&
para);
    ?
    BOOL BindPlusAbleForm2(CSymbInst*& pInst, CMapStringToOb&
para);
    ?

    BOOL SimpPlusAbleForm2(CSymbInst*& pInst,
CMapStringToOb& para);
    ?
    BOOL SimpMultAbleForm1(CSymbInst*& pInst,
CMapStringToOb& para);
    ?
    BOOL SimpDivAbleForm1(CSymbInst*& pInst, CMapStringToOb&
para);
    ?
    BOOL SimpDivAbleForm2(CSymbInst*& pInst, CMapStringToOb&
para);
    ?
    BOOL SimpPlusInt0(CSymbInst*& pInst, CMapStringToOb& para);

```

```

        ?
    BOOL SimpMultInt0(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL SimpMultInt1(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
    BOOL SimpDivBy1(CSymbInst*& pInst, CMapStringToOb& para);
        ?
    BOOL SimpNegInt(CSymbInst*& pInst, CMapStringToOb& para);
        ?
    BOOL SimpInvInv(CSymbInst*& pInst, CMapStringToOb& para);
        ?
    BOOL SimpNegDivNeg(CSymbInst*& pInst, CMapStringToOb&
para);
        ?
クラス CJDDescDeriv base->CJDDescrretizer
    BOOL Do(CObject *);
        ?
クラス CJDDescUpwndFstOrd base->CJDDescrretizer
    BOOL Do(CObject *);
        ?
クラス CJDShiftToPos base->CJDDescrretizer
    BOOL FindStartCond(CSymbInst* pInst);
        ?
    BOOL Do(CObject *);
        ?
クラス CJDMapToFit base->CJDDescrretizer
    BOOL Do(CObject *);
        ?
クラス CJDCheckDepend base->CJDDescrretizer
    CTargetDep targetVar[40];
        ?
    int targetNo, checkNo;
        ?
    BOOL CanThisCalc(CSymbInst *pI);

```

```

        ?
        BOOL Do(CObject *);
        ?
クラス CJDCheckIfIncluded base->CJDAlgManip
        CViecFC *m_checked;
        ?
        BOOL m_bResult;
        ?
        CJDCheckIfIncluded();
        ?
        BOOL Do(CObject *);
        ?

... in JD_ALG.H
クラス CJDSimpEqt base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDExpEqt base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDExpEqt2 base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDBindEqt base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDExpEqtNum base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDBindEqtNum base->CJDDescretizer
        BOOL Do(CObject *);
        ?
クラス CJDFormEqt base->CJDDescretizer
        SOLV_TYPE m_solvType;
        ?
        BOOL FindStartCond(CSymbInst* pInst);

```



```

    ?
    SOLV_TYPE SetSolvForm(CSymbInst *, BOOL=TRUE);
    ?
    BOOL FormImplicit(CSymbInst *pEq1);
    ?
    BOOL FormExplicit(CSymbInst *pEq1);
    ?
    BOOL FormExplicitBC(CSymbInst *pEq1);
    ?
    BOOL Do(CObject *);
    ?

```

... in VAR\_WTC.H

クラス CPDE3DimVar base->CPDEBase

```
CPDE3DimVarSP *m_pSP;
```

?

```
CPDE3DimVarCP *m_pCP;
```

?

```
void Create();
```

?

```
BOOL SendMsg(int msg, CObject* el);
```

あるノードにメッセージを送信するメンバー関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extent に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関

数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL CheckValue(CPDEObjPrg \*);**

?

クラス **CPDE3DimVarSP** base->CPDEBase

**CPDE3DimVar \*m\_pV;**

?

**BOOL m\_bAbsoluteValue;**

?

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDE3DimVarCP** base->CPDEBase

**CMSCInteger \*m\_pI, \*m\_pImax, \*m\_pDI;**

?

CMSCInteger \*m\_pJ, \*m\_pJmax, \*m\_pDJ;

?

CMSCReal \*m\_pmaxVal, \*m\_pzeroVal;

?

CPDEBttn \*m\_pmaxCol, \*m\_pzeroCol, \*m\_pvalCol;

?

CViecFC \*m\_pVarName;

?

int m\_nVarNo;

?

void Create();

?

void InitVal();

?

BOOL SendMsg(int msg, CObject\* el);

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdcはデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、pは親ノードからオフセット座標の参照変数である。

クラス CPDE2DimVar base->CPDEBase

CPDE2DimVarSP \*m\_pSP;

?

CPDE2DimVarCP \*m\_pCP;

?

**void Create();**

?

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc**はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p**は親ノードからオフセット座標の参照変数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL CheckValue(CPDEObjPrg \*);**

?

クラス **CPDE2DimVarSP base->CPDEBase**

**CPDE2DimVar \*m\_pV;**

?

**BOOL m\_bAbsoluteValue;**

?

**CPDE2DimVarSP();**

?

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDE2DimVarCP** base->**CPDEBase**

**CMSCInteger \*m\_pI, \*m\_pImax, \*m\_pDI;**

?

**CMSCInteger \*m\_pJ, \*m\_pJmax, \*m\_pDJ;**

?

**CMSCReal \*m\_pmaxVal, \*m\_pzeroVal;**

?

**CPDEBttn \*m\_pmaxCol, \*m\_pzeroCol, \*m\_pvalCol;**

?

**CViecFC \*m\_pVarName;**

?

**int m\_nVarNo;**

?

**void Create();**

?

**void InitVal();**

?

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

クラス **CPDE1DimVar** base->CPDEBase

**CPDE1DimVarSP \*m\_pSP;**

?

**CPDE1DimVarCP \*m\_pCP;**

?

**void Create();**

?

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これ

によってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

```
int OnAccess(CPoint *pp);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
int OnCommand(UINT);
```

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス `CPDE1DimVarSP` base->`CPDEBase`

```
CPDE1DimVar *m_pV;
```

```
?
```

```
BOOL m_bAbsoluteValue;
```

```
?
```

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

```
int OnAccess(CPoint *pp);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
int OnCommand(UINT);
```

ユーザがノードのアクセスメニューを選択した時にその選択された

コマンドとともに呼ばれるメンバー関数である。

クラス CPDE1DimVarCP base->CPDEBase

CMSCInteger \*m\_pJ, \*m\_pJmax, \*m\_pDJ;

?

CMSCReal \*m\_pmaxVal, \*m\_pzeroVal;

?

CPDEBttn \*m\_pmaxCol, \*m\_pzeroCol, \*m\_pvalCol;

?

CViecFC \*m\_pVarName;

?

int m\_nVarNo;

?

void Create();

?

void InitVal();

?

BOOL SendMsg(int msg, CObject\* el);

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdcはデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、pは親ノードからオフセット座標の参照変数である。

... in PDE\_UTL.H



クラス CMSInteger base->CMSConst

void CopyBody(CViecFC \*);

?

void SerializeBody(CArchive& ar);

?

int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、整数の値を変えることができる。

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdc はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、p は親ノードからオフセット座標の参照変数である。

int m\_value;

?

クラス CMSReal base->CMSConst

double m\_value;

?

void CopyBody(CViecFC \*);

?

void SerializeBody(CArchive& ar);

?

int CalcExt(dirMode dir=dirUp);

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

void OnDraw(CDC \*pdc, const CPoint& p);

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdc はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、p は親ノードからオフセット座標の参照変数である。

int OnCommand(UINT);

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

```
int OnAccess(CPoint *pp);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

```
CMSCReal();
```

?

クラス CMSCTRealSub base->CMSCTReal

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

クラス CMSUnknown2 base->CMSUnknown

```
BOOL SendMsg(int msg, CObject* el);
```

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス CMSFract base->CMSNumber

```
int m_value;
```

?

```
unsigned int m_deno;
```

?

```
CMSFract(int num = 1, unsigned int den = 1);
```

?

```
void gdc();
```

?

```
BOOL IsNeg();
```

?

```
void CopyBody(CViecFC *);
```

?

**void SerializeBody(CArchive& ar);**

?

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**int OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);**

ノードがフォーカスされている状態でキーが押されたときに呼ばれるメンバー関数であり、矢印キーで分母と分子の値を変えることができる。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**BOOL IsEqual(int n, unsigned int d = 1);**

与えられた引数の値と比較して等しいかを判断するメンバー関数である。引数は **n** が分子で **d** が分母の意味をする。

**BOOL Plus(int, unsigned int);**

与えられた引数を元の値に加えるメンバー関数である。引数が分数型の場合である。

**BOOL Plus(int);**

与えられた引数を元の値に加えるメンバー関数である。引数が整数型（分母が1）の場合である。

**BOOL Mult(int, unsigned int = 1);**

与えられた引数を元の値に掛算して結果をキープするメンバー関数である。引数が分数型の場合である。

クラス **CMSRealEx** base->**CMSNumber**

**double m\_value;**

実数の値を維持するメンバー変数である。

**BOOL IsNeg0;**

実数が負かどうかを判断する関数である。

**void CopyBody(CViecFC \*);**

ノードの複製を作成するメンバー関数である。

**void SerializeBody(CArchive& ar);**

ノードを保存するメンバー関数である。

**int CalcExt(dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**void Plus(double d);**

与えられた実数の引数 **d** を元の値に加えるメンバー関数である。

**void Mult(double d);**

与えられた実数の引数 **d** を元の値に掛算し保存するメンバー関数である。

クラス **CMSPDEx** base->**CMSMatrix**

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

クラス **CMSPDEVct** base->**CMSPDEx**

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これ

によってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

クラス `CPDEBttm` base->`CViecFC`

```
COLORREF m_color; h
```

色を設定するメンバー変数である。

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

```
int OnFocus(CPoint &p);
```

与えられた座標がノードのフォーカス範囲内に位置しているかを判断するメンバー関数である。

```
int OnDrop(const CPoint& last);
```

ドラッグされている場合ドロップされたときに最後の座標と共に呼ばれるメンバー関数である。

クラス `CEFile` base->`CViecFC`

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。`pdc` はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、`p` は親ノードからオフセット座標の参照変数である。

```
CRect ToCalcExt(CSymbInst*, dirMode d=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、`m_extent`に保存するメンバー関数である。

```
void ToDraw(CSymbInst *pInst, CDC *pdc, const CPoint& shift);
```

参照型のメンバー関数でリンクされた状態でインスタンスの描画を行うメンバー関数である。インスタンスが現れる姿を自分で描画しないで自分がリンクしているノードに負かせて描画してもらうのである。

```
... in PDE_PRG.H
```

```
typedef struct
```

```
    short int VarNo;
```

```
    char pszStatus[22];
```

```
    struct
```

```
        char varName[16];
```

```
        short int type;
```

```
        short int size[3];
```

```
        unsigned long pBody;
```

```
    } varList[1];
```

```
} VAR_STRUCT;
```

```
クラス CPDEObjPrg base->CPDEObject
```

```
    BOOL m_bGen, m_bBreaking;
```

```
    ?
```

```
    CString sPathName, sFileName, sFileExt, sFileTitle, sPath;
```

```
    ?
```

```
    STARTUPINFO startupInfo;
```

```
    ?
```

```
    PROCESS_INFORMATION procInfo;
```

```
    ?
```

```
    HANDLE hVar, hSig, hThr;
```

```
    ?
```

```
    VAR_STRUCT *pVar;
```

```
    ?
```

```
    CString sSig, sVar, sBrk;
```

```
    ?
```

**CStringArray strAry;**

?

**int m\_cmd;**

?

**CRITICAL\_SECTION memSync;**

?

**CPDBrkPnt \*m\_pBrkPnt;**

?

**CPDEBrkObj \*m\_pBrkObj;**

?

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**void SerializeBody(CArchive& ar);**

?

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc**はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p**は親ノードからオフセット座標の参照変数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL SendMsg(int, CObject\*);**

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL SetMenuForPDFor(CMenu \*pMenu);**

?

**BOOL OnCommandForPad(int msg, CPad \*pPad);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**BOOL OnCommandForSymbInst(int msg, CSymbInst \*pPad);**

?

**void ToFindCandAndSelect(CViecFC \*, CRuntimeClass \*);**

?

**BOOL AddFileToElem(CViecFC \*, int no);**

?

クラス **CPDEBrkObj** base->CPDEBase

**CPDEBttm \*m\_resetBtn, \*m\_onOffBtn, \*m\_trigBtn;**

?

**HANDLE hBrk;**

?

**BOOL m\_bBreak;**

?

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extentに保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdcはデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、pは親ノードからオフセット座標の参照変数である。



**CRect ToCalcExt(CSymbInst\*, dirMode d=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void ToDraw(CSymbInst \*pInst, CDC \*pdc, const CPoint& shift);**

参照型のメンバー関数でリンクされた状態でインスタンスの描画を行うメンバー関数である。インスタンスが現れる姿を自分で描画しないで自分がリンクしているノードに負かせて描画してもらうのである。

クラス **CPDBrkPnt** base->CPDSubst

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

クラス **CPDFFile** base->CPad

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

クラス **CPDESngVarBox** base->CPDEBase

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

クラス **CPDESngVar base->CViecFC**

**CStringArray strAry;**

?

**char m\_pszVarName[21];**

?

**int m\_nVarNo;**

?

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent** に保存するメンバー関数である。

**int OnAccess(CPoint \*pp);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdc はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、p は親ノードからオフセット座標の参照変数である。

... in PDE\_ADV.H

クラス CPDEInfo base->CPDEBase

```
CPDEMeshInfo *m_meshInfo;
```

?

```
int CalcExt(dirMode dir=dirUp);
```

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、m\_extent に保存するメンバー関数である。

```
void OnDraw(CDC *pdc, const CPoint& p);
```

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。pdc はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、p は親ノードからオフセット座標の参照変数である。

```
BOOL CreatePDEsInfo();
```

?

```
BOOL OnCreated(EnumOnCreated = SKIP_IF_CREATED);
```

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

```
BOOL SendMsg(int msg, CObject* el);
```

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

```
int OnAccess(CPoint *p = NULL);
```

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

クラス CPDERange base->CPDEBase

```
int m_rangeNo;
```

?

**CRangeOnAxis \*m\_range[3];**

?

**CPDERange();**

?

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

クラス **CPDEProblem base->CPDEBase**

**CPDEInfo \*m\_pdeInfo;**

?

**int OnAccess(CPoint \*p = NULL);**

ユーザによるノードへのアクセスが起こったときに（マウス右ボタンが押されたとき）呼ばれるメンバー関数である。

**int OnCommand(UINT);**

ユーザがノードのアクセスメニューを選択した時にその選択されたコマンドとともに呼ばれるメンバー関数である。

**int CalcExt(dirMode dir=dirUp);**

ノードの表示される範囲（長方形）を計算するメンバー関数。これによってノードの画面上の占める長方形の領域が計算し、**m\_extent**に保存するメンバー関数である。

**void OnDraw(CDC \*pdc, const CPoint& p);**

自分のノードをウィンドウ上に現れるべき姿で描画するメンバー関数である。**pdc** はデバイスコンテキストで描画に必要な情報やリソース（ペン、ブラシ、フォント）を持つオブジェクトへのポインタであり、**p** は親ノードからオフセット座標の参照変数である。

**BOOL SendMsg(int msg, CObject\* el);**

送られるメッセージを処理する関数である。2 つの引数とともに呼ばれる。一つ目は整数型でメッセージそのものを意味する。送られる側はこの整数値でメッセージを判断する。2 つ目の引数はそのメッセージで用いられるオブジェクトへのポインタである。

**BOOL OnCreated(EnumOnCreated = SKIP\_IF\_CREATED);**

ノードが生成された後に呼ばれるメンバー関数で内部の初期化を行うメンバー関数である。引数自ノードだけか子ノードまで初期化するかを示す列挙型の変数である。

### 3. ノードの基本的な親子構成

#### CPDEProblem

一番トップのノードで PDE 問題全体を格納するためのコンテナ  
3 つのメンバーから構成される : CPDEInfo (問題入力)、  
CPDERunSpread (プログラム生成の全プロセスを格納するコンテナ)、  
CPDEObjPrg (作成される計算プログラムを格納するコンテナ)

#### CPDEInfo

PDE 問題の入力オブジェクトとして問題を解決するための必要な入力情報を格納するコンテナである。これのメンバーとしては、メッシュお呼び空間変数情報の CPDEMeshInfo、時間変数情報の CPDETimeVar、複数の物理量などの従属変数の定義 CPDEDepInfo、パラメーターのフォルダー CPDEParaInfo、初期ルーチン CPDProc、入力ファイル CPDEFileFold などがある。

#### CPDERunSpread

シミュレーションプログラム生成の全プロセスを表現している。ソルバーセットの数だけ複数のメンバープロセスから構成される。それぞれの子プロセスは CPDEProcess クラスからできている。

#### CPDEObjPrg

生成されるプログラムを格納するコンテナクラスである。

#### CPDEMeshInfo

メッシュお呼び空間関係の情報が含まれる。3 つのノードから構成される ; 空間変数定義 CPDESpaceVars、境界位置 CPDEBoundDefs , メッシュの描画 CPDEMeshMap

#### CPDETimeVar

時間変数定義、時間変数に関する情報や定義が格納されている。時間を

代表する変数 `CMSIndep` を始め、`CPDESpaceVar` と同じようなメンバーから構成される。

### **CPDEDepInfo**

物理量を表す従属変数に関する情報が格納されている。3つ以上のメンバーを持つ。1つ目のメンバーはその物理量を代表する記号の定義で、例えば温度ならば `T`、圧力ならば `P` など。2つ目のメンバーはその物理量を説明する文字列になり、たとえば `Temperature,Pressure`。3つ目のメンバーはその物理量のメッシュ上の定義位置の絵が描画されている。メッシュの真ん中にあるか左壁にあるか速度成分かなどの情報を維持する。4つ目行こうはソルバーに必要な一セットの情報をもち、ソルバーセット `CPDESolvSet` と呼ぶ。

### **CPDESolvSet**

ソルバーを生成するための必要な情報。ターゲットとなる変数、支配偏微分方程式とその支配範囲 `CPDEDepPDE`、境界条件 `CPDEBoundCond` がある。

### **CPDEParaInfo**

計算に用いられる様々なパラメータが格納されるコンテナ

### **CPDESpaceVars**

空間の次元数と同じ個数の空間変数定義 `CPDESpaceVar` から構成される。1次元問題なら1個、2次元問題なら2個

### **CPDEBoundDefs**

境界定義に関する情報が格納されるコンテナである。境界（壁）の数だけの子メンバーを持つ。それぞれの境界定義は `CPDEBoundDef` クラスで表現されている。

### **CPDEBoundDef**

このノードが名前が境界名になる。その子メンバーは1つ以上ある。1つ目のメンバーはその境界がどの位置にあるどの軸に向かっているかつまり壁の向きと位置を表している。2つ目以降のメンバーは壁の次元数による。1次元ならなし。2次元ならば境界線になりその線の始点と終点を表すために1個必要になる。3次元ならば境界ボックス（長方形）になりボックスの範囲を表す2つの個が必要になる。

### **CPDESpaceVar**

空間変数定義、空間変数に関する定義が格納されている。空間を代表する変数 `CMSIndep` (`x`、`y`、`z`)、1メッシュの区間 `CMSIndepConst` (`dx`、`dy`、`dz`)、空間インデックス `CMSIndex` (添え字に用いられる `i,j,k`)、空間の分割数 `CMSMeshSize` (`IMAX=20`)

## 4. NCAS のデータ構造

NCAS のデータ構造は木構造あるいは階層的構造になっている。木構造中にあるすべての要素またはノードがルートノード以外親ノードを持っており、また子ノードをもつことができる。NCAS システムの中のすべてのオブジェクト（数式、問題定義、途中結果、プロセス、。。。）はこのノードの性質を継承している。このノードという基本要素を C++ 言語でのクラスでインプリメントしている。その基本クラスのクラス名は CViecFC という。NCAS システムの中で新しい種類の要素を定義するときはこの基本クラスを継承しなければならない。

以下の例では A というノードに二つの子ノード B, C を持つように A, B, C クラスを定義し親子関係を AddTail メンバー関数で実現している。

```
class A : public CViecFC {...};  
class B : public CViecFC {...};  
class C : public CViecFC {...};
```

```
CViecFC *pA, *pB, *pC;  
pA = new A;  
pB = new B;  
pC = new C;  
pA->AddTail(pB);  
pA->AddTail(pC);
```

NCAS システムのすべてのオブジェクト（ノード）を作成するときに上述した例のように自分のクラスを設計・実装すればよい。上述の例において例えば B オブジェクトの親ノードを求めたいとき pB->GetParent() というようにメンバー関数を呼べばよい。pA の 1 番目の子ノードを取得するときは pA->GetChild(0) というようにメンバー関数を呼べばよい。

## 5. 参照型 Object とインスタンス Object

## 6. メッセージの伝播と SendMsg

## 7. MSG\_DO\_JOB と CJobDoer