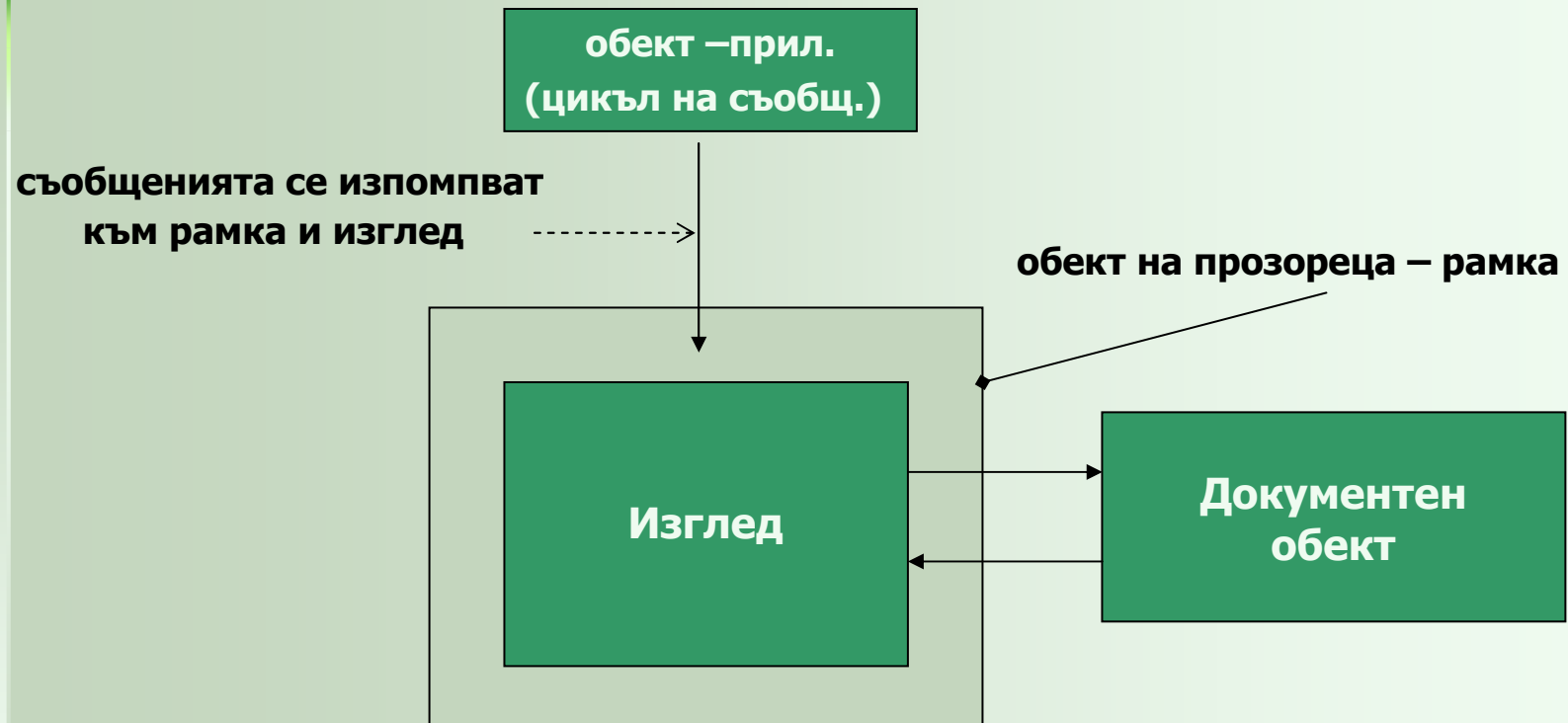


Програмна архитектура Document/View



Document class:

`GetFirstViewPosition();`
`GetNextView(); OnNewDocument(); OnOpenDocument(); Serialize()`

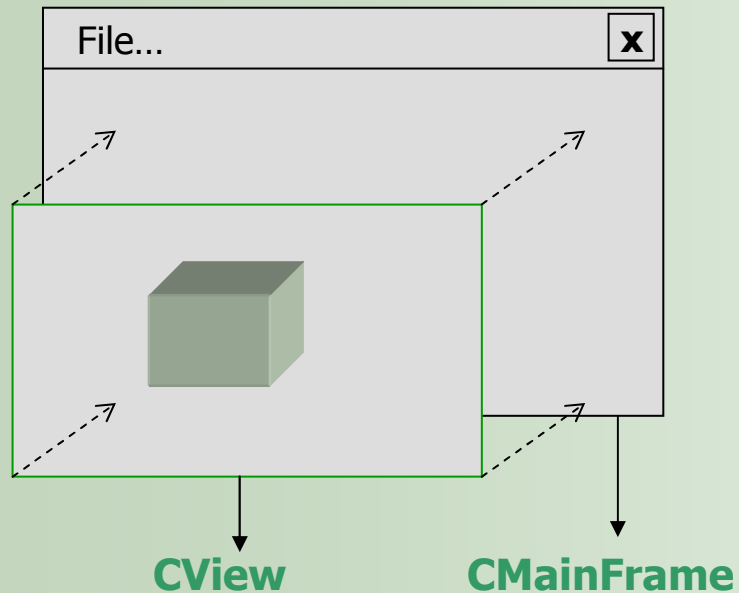
View class

```
CViewCountDoc* pDoc =  
    dynamic_cast<CViewCountDoc*>(GetDocument());  
Добре е в прил. да има собствена реал. на GetDocument():  
ViewCountDoc* CViewCountView::GetDocumet()  
{  
    CViewCountDoc* pDoc =  
        dynamic_cast<CViewCountDoc*>(GetDocument());  
    ASSERT_VALID( pDoc);  
    Return pDoc;  
}  
void CMyView::OnDraw(CDC* pDC)  
{  
    CMyDoc* pDoc = GetDocument();  
    CString string = pDoc->GetString();  
    CRect rect;  
    GetClientRect(&rect);  
    pDC->DrawText( string, rect,.....);  
}  
....OnDraw(); OnInitialUpdate(); OnUpdate()..
```

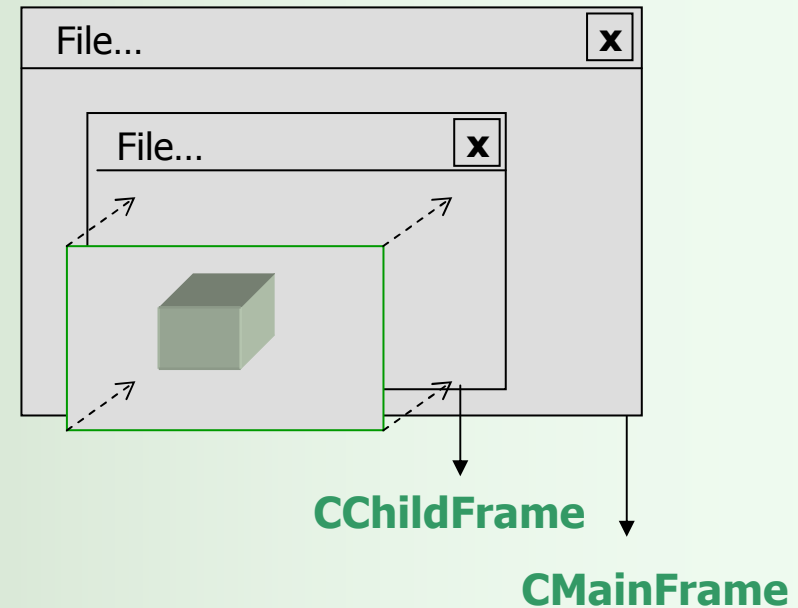
Main Frame & Child Frame Windows

OnClose(); преоразмеряване; работа с toolbar..
Произхождат от CFrameWnd и CMDIFrameWnd

SDI



MDI



можем да модифицираме поведението на MDI child frame (CChildFrame):
`BOOL CChildFrame::PreCreateWindow(CREATESTRUCT& cs) {...}`

**Application class: наследява CWinApp.
Инициализации; стартиране; качване на doc. Template...**

Document Template

Прави връзка м/ду (при това само по 1) document class , view, frame. Има CSingleDocTemplate и CMultipleDocTemplate. Съдържа:

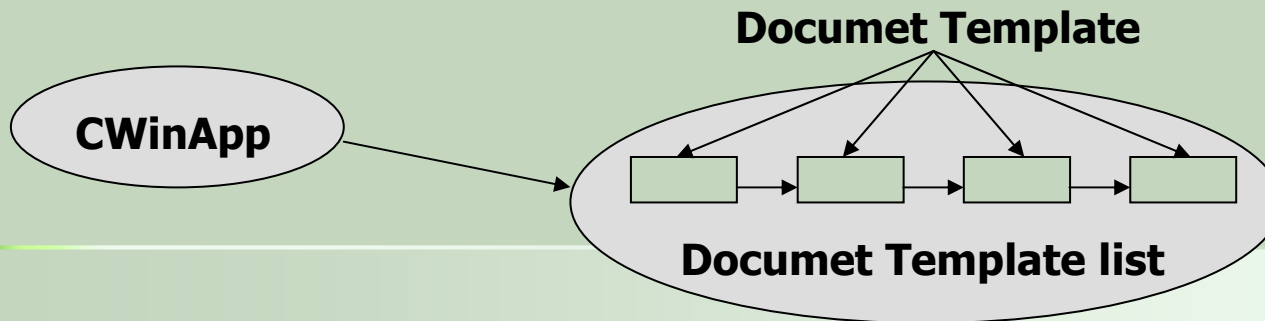
- **документен низ (7 полета),
\nMy_app\My_app\n\n\nMy_app.Document\nMy_app Document**
- **Иконата на изображението**
- **Менюто при активно изображение.**

SDI документния шаблон се създава още в InitInstance():

```
CSingleDocTemplate * pDocTemplate;  
pDocTemplate = new CSingleDocTemplate( IDR_MAINFRAME, RUNTIME_CLASS(  
CMyDoc), RUNTIME_CLASS( CMainFrame), RUNTIME_CLASS( CMyView));  
AddDocTemplate(pDocTemplate);  
// следват редове зарег. на док. типове в REG, създаване обектите от  
шаблона, показване рамката и изгледа.
```

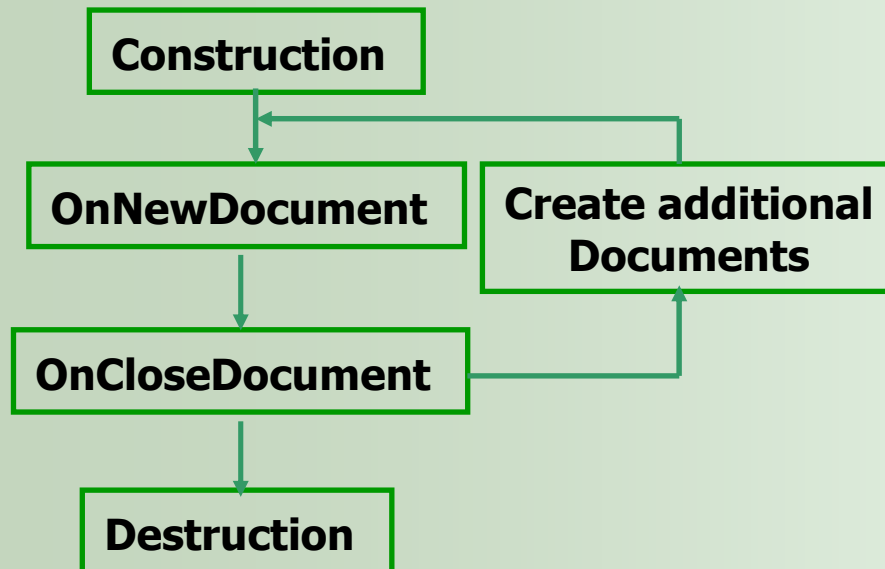
Включва се цикълът за обр. съобщения, като командните съобщения (от меню, акселер. , toolbar) се насочват за обработка към обектите - изглед → документ → шаблон → рамка → приложение → ::DefWindowProc() (command routing).

Ако MDI има повече документи и изображения (CDocManager):

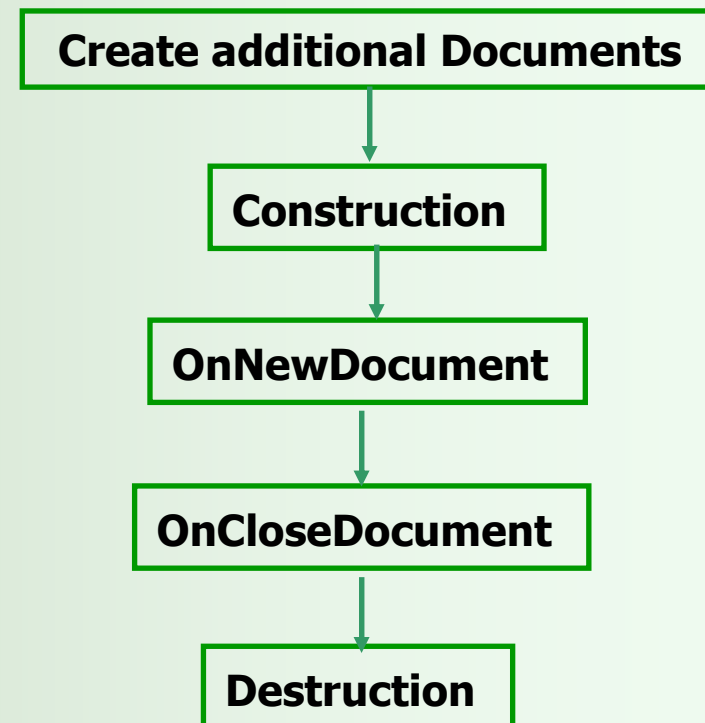


Време на живот на документа и изображението

SDI



MDI

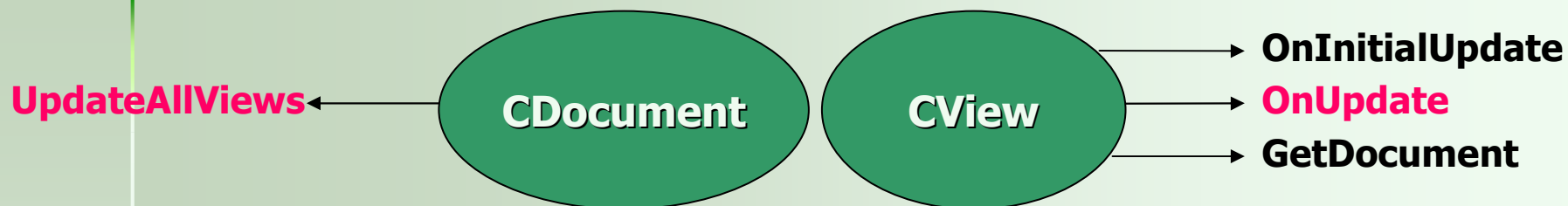


CWinApp::OnFileNew() – създава док от template; **OnNewDocument()**→**DeleteContents()**

CWinAp::OnFileOpen()- диалог;**CDocument::OnOpenDocument()**→**DeleteContents()**,сери.

Updating Document & Views

класовете документи и изображения си взаимодействат:



Основни View класове към MFC

CView – Стандартен дъщерен прозорец за вашето изображение.

CScrollView – Добавя scroll възможности към CView .

CFormView – Предоставя диалог за вграждане на контроли в изображението

CRecordView – Произведен на горния, използва се при работа с ODBC БД – изобразява запис от ODBC база данни.

CDaoRecordView – Подобен на горния, но за работа с DAO .

CEditView – Съдържа edit control, който попълва клиентската област .

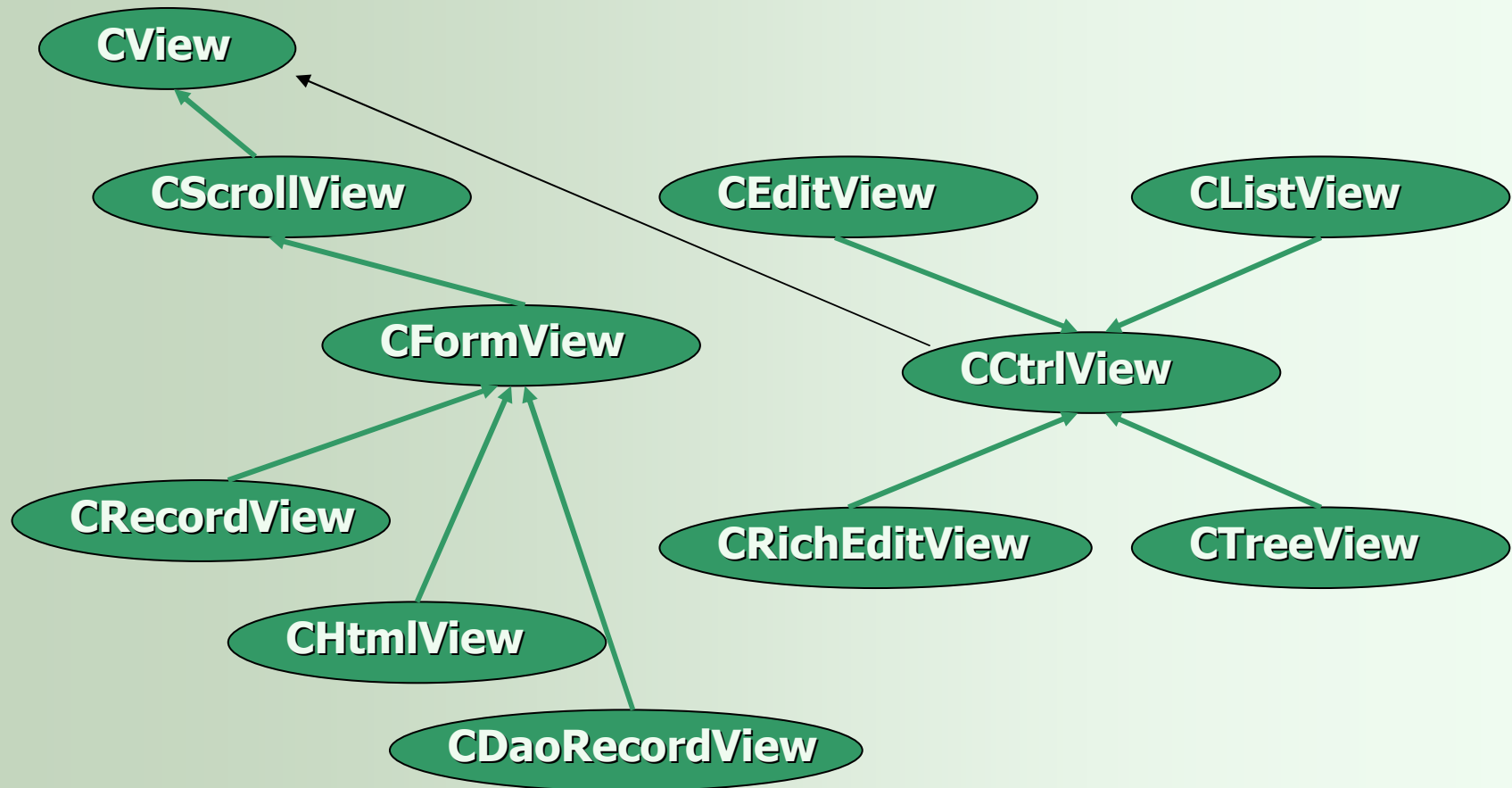
CTreeView – Съдържа tree view control, който попълва клиентската област на изображението.

CRichEditView – Съдържа rich edit control, който попълва клиентската област на изображението.

CListView – Съдържа list control.

CHtmlView – Създава изгледи от html формати и други формати, поддържани от контрола WebBrowser на Microsoft Internet Explorer.

Наследствени връзки между тези класове



CScrollView

```
void CMyView::OnInitialUpdate()  
{  
    CScrollView::OnInitialUpdate();  
    SetScrollSizes(MM_TEXT, CSize(1280,1024)); // +стъпка скролиране  
}
```

Основни ф-ии на класа:

```
CPoint pos = GetScrollPosition();    ScrollToPosition(CPoint(100,100));  
CSize size = GetTotalSize();        // връща лог. размер на изгледа  
int nWidth = size.cx;  
int nHeight = size.cy;  
SetScaleToFitSize(GetTotalSize()); // мащабира лог. изглед  
SetScrollSize(...) // за динамична корекция на лог изглед
```

OnDraw(&dc)

Преди OnDraw(), във ф-ия OnPaint(): **CPaintDC dc(this)**; OnPrepareDC(&dc);
OnDraw(&dc);

Ако искате OnDraw() да прерисува само инвалидизирания правоъгълник.:

```
CRect rect;
```

```
pDC->GetClipBox(&rect); // попълва размера на инвалид. правоъгълник.
```


HTML изгледи

CHtmlView: обвивка на IWebBrowser2

Невиртуални операции (готови методи) на класа:

GetBusy показва дали в момента е download
GetLocatinURL извлича URL адресана текущата страница
GoBack **GoForward** **Refresh**
Stop преустановява зареждането
Navigate по зададен URL адрес изобразява текуща страница

ето пример:

ON_COMMAND(ID_BACK, OnBack) в message map на CMyView
ON_COMMAND) ID_REFRESH, OnRefresh)

...

```
void CMyView::OnBack()  
{ GoBack() }
```

...

Преход към друга страница можете да направите:

```
Navigate( _T(http://www.microsoft.com));
```

“тънки клиенти”:

```
void CHtmlClockView::OnInitialUpdate()  
{ ....Navigate(string);...
```

Дървовидни и списъчни изгледи

CTreeView

CTreeCtrl& CTreeView::GetTreeCtrl();

Интересни методи на класа:

::SetImageList() // свързва текста с картинка
::InsertItem() // вмъкване, корен, подредба, картинка и т.н.
::DeleteItem()
::SetItemText() **::SortChildren()** **::GetSelectedItem()**

CListView: получава функционалност от съответния контрол

Създаване на изгледи, базирани на контроли (наследници на CCtrlView:

- * да се декларира клас, наследник на напр. CCtrlView, като в конструктора на базовия се подава името на създавания.
- * да се опише ф-ия (GetTabCtrl() или подобна) връщаща this към обхващания контрол:

CTabCtrl& GetTabCtrl() const {return (CTabCtrl*) this;}

- * да се предефинират поне 2 метода:

CMyView::PreCreateWindow(..) // на CWnd; указва стил; инициализира библиотеката за общи контроли (::InitCommonControls())

::OnInitialUpdate() // начално изобразяване

- После **GetTabCtrl().метод_на_tab_контрола()**

Създаване на много документи и много изгледи в приложението

MDI - разлики с SDI:

1. Много отворени документи
2. Различни типове документи в MDI
3. MDI поддържа меню-опция Windows...
4. Много менюта в MDI
5. Много прозорци-рамки в MDI

MDI:

- наследява главната рамка не от `CFrameWnd`, а от `CMDIFrameWnd`.
- * наследява дъщерните рамки от `CMDIChildWnd`
- за шаблони на документа се използва не `CSingleDocTemplate` а `CMultiDocTemplate` с указан параметър в конструктора му (където се указваше прозорецът рамка) не главния, а класът на дъщерния прозорец – рамка.
- MDI имат поне два ресурса за меню – когато няма отворен документ и когато има поне 1 отворен.

Превключване на менюта; изгледи, отворени документи... се поемат от генериран код

Различни типове документи в MDI приложение

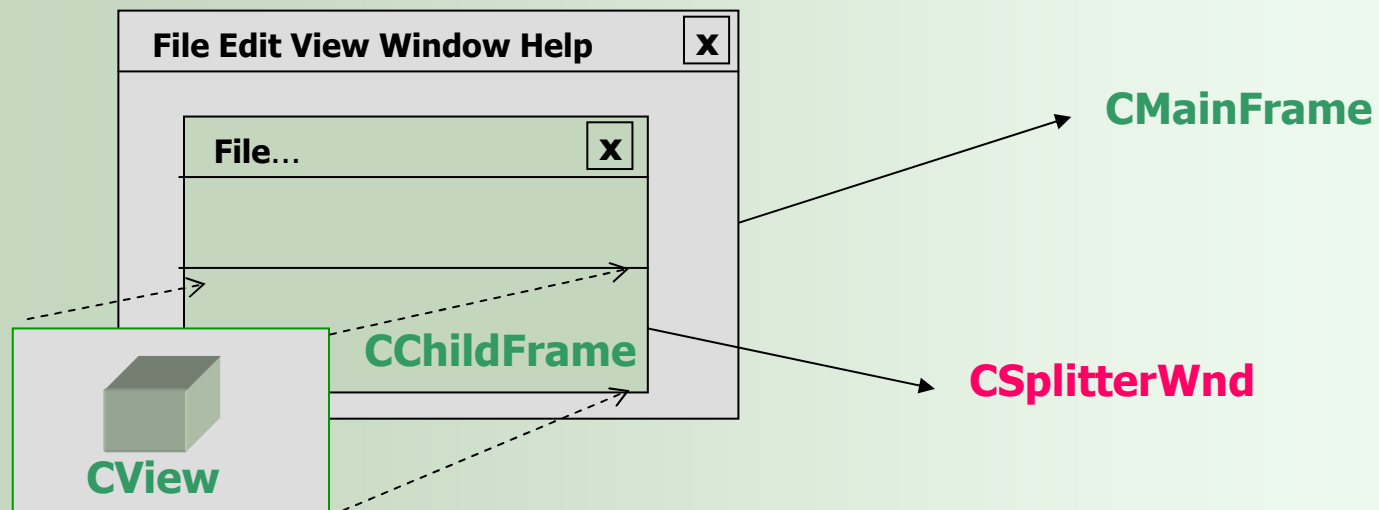
За да създадем прил. Обработващо 2 типа документи(при File→New):

- Създаваме нов клас за документ и нов клас за изглед – например CPiramDoc и CPiramView. Класовете да са динамично създаваеми.
 - За новия тип създайте 4 нови ресурса – икона, меню, таблица с акселератори (ако трябва) и документен низ. Всички те имат общ ID – например IDR_PIRAMTYPE.
 - Ако е нужно се създава и собствена дъщерна рамка за новия тип док.
 - Модифицираме InitInstance() за добавяне новия документен шаблон:
CMultiDocTemplate* pDocTemplate;
pDocTemplate = new CMultiDocTemplate(описаните по-долу 4 парам.);
 - *ресурсен ID*
 - *Указател към CRuntime class (RUNTIME_CLASS()) за документа.*
 - *Указател към CRuntime class за рамката.*
 - *Указател към CRuntime class за изгледа (CPiramView).*
- регистрирате новия шаблон:
AddDocTemplate(pDocTemplate);
- Ако имате повече документни типове – за всеки в InitInstance() подобен код. CMultiDocTemplate има грижа да организира списъка.

Това може да се направи и в SDI, ако искате да работите с повете документни типове (но едва ли е нужно)

Разцепени изображения

Ето връзката между елментите на split изображението



Създаване на приложение, поддържащо динамични сплитери (max 2 части, указани при създаване; имат обща лента за скролиране; по подразбиране се базират на общ view, но ако създадете наследник на CSplitterWnd и предефинирате ::CreateView() – то може и на различен)

1. деклариране на нова пром. към CChildFrame (или CMainFrame):

```
// Implementation
```

```
private:
```

```
CSplitterWnd m_wndSplitter;
```

2. предефиниране на виртуалната OnCreateClient() на рамката :

```
BOOL CChildFrame:: OnCreateClient( LPCREATESTRUCT lpcs,  
CCreateContext* pContext)
```

```
{ CSize size (10, 10);
```

```
Return m_wndSplitter.Create(this, 2, 1, size, pContext);}
```

Параметър1: родителя на прозореца със сплитери – прозорец- рамка

Параметър2: максималния брой панели на които може да се раздели

Параметър4: макс. ширина и височина при които панелите се създават

Параметър5: стандартна структура, едно от полетата ѝ е име на класа на изгледа, използван за панелите, *doc; *doc_template.

3. CSplitView::OnDraw()

CSplitterWnd има методи за придвижване на разделителя

Може създаване вложени прозорци с дин. Сплитери , в прозорец-статичен

Синхронизиране с промените в документа става както беше описано

Създаване на приложение, поддържащо статично split изображение (max 16 X 16, създадени при начално изобразяване; променят се само размерите; базират се на разл. Изображения: пр. Windows Explorer)

- Ако в AppWizard укажете създаване не с MFC-базиран стил на прил., а Windows Explorer базирано, то се структурират 2 статични сплитера
- в класа на прозореца рамка (CMainFrame или CChildFrame):
CSplitterWnd m_wndSplitter;
- предефиниране :
BOOL CChildFrame::OnCreateClient(LPCREATESTRUCT lpc, CreateContext* pContext)
{
if (m_wndSplitter.CreateStatic(this, 2, 1) == FALSE) {..}
// **явно, динамично създаване на изглед** за всеки статичен панел от прозореца:
BOOL bCreated = m_wndSplitter.CreateView(ред_0, кол_0, нач.координати, RUNTIME_CLASS(клас изглед) за панеларазмери на панела);
bCreated = m_wndsplitter.CreateView(1, 0,...);
// координатите са долу ляво, указва се и изглед за втория split view
SetActiveView((CView*) m_wndsplitter.GetPanel(0,0); //активен панел
Return TRUE;
}