

MFC单文档

将内容在内存中画好，
再发送给DC去画
减少频率，解决闪烁

原理

双缓冲

第一步，创建内存
兼容dc

第二步，创建兼容
位图

第三步，内存dc选
择位图

第四步，绘制内存
dc

第五步，复制到实
际DC

```
// CCADView drawing
void CCADView::OnDraw(CDC* pDC)
{
    CCADDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    CRect rc;
    GetClientRect(rc);

    //双缓冲区
    //1.创建一个内存兼容dc
    CDC memDC;
    memDC.CreateCompatibleDC(pDC);

    //2.创建兼容位图
    CBitmap bitmap;
    bitmap.CreateCompatibleBitmap(pDC, rc.Width(), rc.Height());

    //3.内存dc选择位图
    memDC.SelectObject(bitmap);
    CBrush brush(RGB(255, 255, 255));
    memDC.FillRect(rc, &brush);

    //4.绘制
    memDC.SelectObject(::GetStockObject(NULL_BRUSH));

    //遍历链表
    POSITION pos = m_Shapes.GetHeadPosition();
    while(pos)
    {
        IShape *pShape = m_Shapes.GetNext(pos);
        pShape->OnDraw(&memDC);
    }

    //绘制当前的图形
    m_pCurShape->OnDraw(&memDC);

    //5.复制信息到dc
    pDC->BitBlt(0, 0, rc.Width(), rc.Height(), &memDC, 0, 0, SRCCOPY);
}
```

虚基类指针，接受实现类对象
通过虚函数实现了代码不变，却不同的方法
(虚函数)

虚基类与设计

```
class IShape
{
public:
    IShape();
    virtual ~IShape();
    virtual void OnDraw(CDC *pDC) = 0;
    virtual void OnLButtonDown(UINT nFlags, CPoint point) = 0;
    virtual void OnLButtonUp(UINT nFlags, CPoint point) = 0;
    virtual void OnMouseMove(UINT nFlags, CPoint point) = 0;
};
```

设计IShape类
实现类，画的方法多种多样，需要的信息也各不相同
直线需要2个点
但多边形就需要多个点
所以不能用SetPoint()
而退一步使用OnLButtonDown()只提示有鼠标点了，至于实现类
需要多少信息，就调用多少次OnLButtonDown()

劫持窗口消息

```
SetCapture(); //响应窗口外消  
ReleaseCapture(); //释放
```