

# Plugin的设计

定义插件的抽象类型，在DLL中导出分配函数，用于创建插件类的派生类，通过虚函数来执行代码

## 1. 定义的抽象类

```
class ICR34Shape
{
public:
    // 获取菜单名
    virtual const char* GetMenuName() = 0;
    // 画
    virtual int OnDraw(HWND hwnd) = 0;
    // 设置起始坐标
    virtual int SetBeginPoint(POINT pt) = 0;
    // 设置终点坐标
    virtual int SetEndPoint(POINT pt) = 0;
    // 释放，通过自杀的方式
    virtual int Release() = 0;
};

typedef ICR34Shape* (*CREATEOBJ)();

#define CREATE_OBJ(type) extern "C" __declspec(dllexport) ICR34Shape* CR34CreateObject() {return new type;}
```

## 2. 派生类

```
class CLine :public ICR34Shape
{
public:
    virtual const char* GetMenuName()
    {
        return "绘制直线(&L)";
    }

    virtual int OnDraw(HWND hwnd)
    {
        POINT pt;
        HDC hdc = ::GetDC(hwnd);
        ::MoveToEx(hdc, m_ptBegin.x, m_ptBegin.y, &pt);
        ::LineTo(hdc, m_ptEnd.x, m_ptEnd.y);
        return 0;
    }

    virtual int SetBeginPoint(POINT pt)
    {
        m_ptBegin = pt;
        return 0;
    }
}
```

```
virtual int SetEndPoint(POINT pt)
{
    m_ptEnd = pt;
    return 0;
}

virtual int Release()
{
    delete this; //自杀
    return 0;
}
private:
    POINT m_ptBegin;
    POINT m_ptEnd;
};

extern "C" __declspec(dllexport) ICR34Shape* CR34CreateObject()
{
    return new CLine();
}
```