



Windows程序设计与MFC

朱晓旭

苏州大学计算机科学与技术学院

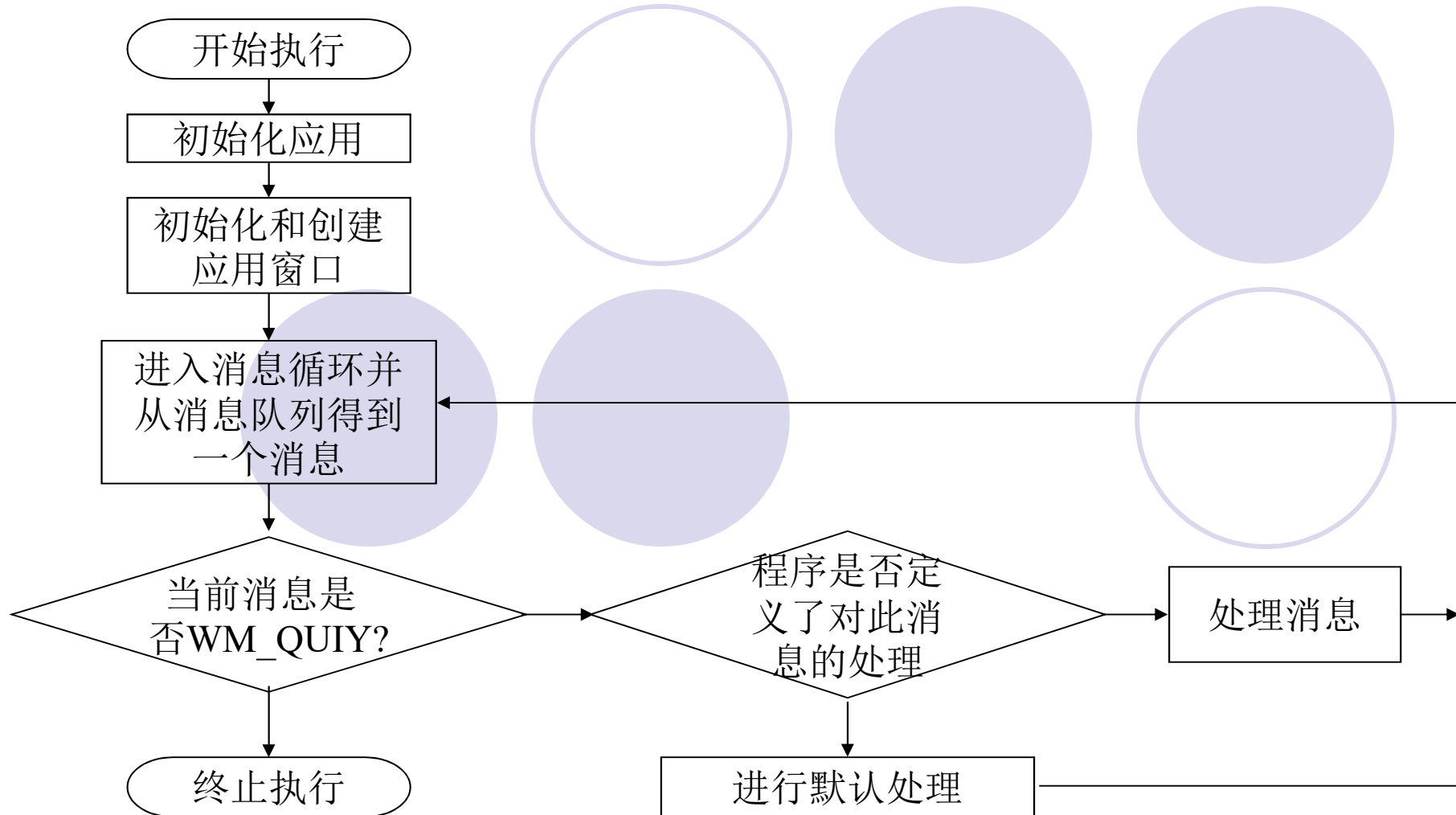
Windows编程基础

- 事件驱动的程序设计
 - 程序无固定流程
 - 以用户为中心的程序设计
 - 消息
 - 资源
- 对象
 - 按钮、菜单、用户数据

最基本的Windows程序

- **WinMain**函数
- 注册窗口类
- 用类创建窗口
- 消息循环

Windows程序的基本结构



Windows常见数据类型

- **HWND**
- **HINSTANCE**
- **HANDLE**
- **POINT与CPoint**
 - 存放鼠标的位置
- **RECT与CRect**
 - 存放矩形区域
- **COLORREF与RGB宏**
 - 颜色

MFC程序设计

- **Microsoft Foundation Class**
- **CObject** 基类之父
- **MFC**类结构复杂、成员函数多
- 对大多数**Windows API**函数进行了封装
- 需要经常阅读类库手册

Windows程序分类

- 基于对话框
 - **Windows**附带的计算器
- 单文档
 - **Windows**附带记事本
- 多文档
 - **Office 97**中的**WORD**
 - **Adobe Photoshop**
- 多个顶级文档
 - **Word 2013**

单文档应用程序

- 用**MFC**实现单文档跟随鼠标画圆程序
 - 文档与视图分离
 - 捕获鼠标事件
 - **CDC**类
 - 请求刷新**InvalidateRect()**;
 - **OnDraw**成员函数
 - 典型的虚函数
 - 选择系统的画刷

- 不在**OnDraw**函数中画圆

```
CRect rc;
```

```
rc.left =point.x -20;
```

```
rc.top =point.y-20;
```

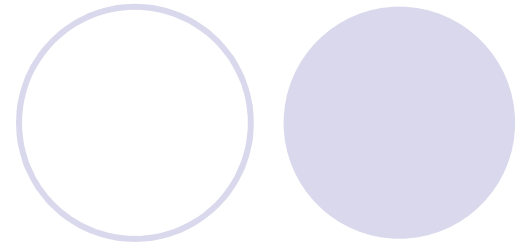
```
rc.right =point.x +20;
```

```
rc.bottom =point.y+20;
```

```
CClientDC dc(this);
```

```
dc.SelectStockObject (LTGRAY_BRUSH);
```

```
dc.Ellipse(&rc);
```



基于对话框的程序

- 在程序中使用对话框
 - 有模式的对话框
 - 无模式的对话框
- 基于对话框的程序
- 控件：使程序具有统一外观
- **DDX**技术
- **UpdateData**（）函数
 - **TRUE** 接受数据
 - **FALSE** 更新控件
- 编写一个简单计算器程序

```
HWND hWnd;  
GetDlgItem(IDC_EDIT1, &hWnd);  
TCHAR buff[128];  
::GetWindowText(hWnd, buff, 128);  
MessageBox(buff);
```

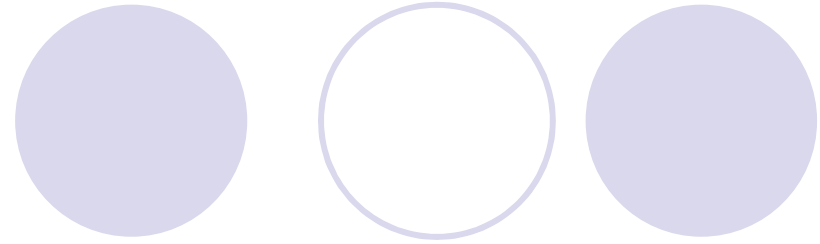
图形设备接口（GDI）

- 应用程序输出与具体设备无关
 - 无需考虑显示卡的种类或打印机的型号
- 直接将输出送到逻辑设备
- 设备环境（**DC**）
- **HDC**与**CDC**（设备环境类）

GDI对象

- 都是**CGdiObject**类的派生类的对象，但从不需要构造**CgdiObject**的对象，它是抽象类
- 常见**GDI**对象
 - 画笔(**CPen**)
 - 画刷(**CBrush**)
 - 字体(**CFont**)
 - 位图 (**CBitmap**)
- 构造了**GDI**对象，在退出程序之前，必须删除
- 为了删除**GDI**对象，必须首先将它从设备环境类中分离出来

画笔使用举例



```
CPen newPen(PS_SOLID ,2,RGB(255,0,0));
```

```
CPen *pOldPen=
```

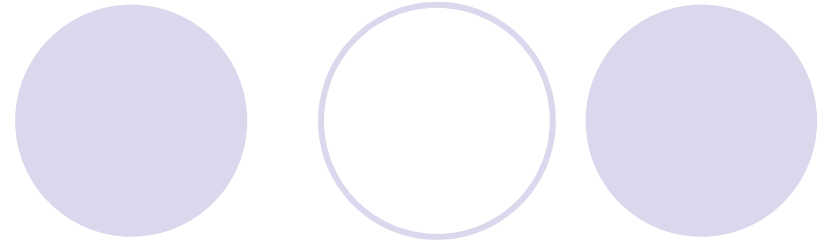
```
    pDC->SelectObject (&newPen);
```

```
pDC->MoveTo (10,10);
```

```
pDC->LineTo (500,500);
```

```
pDC->SelectObject (pOldPen);
```

画刷使用举例



```
CBrush brush1;  
brush1.CreateHatchBrush  
    (HS_CROSS,RGB(0,255,0));  
CBrush *pOldBrush=pDC->SelectObject(&brush1);  
CRect rc(100,100,300,300);  
pDC->Rectangle (&rc);  
pDC->SelectObject (pOldBrush);
```

图片的使用

- CBitmap bmp;
- bmp.LoadBitmap(IDB_BITMAP1);
- BITMAP bitmap;
- bmp.GetBitmap(&bitmap);
- int bmpwidth=bitmap.bmWidth;
- int bmpheight=bitmap.bmHeight;
- CDC memDC;
- memDC.CreateCompatibleDC(pDC);
- memDC.SelectObject(bmp);
- pDC->BitBlt(0,0,bmpwidth,bmpheight,&memDC,0,0,SRCCOPY);

字体使用

```
LOGFONT lf;  
pDC->GetCurrentFont()->GetLogFont(&lf);  
CFont *pOldFont;  
lf.lfCharSet=40;  
lf.lfHeight=-30;  
lf.lfWidth=0;  
wcscpy(lf.lfFaceName, TEXT("隶书"));  
CFont font;  
font.CreateFontIndirect( &lf);  
pOldFont=pDC->SelectObject( &font);  
pDC->TextOut(10,10,TEXT("Haha"));  
pDC->SelectObject(pOldFont);
```


库存**GDI**对象



- Windows预定义的常用 **GDI**
- 有常用画笔、画刷、字体等
- 库存**GDI**对象用**SelectStockObject**选取
- 可以向设备环境中选入库存**GDI**对象来分离用户自定义**GDI**

定时器的使用

- **SetTimer**

- 设置定时器

- **KillTimer**

- 释放定时器

- **OnTimer**

- 处理定时器消息

获取当前时间

- CTime StartTime1 = CTime::GetCurrentTime();
- m_time= StartTime1.Format("%Y年%m月%d日-%H时%M分%S秒");