程序编程：

用鼠标左键拖动一个椭圆在客户端内移动

代码如下：

1. 添加私有成员变量

private:

CRect m\_rect;

CPoint m\_pointMouse;

BOOL m\_capture;

};

1. 在构造函数中进行初始化

CEXAM00View::CEXAM00View()

{

// TODO: add construction code here

m\_rect.SetRect(100,100,300,300);

m\_capture=FALSE;

}

1. 添加OnDraw（）函数

void CEXAM00View::OnDraw(CDC\* pDC)

{

CEXAM00Doc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

pDC->Ellipse(m\_rect);

// TODO: add draw code for native data here

}

1. 添加消息句柄

void CEXAM00View::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

if(m\_rect.PtInRect(point))

{

// SetCapture();

m\_capture=TRUE;

m\_pointMouse=point;

}

CView::OnLButtonDown(nFlags, point);

}

void CEXAM00View::OnMouseMove(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

if(m\_capture)

{

InvalidateRect(m\_rect);

CSize offset(point-m\_pointMouse);

m\_rect+=offset;

InvalidateRect(m\_rect);

m\_pointMouse=point;

}

CView::OnMouseMove(nFlags, point);

}

void CEXAM00View::OnLButtonUp(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

if(m\_capture)

{

::ReleaseCapture();

m\_capture=FALSE;

}

CView::OnLButtonUp(nFlags, point);

}

编程题

本题要求在视图窗口中显示一个椭圆，椭圆的内切矩形存放在视图类的成员变量m\_rectDraw中，其初始值是（0,0,5,5）。当鼠标左键在视图窗口中单击时，椭圆的横纵向各减少5，鼠标右键在视图类窗口中单击时，椭圆的横纵各增加5.当内切矩形的尺寸等于（5,5）时，不再减小。当内切矩形尺寸等于（100,100）时，不再增加。请编写视图类的绘制函数OnDraw(),鼠标左键单击相应函数OnLButtonDown(),鼠标右键单击相应函数OnRButtonDown(),

答题说明：

1. 绘制椭圆的函数是设备上下文类CDC的成员函数Ellipse(LPCRECT lprect)
2. 使一个矩形区域无效并引起重画的操作函数是InvalidateRect(LPCRECT lprect)
3. 使一个窗口无效并引起重画的操作函数是窗口类成员函数InvalidateRect(LPCRECT lprect)

程序代码如下所示：

1. 首先在CVIEW中添加私有变量CRect m\_rectDraw;

private:

CRect m\_rectDraw;

};

2.在构造函数中对m\_rectDraw进行初始化

CEXAM3View::CEXAM3View()

{

m\_rectDraw.SetRect(0,0,5,5);//自己初始化，在构造函数中添加的

// TODO: add construction code here

}

3在CVIEW中添加OnDraw()函数

void CEXAM3View::OnDraw(CDC\* pDC)

{

CEXAM3Doc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

pDC->Ellipse(m\_rectDraw);

// TODO: add draw code for native data here

}

1. 在CVIEW中添加消息句柄

void CEXAM3View::OnRButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

m\_rectDraw.right+=5;

m\_rectDraw.bottom+=5;

if( m\_rectDraw.right>=100)

{

m\_rectDraw.right=100;

}

if( m\_rectDraw.bottom>=100)

{

m\_rectDraw.bottom=100;

}

Invalidate();

CView::OnRButtonDown(nFlags, point);

}

void CEXAM3View::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

m\_rectDraw.right-=5;

m\_rectDraw.bottom-=5;

if( m\_rectDraw.right<=5)

{

m\_rectDraw.right=5;

}

if( m\_rectDraw.bottom<=5)

{

m\_rectDraw.bottom=5;

}

Invalidate();

CView::OnLButtonDown(nFlags, point);

}

注释：红色部分为添加的代码。非红色部分是MFC中本身或自动生成的。

1. 在视图窗口中显示一个椭圆，初始的填充颜色为红色，当鼠标左键在包含椭圆的矩形区域内单击时，椭圆的填充颜色在红色和蓝色之间切换。请编写视图类的绘制函数OnDrAw(),鼠标左键单击相应消息OnLButtonDown.

答题说明：

1. 绘制椭圆函数设备上下文类CDC的成员函数Ellipse(LPCRECT lpRct)
2. 判断一个点是否落入一个矩形中的函数是矩形类CRect的成员函数

PtInRect(Cpoint pt)

(3) 使一个矩形区域无效并引起重画操作的函数是视图类CView的成员函数InvalidateRect(LPCRECT lpRect)

(4) 产生颜色值可以用宏RGB(r,g,b),颜色值的类型是COLO REF

(5) 初始的绘制颜色红色已经在视图类的构造函数中进行了初始化

代码如下

1. 首先在CVIEW中添加私有变量CRect m\_rectDraw;

private:

COLORREF m\_color;

CRect m\_rect;

};

1. 在构造函数中对m\_rect和m\_color进行初始化

CExam0View::CExam0View()

{

// TODO: add construction code here

m\_color=RGB(255,0,0);

m\_rect.SetRect(0,0,300,300);

}

3在CVIEW中添加OnDraw()函数

void CExam0View::OnDraw(CDC\* pDC)

{

CExam0Doc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

// TODO: add draw code for native data here

CBrush brush(m\_color);

CBrush \*OldBrush=pDC->SelectObject(&brush);

pDC->Ellipse(m\_rect);

pDC->SelectObject(OldBrush);

}

1. 在CVIEW中添加消息句柄

void CExam0View::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

if(m\_rect.PtInRect(point))

{

if(m\_color==RGB(255,0,0))

{

m\_color=RGB(0,0,255);

}

else

m\_color=RGB(255,0,0);

InvalidateRect(m\_rect);

}

CView::OnLButtonDown(nFlags, point);

}

注释：红色部分为添加的代码。非红色部分是MFC中本身或自动生成的。