

用的VS2019

可能碰到的问题

1. `CStatusBar` `m_wndStatusBar;`

不要使用所说的CMFCStatusBar，如果你原先这个可能可以。

我使用这个会有异常中断

2.

```
xx=point.x; yy = point.y; //取出坐标信息
sprintf_s(p1, "%4d", xx); //转化为字符串
m_wndStatusBar.SetPaneText(2, CString(p1), TRUE); //在第2个区域显示x坐标
sprintf_s(p1, "%4d", yy); //转化为字符串
m_wndStatusBar.SetPaneText(3, CString(p1), TRUE); //在第3个区域显示y坐标
```

使用_s版本的sprintf

然后SetPaneText里的p1转化为CString，CString可以自动转化为LPCWSTR，还有其他的方法可以解决，主要是一个编码问题。

3.void CMyxxxxxDoc::DDALine(CClientDC* DCPoint)函数的实现中实习指导书的这里注释行里有行代码被注释了，应该是有的。（下面还有这种问题）

```
if (x0 > x1)
{
    x = x0; x0 = x1; x1 = x; //交换起始、终止点
    x = y0; y0 = y1; y1 = x;
}
```

4.这里先将第1、2号点存入数组后面一句又被注释了

```
void CMy2019302130011Doc::Bezier(CClientDC* DCPoint, int mode)
{
    CPoint p[1000]; //设置一个数组存储完整的Bezier曲线控制点
    int i, j;
    i = 0, j = 0;
    p[i++] = group[j++]; //先将第1, 2号点存入数组
    p[i++] = group[j++];
```

5.这个函数下面实现部分里面写的pDC应该是参数里的DCPoint

```
void CMy2018302130018Doc::Bezier_4(CClientDC *DCPoint, int mode, CPoint p1, CPoint p2, CPoint p3, CPoint p4)
{
    int i, n;
    CPoint p;
    double t1, t2, t3, t4, dt;
    CPen pen;
    n = 10;
    if (mode) //mode=1时, 用异或的方式画可擦除的黑色曲线, 调整形状
    {
        DCPoint->SetROP2(R2_NOT);
        pen.CreatePen(PS_SOLID, 1, RGB(0, 0, 0));
    }
}
```

移到起点这里这句话又被注释了。

```
CPen *pOldPen = DCPoint->SelectObject(&pen);
dt = 1.0 / n; //参数t的间隔, 分10段, 即用10段直线表示一段曲线
DCPoint->MoveTo(p1); //移到起点
for(i=1; i<=n; i++) //用Bezier参数方程计算曲线上等间隔的10个点
```

6. 这里有句被注释掉了，然后pDC是DCPoint

```
void CMy2019302130011Doc::GenerateGraph(CClientDC* DCPoint)
{
    group[0].x = 100; group[0].y = 100; //图形数据准备
    group[1].x = 200; group[1].y = 100;
    group[2].x = 200; group[2].y = 200;
    group[3].x = 100; group[3].y = 200;
    group[4].x = 100; group[4].y = 100;
    PointNum = 5;
    DrawGraph(DCPoint); //画图形
}
```

7.这里的pDC是同样的问题

```
void CMy2019302130011Doc::DrawGraph(CClientDC* DCPoint)
{
    int i;
    CPen pen, * pOldPen;
    DCPoint->SetROP2(R2_COPYPEN);
    pen.CreatePen(PS_SOLID, 1, RGB(255, 0, 255));
    pOldPen = DCPoint->SelectObject(&pen);
    DCPoint->MoveTo(group[0]);
    for (i = 1; i < PointNum; i++)
        DCPoint->LineTo(group[i]);
    DCPoint->SelectObject(pOldPen);
}
```

8.同上, 以及p改成seedpoint

```
void CMy2019302130011Doc::SeedFill(CClientDC* pDC, CPoint p)
{
```

9.这句被注释了

```

if (MenuID == 20) { //种子填充:画边界
    if (PressNum == 0) {
        mPointOrign = point;
        mPointOld = point;
        mPointOld1 = point; //记录第一点
        PressNum++;
        SetCapture();
    }
    else {
        ht.MoveTo(mPointOrign); //擦除橡皮筋
        ht.LineTo(point);
        pDoc->group[0] = mPointOrign; //借助DDA直线函数画边界
        pDoc->group[1] = point; // ←
        pDoc->DDALine(&ht);
        mPointOrign = point;
        mPointOld = point;
        PressNum++;
    }
}

```

10.Doc类里的PointNum最好直接初始化或者构造函数里初始化。

不然可能下面的边缘填充中第一个画的多边形显示不出来（因为PointNum的值可能给了一个很奇怪的值，主要还是在OnFillEdge()函数中指导书给的代码没有把PointNum初始化为0。

```

public:
    CPoint group[100]{}; //定义数组
    int PointNum = 0;
    COLORREF m_crColor; //保存图形颜色

```

以下在遇到CPoint名字修改成pDC的就不再提及了

11.这句话被注释掉了

```

void CMy2019302130011Doc::CohenSutherland(CClientDC* pDC, CPoint p1, CPoint p2)
{
    int code1, code2, code, x, y, x1, y1, x2, y2;
    pDC->SetROP2(R2_COPYPEN);
    CPen Pen;
    Pen.CreatePen(PS_SOLID, 2, RGB(255, 0, 0));
    CPen* OldPen = pDC->SelectObject(&Pen);
    x1 = p1.x; y1 = p1.y;
    x2 = p2.x; y2 = p2.y;
    code1 = encode(x1, y1); //对端点编码
    code2 = encode(x2, y2); // ←
    while (code1 != 0 || code2 != 0)

```

12.圆裁剪的代码没给对

鼠标移动部分没告诉要改，裁剪部分没写好，可以修改函数参数，传入圆心和圆上的点（即要半径）

圆裁剪完成后原来的圆的生成无法完成，原因是下面这个函数返回的矩形框的大小为0

```

DCPoint->GetBoundsRect(&rc, 0);

```

需要修改成下面这样


```
CRect rc;  
GetClientRect(AfxGetMainWnd()->m_hWnd, &rc);  
BCircle(DCPoint, &rc, p1, p2);
```