**第三章作业**

将例题3.1 用重新用函数封装，要求如下：

声明函数原型：

ATOM MyRegisterClass(HINSTANCE hInstance);//注册窗口函数

BOOL Create(HInstance, int);创建窗口实例函数

int Run(); //窗口消息循环函数

LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wparam, LPRARAM lparam); //窗口消息处理函数

用以上函数重新写3.1并添加,鼠标左键消息处理：即

WM\_LBUTTONDOWN:

调用自己写的函数作响应。

Void OnLButtonDown(HWND hwnd, UINT message, WPARAM wparam, LPRARAM lparam)

{

MessageBox(hWnd, “Test Success！”, “My Information”, MB\_OK);

}

**第四章作业**

# 实验目的

1. 了解GDI概念
2. 理解图形刷新：什么时候发生，发生时需要做什么
3. 掌握设备环境的使用
4. 掌握映射模式的设置以及各种绘图工具的使用

# 实验内容

## (一) 编写一个简单的画图程序，要求如下：

1. 在例题3.1程序（可以用vc自动生成程序）的基础上，增加画图功能；
2. 在窗口函数中处理鼠标左键单击消息。在该消息处理中，绘制一条直线段，要求线段为实线，宽度为 2 个像素，颜色为淡蓝色(132, 129, 255)。线段端点

为(0, 0)和(100, 100)。（注意，首先需要获取到设备环境，然后再进行绘制）

1. 认真调试，让程序顺利运行，并且结果正确。

## (二) 程序二：刷新处理。要求如下：

1. 在程序**(**一**)**的基础上，完善该程序；
2. 大家调整一下窗口大小，是不是发现直线段没了。这是因为，窗口需要刷新。当需要刷新时，操作系统会向窗口发送 WM\_PAINT 消息，我们需要在 WM\_PAINT 消息中，对刷新请求进行处理。
3. 怎么处理呢？我们可采用最直接的方法：在 WM\_PAINT 消息中，再把直线段绘制一遍（注意，在绘制前，也需要先获取到设备环境）。这样，当窗口需要刷新时，该直线段就会重新绘制一遍。
4. 认真调试，让程序顺利运行，并且结果正确。（再调整一下窗口，结果正确吗？）
5. 思考一下，另外一种处理刷新的方法应该怎样实现：保存当前绘制的副本，在下一次刷新请求时，用副本中的内容填补无效区域。课后，大家在百度上找找，这样的代码应该如何实现。

## (三) 程序三：刷新进阶。要求如下：

(1) 在上述程序的基础上，改进该程序；

### (2) 定义一个 BOOL 型全局变量 bStatus，初如化其值为 TRUE；

(3) 在 WM\_PAINT 消息处理中，当 bStatus 为 TRUE 时，绘制一个矩形，其左上角在(10, 10)处，右下角在(150, 100)处。要求矩形框为虚线，线宽为 1，矩形边框颜色为黑色，内部填充色为(100, 167, 201)。当 bStatus 为 FALSE 时，绘制一条直线段，要求线段为实线，宽度为 2 个像素，颜色为淡蓝色(132, 129,

255)。线段端点为(0, 0)和(100, 100)。

### (4) 在 WM\_LBUTTONDWON 消息处理中，反转 bStatus 的值，即 bStatus = !bStatus

（真变假，假变真）。写完后，运行程序。大家有没发现，窗口中的绘制内容并不会随 bStatus 的值而更新，我们必需调整窗口大小，窗口中的内容才会更新。即只有请求刷新时，窗口才会刷新重绘。

(5) 那有没有无需调整窗口大小请求刷新的方法呢？有的，可通过函数

### InvalidateRect()，大家在 WM\_LBUTTONDOWN 消息处理中调用该函数。发现

没，现在无需调整窗口大小，窗口内容就自动刷新了。函数原型如下：

**BOOL InvalidateRect(**

**HWND** *hWnd*，// 窗口句柄

**CONST RECT\*** *lpRect***,** // 矩形区域

**BOOL** bErase //是否擦除背景

**);**

该函数会放一个 WM\_PAINT 消息到消息队列中，因此相当于让窗口变脏重绘。可

以这样调用该函数：InvalidateRect(hWnd, NULL, TRUE)；关于该函数的具体说明，大家可以去百度。

1. 认真调试，让程序顺利运行，并且结果正确。
2. 总结：绘制代码可以放在 WM\_PAINT 消息处理中，用于处理刷新。而当代码中需要重绘时，可以通过 InvalidateRect()发送一个 WM\_PAINT 消息。

(四) 分别调用系统定义的四种笔样式PS\_DOT, PS\_DASHDOT,PS\_DASHDOTDOT,PS\_DASH画四个圆，看一看有什么差别。然后调用系统定义的6中画刷画出圆角矩形，调用系统定义的6钟画刷来画出圆角矩形。调用函数Pie画一个圆，红黄蓝各占三分之一。

1. （五）**请在屏幕上画一面五星红旗**（选作：飘扬效果？）

（注五边形画法：看所附例题）

附：参考代码

1. 获取设备环境

BeginPaint/EndPaint；GetDC/ReleaseDC。思考：应该如何在两者中选择？

1. 创建及使用画笔 （用于线框绘制）

### CreatePen/SelectObject/DeleteObject。注意：在删除创建的画笔之前，需要恢

复系统的原有画笔

(3) 创建及使用画刷 （用于内部填充）

### CreateSolidBrush/SelectObject/DeleteObject。注意：在删除创建的画刷之前，

需要恢复系统的原有画刷

**C语言复习巩固练习题**

1. C语言复习巩固练习：数组作为函数参数练习

编写一个函数，该函数能够统计出形式参数（数组）中所有元素的和，及最大元素的序

然后编写主函数测试，测试数组的元素值由键盘输入。

参考： double GetSumAndMaxIndex( double a[], int length, int\* index);

//由于每一个函数只能有一个返回值，题目要求求和，所以和作为返回值，而另外一个//要求要取得最大元素的序号，显然不能再返回该该序号了。如果同一函数需要返回两//个值，有以下办法： 1.用指针作为参数 如以上参考， 2.返回结构值

//参考的函数使用方法： 在主函数中定义数组 double arr[5], int indexMax;

//调用语句GetSumAndMaxIndex(arr, 5, &indexMax);

2.做一个某小型旅馆的预定系统，总共有4间房间  
房1，1张床  
房2,   1张床  
房3， 2张床，阳台  
房4， 2张床，阳台

系统要达到的功能菜单如下:输入  
0，退出  
 1，建一个新的顾客信息  
      其中包括顾客姓名和电话（结构体实现）  
 2，打出所有的顾客（顾客信息可以选择两种实现方式，结构数组、结构链表，鼓励用结构链表实现）并显示出已经有记录的顾客的姓名及电话  
 3，预定一个房间  
      其中包括预定日期  
      预定房间类型（双人间还是单人间）  
      是否包含早餐  
      是否包含车库  
 4，预订早餐  
      为哪位顾客预定早餐（输入名字）  
      预定几号的早餐（输入日期）  
 5，显示所有已经预定的信息  
      预定日期  
      预定金额  
      （单人间是１49元  
         双人间是169欧  
          早餐是19.9元  
         车库是15元）  
      房间号及是否包含早餐，车库