1. 文本操作包括哪些步骤
2. 获得文本句柄
3. 设置字体、字符大小、字符颜色等有关属性
4. 将这些属性选入设备环境
5. 什么是物理字体、什么是逻辑字体？

物理字体是为特殊设备设计的，因而是设备相关的。

逻辑字体定义的字符集是设备无关的，它可以精确标度，因此得到广泛应用。

1. 选择系统字体的步骤是什么？

(1) 定义字体句柄变量：

HFONT hF; //hF为字体的句柄

(2) 调函数GetStockObject获得系统字体句柄

它返回的是系统的缺省字体

hF= GetStockObject( )：

(3) 调用函数SelectObject将字体选入设备环境

SelectObject(hdc,hF)；

1. 怎么设置字体的颜色？

设置字体颜色：SetTextColor(hdc,crColor)；

1. 文本输出的过程是什么？

1 获取字体信息

2 格式化文本

2.1 确定后续文本坐标

2.2确定换行时文本坐标

3 调用函数输出文本

1. 如何获取字体信息？

输出文本之前要获取字体的信息，如字符高度等，以确定输出格式和下一行字符的位置

GetTextMetrics (hdc,&tm)；//tm为TEXTMETRIC结构

调用该函数时，系统将当前字体的信息拷贝到tm标识的TEXTMETRIC结构中

1. 如何格式化文本？

(1) 确定后续文本坐标

确定后续文本的坐标,应先获取当前的字符串的宽度, 该工作由GetTextExtentPoint32函数完成，并把它存储于一个SIZE结构中。

BOOL GetTextExtentPoint32

(

HDC hdc,

LPCTSTR lpszString,//指定的字符串

int nLength, //字符串中的字符数

LPSIZE lpSize //返回字符串宽度及高度的SIZE数据结构的地址

)

(2)确定换行时文本坐标

通过计算当前行文本字符的高度与行间隔之和，即可得到换行时文本的起始坐标

字符的高度与行间隔均存储在tm指向的TEXTMETRIC结构中，换行时Y轴上文本的起始坐标cy为：

cy=tm.tmHeight+tm.tmExternalLeading;

1. 如何文本输出？

常用的文本输出函数TextOut原型如下：

BOOL TextOut

(

HDC hdc,

int X, int Y, //X, Y为用户区中字符串的起始坐标

LPCTSTR lpstring, //lpstring为显示的字符串

int nCount //nCount为字符串中的字节数

)；