2001

JX18

33

档　　号

档案馆号

缩 微 号

广东交通职业技术学院

案卷题名　《C++编程技术》实验指导书

归档单位

起止日期　2001年　月　日至　2001年　3月　日

保管期限　长期

密　　级

《C++编程技术》实验指导书

1、 汉字输入速度测试

2、 自编扫雷程序

3、 图像的特技显示

广东交通职业技术学院

计算机系　孙永林 编

2001 年 3 月 10 日

卷内文件目录

案卷号	档案室编	1-2001-JX18-033
	档案馆编	

顺序号	文号	责任者	题名	日期	页号	备注
1		孙永林编	《C++编程技术》实验指导书	01.3.10	1/14	长期

装订线

《C++编程技术》程序设计实验一指导书

题目：《汉字输入速度测试》

要求：

1、使用单窗体设计，窗体中用两个 Memo 组件：一个用作样本框，用于样本的显示；另一个用作输入框，用于汉字的输入和编辑。

2、用 Button 组件设置功能：[取样本]、[输入开始]、[结束输入]、[初始化]等。

3、[取样本]功能：从一个文本文件 yb.txt 中取样本字样到样本框显示出来。并保护样本文字不允许作任何修改。样本字数设置为 100 个汉字，误差不超过 10%。

4、[输入开始]功能：点击后开始记时，同时可以在输入框中开始输入汉字。

5、[结束输入]功能：点击后不允许再输入，同时计算得分并显示。

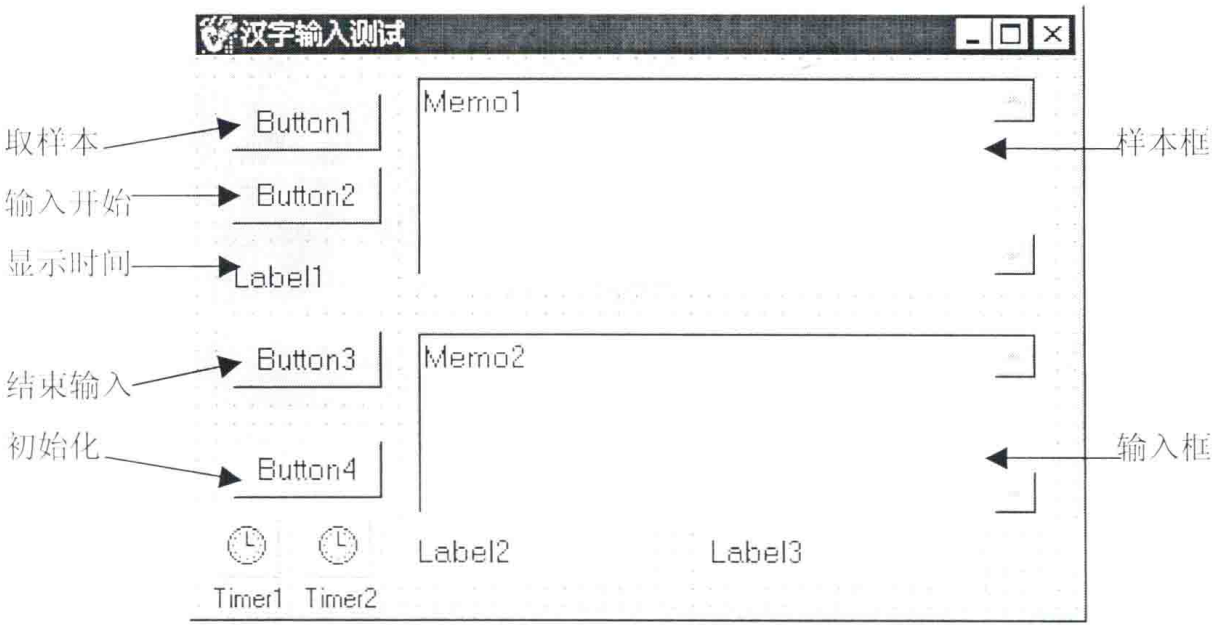
6、如果打字时间到 5 分钟，将自动停止，并计算得分及显示。

7、得分计算方法：标准分为 100 分，打字正确 1%得 1 分，5 分钟为标准完成时间，并自动计算得分。如果打字正确率超过 98%，每提前完成 10 秒加 1 分。

8、每 10 秒钟检测一次输入的正确率。

设计参考说明：

1、参考窗体设计



2、设置提示

全程变量：int ptime=5;　//设置输入时间限制在 5 分钟。

```
int pt1 到 pt6,pmin,psec;
struct time pt;  //设置一个时间结构变量，要括入 dos.h。
```

控件属性：

```
Memo1->ScrollBars=ssVerlical
Memo2->ScrollBars=ssVerlical
Timer1->Interval=1000          //1 秒
Timer2->Interval=10000         //10 秒
```

3、从 yb.txt 中取文字到 Memo 中的方法：

```
void __fastcall Twbm::Button1Click(TObject *Sender)
{
  AnsiString s1;
  int pb1=100;
  int tlen1,tlen;
  Memo1->Text="";
  try{Memo1->Lines->LoadFromFile("yb.txt");}
  catch(Exception &E)
    {ShowMessage("样本文件不存在！");return;}
  s1=Memo1->Text;
  tlen1=Memo1->GetTextLen();
 //去掉最后的回车符并计算汉字个数
  if(tlen1>0)
   while (s1[tlen1]==10||s1[tlen1]==13)   tlen1--;
  tlen=int((tlen1+1)/2);
  //判断字数是否 100
  if(tlen<pb1-pb1*0.1||tlen>pb1+pb1*0.1)
   { if(tlen<pb1-pb1*0.1)
        ShowMessage("字数少于"+IntToStr((int)(pb1-pb1*0.1))+"！");
     else ShowMessage("字数多于"+IntToStr((int)(pb1+pb1*0.1))+"！");
     Memo1->Text=""; return;
   }
  Label3->Caption="样本字数："+IntToStr(tlen);
  Memo1->ReadOnly=true;
  Button1->Enabled=false;
  Button2->Enabled=true;
  }
```

4、录入开始事件设计方法

```
void __fastcall Twbm::Button2Click(TObject *Sender)
{//录入开始
  Memo2->Text="";
  Timer1->Enabled=true;        //执行时钟事件 Timer1
```

```
    Timer2->Enabled=true;          //执行时钟事件 Timer2
    gettime(&pt);
    pt1=pt.ti_hour;
    pt2=pt.ti_min;
    pt3=pt.ti_sec;
    pt4=0; pt5=0;pt6=0;
    Button2->Enabled=false;
    Button3->Enabled=true;
    Memo2->ReadOnly=false;
    Memo2->SetFocus();
}
```

5、时间显示的设计及判断完成限制时间的设计方法

```
void __fastcall Twbm::Timer1Timer(TObject *Sender)
{   //时钟事件   ptime 为设置的时间数
    String ps1,ps2,ps3;
    int phour;
      gettime(&pt);
      pt4=pt.ti_hour;
      pt5=pt.ti_min;
      pt6=pt.ti_sec;
      phour=pt4-pt1; pmin=pt5-pt2; psec=pt6-pt3;
      if(psec<0) {pmin--; psec+=60;}
      if(pmin<0) {phour--; pmin+=60;}
      ps3=IntToStr(59-psec);
      ps2=IntToStr(ptime-pmin-1);
      //ps1=IntToStr(phour);
      Label1->Caption=ps2+":"+ps3;
      if(pmin==ptime) Button3Click(Sender);//调用按纽 1"录入结束"的事件
}
```

6、正确率的判断和显示

```
void __fastcall Twbm::Timer2Timer(TObject *Sender)
{AnsiString s1,s2;
  int tlen2,right,i;
  s1=Memo1->Text;
  s2=Memo2->Text;
  tlen2=Memo2->GetTextLen();
  /*去掉最后的回车符*/
  if(tlen2>0)
    while (s2[tlen2]==10||s2[tlen2]==13)
      tlen2--;
  right=0;
```

```
  for(i=1;i<=tlen2;i++)
    { if(s1[i]==s2[i]) right++;
    }
  right=int((right+1)/2);
  Label2->Caption="录入字数："+IntToStr(int((tlen2+1)/2))
        +"      正确字数："+IntToStr(right);
}
```

7、结束录入事件设计参考方法

```
void __fastcall Twbm::Button3Click(TObject *Sender)
{//结束录入
   AnsiString s1,s2,sdf;
   int i,right,tlen1,tlen2,tlen,vh,vm,vs,vc1,vc2,vdf;
   float vb1,vb2;
   gettime(&pt);
   Timer1->Enabled=false;
   Timer2->Enabled=false;
   if(pmin==ptime)   ShowMessage("时间到！");
   pt4=pt.ti_hour;
   pt5=pt.ti_min;
   pt6=pt.ti_sec;
   vh=pt4-pt1; vm=pt5-pt2; vs=pt6-pt3;
   if(vs<0) {vm--; vs+=60;}
   if(vm<0) {vh--; vm+=60;}
  s1=Memo1->Text;
  s2=Memo2->Text;
  tlen1=Memo1->GetTextLen();
  tlen2=Memo2->GetTextLen();
  /*去掉最后的回车符*/
  if(tlen1>0)
    while (s1[tlen1]==10||s1[tlen1]==13)
      tlen1--;
  if(tlen2>0)
    while (s2[tlen2]==10||s2[tlen2]==13)
      tlen2--;
  if(tlen1<=tlen2) tlen=tlen1;
    else tlen=tlen2;
  right=0;
  for(i=1;i<=tlen;i++) if(s1[i]==s2[i]) right++;
  right=int((right+1)/2);
  Label2->Caption="录入字数："+IntToStr(int((tlen2+1)/2))
                  +"正确字数："+IntToStr(right);
  //加分处理
```

```
  vb1=float(tlen1)/200.0;
  vb2=(float(right)/float(tlen1))*100;
  vc1=int(float(right)/vb1+0.5);
  if(vb2>=98)
    vc2=int((ptime*60-(vm*60+vs))/10+0.5); //10 秒加 1 分
    else vc2=0;
  if((vc1+vc2)<0) vdf=0;
    else vdf=vc1+vc2;
  sdf=IntToStr(vdf);
//显示得分
  Label3->Font->Color=clRed;
  Label3->Caption="录入得分： "+sdf;
  pt1=0;pt2=0;pt3=0;
  Memo2->ReadOnly=true;
  Button3->Enabled=false;
}
```

8、初始化功能的实现

```
void __fastcall Twbm::Button5Click(TObject *Sender)
{//初始化功能
   Memo1->Text="";
   Memo2->Text="";
   Button1->Enabled=true;
   Button2->Enabled=false;
   Button3->Enabled=false;
   Label1->Caption="";
   Label2->Caption="";
   Label3->Caption="";
   }
```

9、防止拷贝的技巧

设置 DragMode=dmAutomatic。

5

《C++编程技术》课程设计实验二指导书

题目：《自编扫雷程序》

实验目的：

学习利用 ImageList 控件设计程序的方法，理解 MouseDown、MouseUp 等事件的使用，并掌握引用标志数组的方法，积累一定的程序设计的经验。

设计要求：

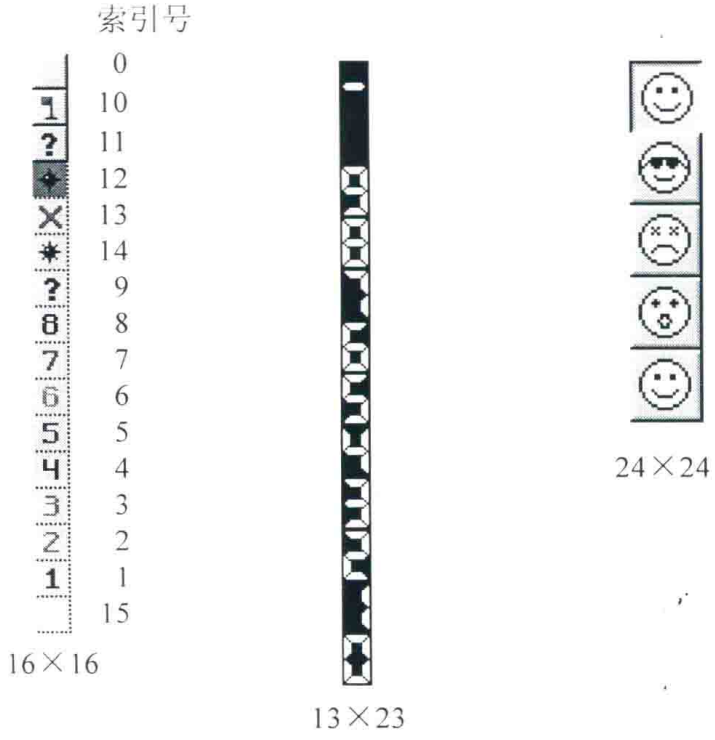
扫雷程序属于游戏类程序。是 Windows 游戏中的一种，设计要求类似于 Windows 游戏的要求。从扫雷游戏窗体来看，都会认为使用 Button 数据来实现各布雷点，实际上用这种方法是行不通的。本实验采用一种直接在窗体上利用 ImageList 控件绘图的方法来实现。为了实验更简单一些，假设：

- 雷区为 10×10；
- 布雷数为 12 个。

设计方法：

1、取位图

利用 ExeScope 工具（可以在华军软件园中下载），打开 Windows 中的扫雷程序（文件名为\windows\winmine.exe），导出其中资源文件中的位图，另存在硬盘上，以备程序使用。取出的图标图形如下：



ExeScope 工具使用方法如下：

• 执行 ExeScope.exe；

• 选择菜单命令 File/Open；接着在“打开”窗口中选择 C:\windows\winmine.exe，并点[打开]；

• 点击 Resourer 行的[+]，再点 Bitmap 行的[+]；

• 选择出所需图标图形，接着执行菜单命令 File/Export 存入硬盘文件中。

2、准备 ImageList 控件

建一工程 Bombp，添加三个 ImageList 控件（本实验只用一个，另两个作为程序改进使用），按照图标长宽设好 ImageList 控件的 Height 和 Width 属性。即：

ImageList1->Height=16; ImageList1->Width=16;

ImageList2->Height=23; ImageList2->Width=13;

ImageList3->Height=24; ImageList3->Width=24;

然后双击 ImageList1，点 Add 按纽找到要加入的图标图形文件，打开后出现提示选择 Yes。调整好各图标的索引号位置，然后选择 Transparent Color 为 cl3DLight，点 OK 完成。在程序改进时，对 ImageList2 和 ImageList3 用同样方法加入图标。

3、ImageList 的 Draw 方法

语法：ImageList->Draw(Canvas,x,y,Index,1 或 0);

x,y 是对应窗体的坐标位置；

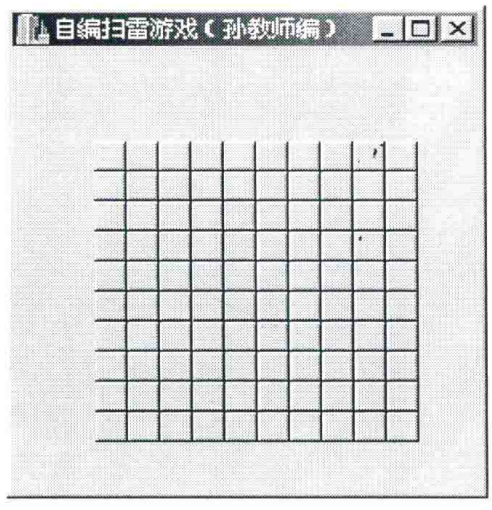
Index 是 ImageList 中的图像索引号。

例如要在窗体中（40，50）位置绘出 ImageList1 的第 3 个图像，语句如下：

ImageList1->Draw(Canvas,40,50,3,1);

4、绘出雷区

雷区图如下：



初始化雷区图

在窗体的 OnPaint 事件写入以下代码：

```
void __fastcall TForm1::FormPaint(TObject *Sender)
{ int i,j;
 for(i=0;i<=9;i++)
  for(j=0;j<=9;j++)
    ImageList1->Draw(Form1->Canvas,40+i*16,50+j*16,0,1);
 }
```

5、定义标志数组并初始化

为了记录各布雷点的状态，定义如下全局变量：

```
int cx,cy;     //鼠标点击返回的当前位置
int bflag[10][10]     //雷的标志数组。-1 表示为雷；0 到 8 表示四周的雷数。
int cflag[10][10]     //当前状态数据。0 表示为原状；1 表示已标志为雷（插上了一个红旗）；
```

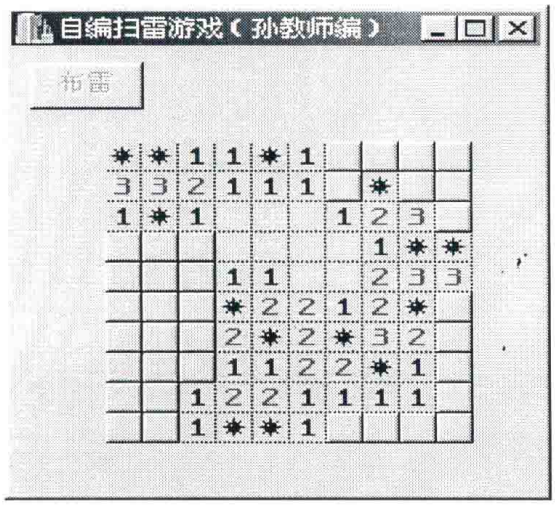
2 表示标上了问号；3 表示该位置已被点开。

两个数组 bflag 和 cflag 要初始化，方法是在 FormCreate 事件中对两个数组初值赋 0。代码如下：

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{ int i,j;
 for(i=0;i<=9;i++)
  for(j=0;j<=9;j++)
   { bflag[i][j]=0; cflag[i][j]=0;}
 }
```

6、布雷和设非雷点的雷数

布雷方法是随机产生 12 个雷区点（x,y），并置 bflag[x][y]=-1 来表示为已布雷。接着根据雷的位置设置四周非雷点的雷数。方法是对每一个布雷点四周的非雷点的 bflag[][]都加 1。如下图所示：



添加一个 Button，Caption="布雷"。其 OnClick 事件如下：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{ int i,j,b=0,bn=12;            //bn=12 为雷数
   randomize();
   while(b<bn)
   { i=random(10); j=random(10);
    if(bflag[i][j]!=-1)
      { bflag[i][j]=-1; b++; }
    else b--;
   }      //产生了 12 个布雷点
  for(i=0;i<=9;i++)
   for(j=0;j<=9;j++)
   { if(bflag[i][j]==-1)
    { if(i-1>=0&&j-1>=0&&bflag[i-1][j-1]!=-1)
        bflag[i-1][j-1]++;
       if(i-1>=0&&bflag[i-1][j]!=-1)
        bflag[i-1][j]++;
       if(i-1>=0&&j+1<=9&&bflag[i-1][j+1]!=-1)
        bflag[i-1][j+1]++;
       if(j-1>=0&&bflag[i][j-1]!=-1)
        bflag[i][j-1]++;
       if(j+1<=9&&bflag[i][j+1]!=-1)
        bflag[i][j+1]++;
       if(i+1<=9&&j-1>=0&&bflag[i+1][j-1]!=-1)
        bflag[i+1][j-1]++;
       if(i+1<=9&&bflag[i+1][j]!=-1)
        bflag[i+1][j]++;
       if(i+1<=9&&j+1<=9&&bflag[i+1][j+1]!=-1)
        bflag[i+1][j+1]++;
    }      //计算出各非雷点的雷数
  }
  Button1->Enabled=false;
}
```

7、左键 MouseDown 事件

当布雷完成以后，用鼠标点击各布雷点，有如下情况：

• 点到非雷点：bflag[i][j]==0　重画该点为□图标；

==1　重画该点为 1 图标；

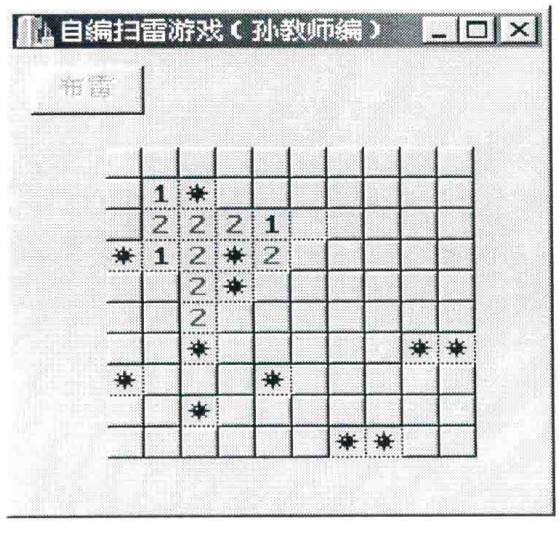
：

==8　重画该点为 8 图标；

并同时设置 cflag[i][j]=3 表示已被点开。程序代码如下：

```
if(bflag[x][y]==0)
   {ImageList1->Draw(Form1->Canvas,40+x*16,50+y*16,15,1);
    cflag[x][y]=3;
   }
  if(bflag[x][y]==1)
   {ImageList1->Draw(Form1->Canvas,40+x*16,50+y*16,1,1);
    cflag[x][y]=3;
   }
  if(bflag[x][y]==2)
   {ImageList1->Draw(Form1->Canvas,40+x*16,50+y*16,2,1);
    cflag[x][y]=3;
   }
  if(bflag[x][y]==3)
   {ImageList1->Draw(Form1->Canvas,40+x*16,50+y*16,3,1);
    cflag[x][y]=3;
   }
```

• 点中雷（bflag[i][j]==-1）：游戏结束，显示出所有的雷。如下图：



算法思路：扫描所有布雷点，如果 bflag[i][j]==-1，用 ImageList 的 Draw 方法重画这些点为雷图标。程序代码如下：

```
if(bflag[x][y]==-1)
 {for(i=0;i<=9;i++)
     for(j=0;j<=9;j++)
      if(bflag[i][j]==-1)
        ImageList1->Draw(Form1->Canvas,40+i*16,50+j*16,13,1);
   return;
 }
```

8、右键 MouseDown 事件

• 点中 cflag[i][j]==0 位置：则该位置被标志为雷，即 cflag[i][j]=1，然后用 ImageList 的 Draw 方法重画该位置。

• 点中 cflag[i][j]==1 位置：则该位置被标志为问号，即 cflag[i][j]=2，然后用 ImageList 的 Draw 方法重画该位置。

• 点中 cflag[i][j]==2 位置：则该位置被标志为原始状态，即 cflag[i][j]=0，然后用 ImageList 的 Draw 方法重画该位置。

• 点中 cflag[i][j]==3 位置：则表明该位置已被点开。不作任何处理。

这部分的程序代码由同学们自己考虑。

当把本实验完成到这里，我们会发现还有很多的不足之处。这些不足之处留给同学们去改进。改进的标准可根据 Windows 中的扫雷游戏。

《C++编程技术》课程设计实验三指导书

题目：《图像的持技显示》

目的： 掌握 C++Builder 对图像的特殊处理技术。

设计要求：

本实验提供两个特技显示方式，要求同学们根据这两种方式的特技算法，自己增加一到二个新的持技显示方式。

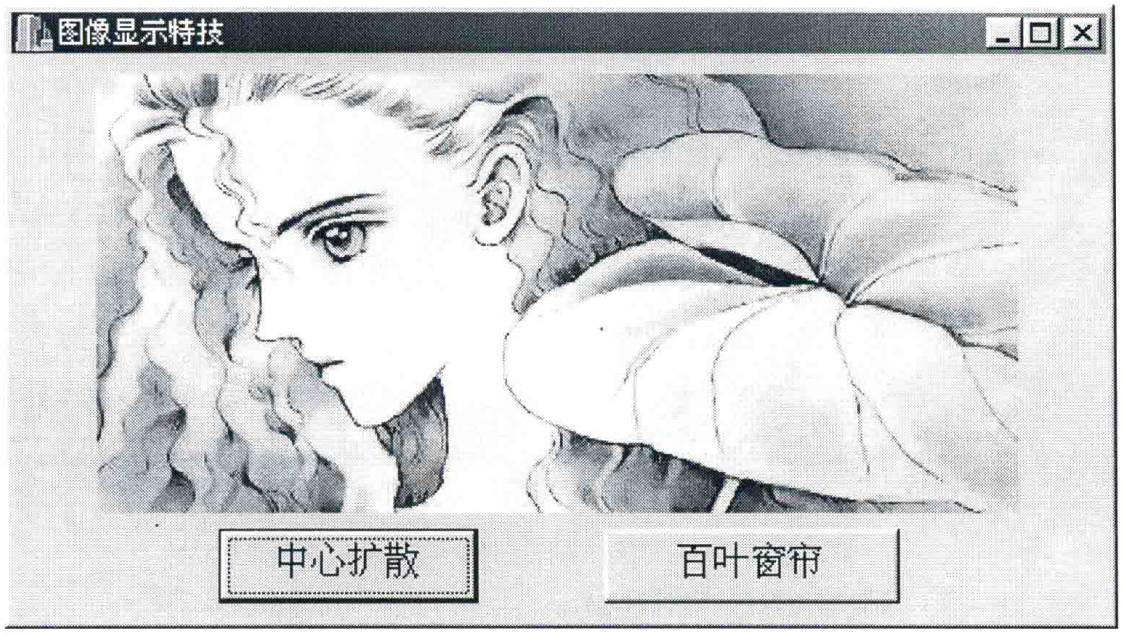
设计内容：

在网页、软件和游戏中，我们经常可以看到图像的各种特技显示，如中心扩散、百叶窗帘、右下推出等，这种动感图像显示，往往给人一种赏心悦目的感觉，给我们的应用程序增加了一些美感。在 C++ Builder 中，我们可以很容易地实现这些功能。下面实验中，将介绍实现中心扩散和百叶窗帘两种显示技巧，并说明在 C++ Builder 中处理图像的基本方法。

在 C++ Builder 中新建一个工程，然后在窗体上添加一个 Image 图像控件，把它的 Picture 属性设置为一幅 bmp 图像，调节 Image 控件尺寸与图像的尺寸大小相同。再在窗体上添加两个 Button 控件，把 Button1 的 Caption 属性设置为“中心扩散”、Button2 的 Caption 属性设置为“百叶窗帘”。

1、中心扩散

对于“中心扩散”的实现，我们利用一定的算法，通过一定的循环次数，每次显示图像的一部分，从图像的中心位置开始显示，直到显示出图像的整体。如下图：



在 Button1 的 OnClick 事件中加入以下代码：

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
 int i,left,top,width,height;
 left = Image1->Left;
 top = Image1->Top;
 width = Image1->Width;
 height = Image1->Height;
 for (i = 0; i <= width; i++)
 { //通过图像的坐标选定图像的一部分
    Image1->Left = left + (width - i)/2;
    Image1->Top = top + height/2 - i*height/width/2;
    Image1->Width = i;
    Image1->Height = i*height/width;
    Image1->Refresh();
  }
}
```

2、百叶窗帘

对于“百叶窗帘”，我们可利用画布（Canvas）提供的矩形拷贝（CopyRect）方法在不同画布之间进行图像复制来实现，该方法声明如下：

void _fastcall CopyRect(const Windows::TRect &&Dest, TCanvas *Canvas, const Windows::TRect &&Source);

把参数 Canvas 指定的源画布矩形区域 Source 复制到目标画布 Dest 的矩形区域。利用这种方法，再通过一定的算法，即可实现“百叶窗帘”的特技显示。

在 Button2 的 OnClick 事件中加入以下代码：

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
 int inum,icount,i,j;
 Graphics::TBitmap *pBitmap = new Graphics::TBitmap();
```

```
pBitmap->Height = Image1->Height;
pBitmap->Width = Image1->Width;
inum =16; //这是百叶窗的叶数
icount = pBitmap->Height/inum;
for (i = 1; i < icount; i++)
  for (j = 0; j <= inum; j++){
    pBitmap->Canvas->CopyRect(Rect(0,icount*j + i - 1,pBitmap->Width,icount*j+i),
          Image1- >Canvas, Rect(0,icount*j + i - 1,pBitmap->Width, icount*j + i));
    Form1->Canvas->Draw(Image1->Left,Image1->Top,pBitmap);
  }
 delete pBitmap;
}
```

Image 图像控件还提供了另外一些很有用的属性和方法，同学们可以充分利用 C++ Builder 的帮助，掌握这些属性和方法，再利用一些程序算法，就可以随心所欲地写出各种图像特技显示的程序。

在本实验中，同学们可以根据自己掌握的技术，增加更多的特技显示方式。