

基于 Delphi 的地图经纬网程序实现

Realization of Grid Lines of Longitude and Latitude Based on Delphi Language

朱瑜馨 张锦宗 (山东聊城大学环境与规划学院 252059)

摘要:20 世纪 80 年代后,随着数字地图与地理信息系统(GIS)技术的发展与应用,侧重于空间信息表达的电子地图应运而生。本文以电子地图中最基本的功能——绘制经纬网、图形放大、缩小等说明 Delphi 中面向对象模式程序的优势及 TCanvas 对象类的强大功能。

关键词:面向对象 Delphi TCanvas

Delphi 是国际知名公司 Borland 的优秀产品,它以面向对象 Pascal 语言为基础,提供了各种应用系统开发的面向对象环境。它采用了完整的面向对象程序语言(Object-Oriented Language)和弹性可重复利用可视化构件库(Visual Component Library,即 VCL),融合了 Microsoft Windows 图形用户界面的先进特性和设计思想。

1 面向对象的开发环境^{[1][2]}

面向对象是 Delphi 的最大特点之一,采用面向对象模式建立起来的程序是由许多不同类型的对象组成的,具有相同特性的对象叫做类(class),其中的每一个对象都是这个类的一个实例。一般意义上,类是对具有相似特征和行为的事物的一种抽象,而对象则是一个具体的事物。从平面设计的角度讲,类是一种数据类型,而对象是该数据类型的一个实现。类定义了一种由域、属性和方法组成的数据结构。如果要定义一个对象的属性和方法,则首先要定义一个具有这些属性和方法的类,然后再生成一个这种类型的实例。对象中的方法决定了要向哪个对象发送消息、发送什么消息以及收到消息后如何处理,即 OOP 模式是以对象为中心的。

在 Delphi 中过程(函数)的定义和调用必须遵循“先声明、后定义、再调用”的原则。

类的声明:

```
type
    类名 = class(父类名)
        类成员
    end;
```

2 Delphi 中的 TCanvas 绘图^{[1][2]}

Delphi 中可以通过定义一系列的对象类,如 TGeoDrv、TGeoObj、TGeoFile……等来实现笛卡儿坐标与屏幕坐标的转换,图形放大、缩小、平移和开窗;读写两种类型的地图

数据文件 *.ban 和 *.xys 文件;按图层控制地图要素,并在每一图层数据中实现图例符号的显示以及不同投影类型的转换;用 TClientDataSet 控件实现了属性数据表的操作功能:打开属性数据表,为关键字段建立索引,检索其他数据。

同样,Delphi 中提供了绘图对象——TCanvas,即画布对象。TCanvas 对象提供了一系列的属性、方法和事件来帮助创建图像,画布对象的属性有五个:Pen(画笔,用于画线)、PenPos(画笔位置)、Brush(笔刷,用于填充)、Font(字体,用于写文字)和 Pixels(像素数组,用于表示图形)。画布对象画提供的一些成员常用方法(Method)有:Draw 方法、Arc 方法、Ellipse 方法、FillRect 方法、MoveTo 方法、LineTo 方法、FrameRect 方法、PolyGon 方法、PolyLine 方法、Rectangle 方法、RoundRect 方法等,在 Delphi 中绘图,实际上就是通过操作画布对象的属性和方法来完成。

3 在 Delphi 中绘制经纬网^[2]

3.1 类的声明

本文经纬网的绘制是通过定义 Tgeodrv(设备驱动对象类)来完成的,并能实现笛卡儿坐标与屏幕坐标的转换、图形的放大、缩小、平移等功能。

//定义设备驱动对象类

```
Type
    TGeoDrv = class(TObject)
    Private
        SScale: single;
        ScreenLeft, SceenTop, SceenRight, SceenBottom: integr,
    Public
        xWorldMin, yWorldMin, xWorldMax, yWorldMax:
        single;
```

```

Canvas; Tcanvas
Constructor Create(Acanvas: Tcanvas)
Procedure Resize (ll, tt, ww, hh: inther); virtual;
Procedure Adjustscale;
Procedure XYWorld2 Screen (x, y: single; var
xx, yy: inther); virtual;
Procedure XYScreen2 World(x1, y1: Intger; var
xx, yy: single); virtual;
Procedure ZoomIn;
Procedure ZoomOut;
Procedure Pan(x1, y1, x2, y2: Intger);
End;

```

为了完成经线和纬线的绘制,在 Tform 类中声明以下两个过程:

```

procedure drawMap;
peocedure drawGrid;

```

3.2 过程的定义

在 Delphi 中的 implementation 区段进行过程的定义。以 constructor create(Acanvas: Tcanvas) 过程为例,其他程序代码略。

```

Constructor TGeoDrv. create ( Acanvas: Tcanvas);

```

```

Begin
  Inherited Create;
  Canvas := Acanvas;
  XWorldMin := -180;
  YWorldMin := -90;
  XWorldMax := 180;
  YWorldMax := 90;
  ScreenLeft := 0;
  SxreenTop := 0;
  ScreenRight := 680;
  ScreenBottom := 500;

```

```
End;
```

3.3 过程的调用

在所有的过程和函数的定义完成之后,就可以在各事件中调用这些过程和函数了。以例进行说明。

```

在程序的全局变量处定义 TGeoDrv 类的变量:GDrv
var GDrv: TGeoDrv;

```

在 FormPaint 事件、DrawMap 事件和 DrawGrid 事件中完成经纬网的绘制。

```

Procedure TformMain. FormPaint ( Sender: TOb-

```

```

ject);

```

```

Begin
  If assigned(GDrv) then DrawMap;
End;
//绘图绘制接口
procedure TformMain.drawMap ;
begin
  GDrv.clear ;
  drawGrid;
end;
//绘制间隔为 10 度的经纬网
procedure TformMain.drawGrid ;
var ll, jj: integer;
  la, lo: single;
begin
  paintbox1.Canvas.Pen.Color := $00804000;
  paintbox1.Canvas.Pen.Width := 0;
  //绘制经线
  lo := -180;
  for ll := -18 to 18 do begin
    la := -90;
    gdrv.pmoveto(lo, la);
    for jj := -9 to 9 do begin
      gdrv.plineto(lo, la);
      la := la + 10;
    end;
    lo := lo + 10;
  end;
  //绘制纬线
  la := -90;
  for ll := -9 to 9 do begin
    lo := -180;
    gdrv.pmoveto(lo, la);
    for jj := -18 to 18 do begin
      gdrv.plineto(lo, la);
      lo := lo + 10;
    end;
    la := la + 10;
  end;
  //绘制 0 度经线、赤道
  paintbox1.Canvas.Pen.Color := clred;
  gdrv.pmoveto(0, -90);
  gdrv.plineto(0, 90);

```

(下转第 74 页)

(上接第 70 页)

```
gdrv.pmoveto(-180,0);
gdrv.plineto(180,0);
//绘制回归线
paintbox1.Canvas.Pen.Color := clyellow;
gdrv.pmoveto(-180,23.5);
gdrv.plineto(180,23.5);
gdrv.pmoveto(-180,-23.5);
gdrv.plineto(180,-23.5);
//绘制极圈
paintbox1.Canvas.Pen.Color := clgreen;
gdrv.pmoveto(-180,-66.5);
gdrv.plineto(180,-66.5);
gdrv.pmoveto(-180,66.5);
gdrv.plineto(180,66.5);
end;
总之,Delphi 的 Tcanvas 对象提供了强大的绘图功
```

能,通过定义一系列的对象类,如 TGeoDrv、TgeoDbj、TgeoFile……等来实现笛卡儿坐标与屏幕坐标的转换,图形放大、缩小、平移和开窗;读写两种类型的地图数据文件 *.ban 和 *.xys 文件;按图层控制地图要素,并在每一图层数据中实现图例符号的显示以及不同投影类型的转换;用 TClientDataSet 控件实现了属性数据表的操作功能:打开属性数据表,为关键字段建立索引,检索其他数据等,从而编写出自己的电子地图程序。

参考文献

- 1 龙湘明、魏强等,Delphi 入门与提高[M],人民邮电出版社,2000,5。
- 2 徐宝林,Delphi 6.0 程序设计与应用[M],机械工业出版社,2003,6。
- 3 孙以义,计算机地图制图[M],科学出版社,2000,8。