

2009 年度第二次全国统一考试

二级开发方向答案仅供参考

一、单项选择题。

1—5、BDDAB

6—10、ADCCA

11—15、DBDCA

16—20、DDBAA

21—25、DDCAD

26—30、ADBAB

31—35、CBDBD

36—40、DCBAD

41—45、CCCBD

46—50、DCBBA

二、多项选择题。

51、ABC

52、ABCD

53、BCD

54、ABC

55、ABC

56、A

57、ABC

58、BCD

59、ABD

60、ABC

三、名词解释。

61、COM 组件：COM 是 Component Object Model 的缩写，即组件对象模型。组件实际上是一些小的二进制可执行程序，它们可以给应用程序，操作系统以及其他组件提供服务。

62、XML：XML（Extensible Markup Language）即可扩展标记语言，它是描述网络上的数据内容和结构的标准。

63、投影变换：投影变换是研究从一种地图投影点的坐标变换为另一种地图投影点的坐标的理论和方法。

64、质量成本：质量成本又称质量费用。根据ISO9000 系列国际标准质量成本的定义是：将产品质量保持在规定的质量水平上所需的有关费用。根据国际标准（ISO）的规定，质量成本是由两部分构成，即运行质量成本（或工作质量成本，或内部质量成本）和外部质量保证成本。运行质量成本是指企业为保证和提高产品质量而支付的一切费用以及因质量故障所造成的损失费用之和。外部质量保证成本是指为用户提供所要求的客观证据所支付的费用。

65、接口提供不同对象之间的连接，它包含了一组函数的数据结构，通过这些数据结构，客户代码可以调用组件的功能。

四、简答题目。

66、栅格结构和矢量结构是地理信息系统的两种空间数据结构。栅格数据结构是以规则的阵列来表示空间地物或现象分布的数据组织。它将地球表面划分为大小均匀紧密相邻的网格阵列，每个网格作为一个象元或像素，由行、列号定义，并包含一个代码，表示该像素的属性类型或量值。矢量数据结构是通过记录空间对象的坐标及空间关系表达空间对象的

几何位置。

栅格数据结构类型具有“属性明显、位置隐含”的特点，它易于实现，且操作简单，有利于基于栅格的空间信息模型的分析，如在给定区域内计算多边形面积、线密度，栅格结构可以很快算得结果，而采用矢量数据结构则麻烦的多；但栅格数据表达精度不高，数据存储量大，工作效率较低。如要提高一倍的表达精度（栅格单元减小一半），数据量就需增加三倍，同时也增加了数据的冗余。

| 矢量数据 | 栅格数据 |
|----------------------|--------------------|
| 1. 数据存储量小 | 1. 数据存储量大 |
| 2. 空间位置精度高 | 2. 空间位置精度低 |
| 3. 容易实行拓扑和网络分析 | 3. 不易实现拓扑和网络分析 |
| 4. 输出简单容易，绘图细腻、精确、美观 | 4. 输出速度快，但绘图粗糙、不美观 |
| 5. 可对图形及属性进行检索、更新和综合 | 5. 便于面状数据处理 |
| 6. 数据结构复杂，获取数据慢 | 6. 能够快速获取大量数据 |
| 7. 图形运算复杂、高效 | 7. 图形运算简单、低效 |
| 8. 多种地图叠合分析困难 | 8. 多种地图叠合分析方便 |
| 9. 不能直接处理数字图像信息 | 9. 能直接处理数字图像信息 |
| 10. 空间分析不容易实现 | 10. 易于进行空间分析 |
| 11. 与遥感影像格式不一致 | 11. 与遥感影像格式一致或接近 |
| 12. 数据输出的费用较高 | 12. 技术开发费用低 |

67、地理对象的拓扑关系，主要有以下三种：

- 相邻：是指对象之间是否在某一边界重合，例如行政区划图中的省、县数据。
- 重合：是指确认对象之间是否在某一部互相覆盖，如巴士线路和道路之间的关系。
- 连通：连通关系可以确认通达度、获得路径等。

68、(1) create TABLE 空间数据

```
(Id          Int Primary Key,  
  X          float,  
  Y          float,  
  Z          float  
)
```

create TABLE 属性数据

```
(Id          Int Primary Key,  
  OID        Int,
```

```

    名称      Varchar(50),
    长度      Varchar(50)
)

```

```

create TABLE 图形信息

```

```

(Id          Int Primary Key,
OID          Int,
颜色        Varchar(50),
类型 ID     int
)

```

```

create TABLE 图形类型

```

```

(类型 ID     Int Primary Key,
类型名称 Varchar(50)
)

```

```

(2) select 属性数据. 名称
from 属性数据
where exists
(select * from 图形类型, 图形信息 where 图形类型. 类型名称='线' and 属性数据. 长度
>=100
and 属性数据. ID=图形信息. ID and 图形信息. 类型 ID=图形类型. 类型 ID)

```

```

(3) select X,Y,Z from 空间数据
where exists
(select * from 图形类型, 图形信息, 属性数据
where 空间数据. ID=属性数据. ID and 属性数据. ID=图形信息. ID
and 图形信息. 类型 ID=图形类型. 类型 ID and 图形信息. 颜色='红色' and 属性数据. 名称
='火车站')

```

69、略。

70、方法 1：中点算法画圆：假定 x 坐标为 x_p 的像素中与该圆弧最近者已确定，为 $P(x_p, y_p)$ ，则下一个像素只能是正右方的 $P_1(x_p+1, y_p)$ 或右下方的 $P_2(x_p+1, y_p-1)$ ，构造函数：
 $F(x, y) = x^2 + y^2 - R^2$ 。设 M 是 P_1 和 P_2 的中点，即 $M = (x_p+1, y_p-0.5)$ 。则 $F(x) < 0$ 时， P_1 离圆

弧更近，取 P1 为下一像素， $F(x) > 0$ 时，P2 离圆弧更近，取 P2 为下一像素， $F(x) = 0$ 时，约定取 P2 为下一像素。

方法 2: Bresenham 算法画圆：求误差初值， $p_1 = 3 - 2r$ ， $i = 1$ ，画点 $(0, r)$ 。求下一个光栅位置，其中 $x_{i+1} = x_i + 1$ ，如果 $p_i < 0$ ，则 $y_{i+1} = y_i$ ，否则 $y_{i+1} = y_i - 1$ 。画点 (x_{i+1}, y_{i+1}) ，计算下一个误差，若 $p_i < 0$ ，则 $p_{i+1} = p_i + 4(x_i - y_i) + 10$ 。 $i = i + 1$ ，若 $x = y$ ，则结束，否则重新计算 y_{i+1} 。

71、答：本大题采用 Delphi+ArcEngine 来实现。

- (1) 需要用到的接口有：Ifeature、IfeatureLayer、IfeatureClass、IfieldEdit。
- (2) 向程序中添加引用单元 esriCarto_TLB, esriGeoDatabase 库。
- (3) 编码实现，假设 MapControl 控件名称为 aeMap, 先要素文件存储在 C 盘根目录下。

```
Var pFeatureLayer: IfeatureLayer; //要素层对象

pFeatureClass:IfeatureClass;//要素类对象

pFeatureCursor :IfeatureCursor;//要素游标

pFeature: IFeature;

pNewField: Ifield; //新字段

pEditField: IfieldEdit;//编辑字段的接口

sRowValue:String;//存储每个要素属性值

I:integer;

Begin

    aeMap.clearLayers;//清空图层

    aeMap.AddshapeFile( 'C:\', '道路' ); //读取线要素文件

    pFeatureLayer: =aeMap.Layer[0]as IfeatureLayer;

    pFeatureClass: =pFeatureLayer.FeatureClass;

    pNewField: =CoField.Creat as Ifield; //实例化字段对象

    pEditField: =pNewField as IfieldEdit;//接口查询至 IfieldEdit

    pEditField.type:=esriFieldTypeString;//为新字段指定类型 String

    pEditField.Name:=' 道路级别' ; //字段名称

    pFeatureClass.AddField(pEditField);//添加新字段到要素类

    pFeatureCursor: =pFeatureLayer Search (nil, false) ;//得到所有要素的游标

    指针
```

```

pFeature:=pFeatureCursor.NextFeature;

While pFeature<>nil do

begin

    pFeature.Value[pFeature: Field.FildField(道路级别)];//存储新的值

    pFeature:=pFeatureCursor.NextFeature;//移动游标

end;

END

```

设计中主要用到的接口，方法有：MapControl、AddshapeFile

编辑字段的接口：IfieldEdit

读出属性的：IFeature.Value[I]

I 为字段索引号

72、答：（1）用 ItopologicOperator 接口为道路创建缓冲区，Buffer（方法）。

（2）为了获取道路的单侧缓冲区范围，也用 ItopologicalOperator 接口的 Cut 方法获得单侧缓冲范围。

（3）用单侧缓冲区的几何形状，Igeometry 接口，对商铺、居民区、农田作基于空间位置的地理查询，先获得都有哪些地物与单侧缓冲区相交、分别得到各自的要素游标 IfeatureCursor，然后遍历游标，用单侧缓冲区与每个要素作空间相交关系运算，得到面积（要用到 Iaera 接口），得到三类地物各自的总面积，在与单价作积运算，求出总拆迁费用。

代码如下：

```

pTopOperator:=pFeature.Shape as ItopologicalOperator;

pBufferGeometry:=pTopOperator.Buffer(10);

pTopOperator:=pBufferGeometry as Itopological(Operator);

pTopOperator.Cut(pLine as IpolyLine,pleftBuffer,pRightBuffer);

//用 cut 方法会得到两个区域，对这两个区域都做计算，由用户决定取哪一个区域的计算结果

pSpatialQuery:=CospatialQuery.Creat as IspatialQuery;

pSpatialQuery.Geometry:=pleftBuffer;

pSpatialQuery.GeometryField:= 'shape' ;

```

```

pSpatialQuery.WhereClause:=' 地物类型=" 商铺" ' ;
pSpatialQuery.SpatialRel:=esriSpatialRelationOverLayer;
//叠置查询
pFeatureCursor:=pFeatureLayer.Search (pSpatialQuery, False);
pFeature:=pFeatureCursor.NextFeature;
pTopOperator:=pLeftBuffer as ITopologicalOperator;
while pFeature<>nil do
begin
pArea:=pTopOperator.Intersect(pFeature.shape, esriGeometry);
dArea 为对面积进行计算的双精度变量
dArea:=pArea.Area+dArea;
pFeature:=pFeatureCursor.NextFeature;
END;
d 商铺:=dArea*4000;
//得到商铺的总费用
同理：可求得其他两种地物的费用。

```