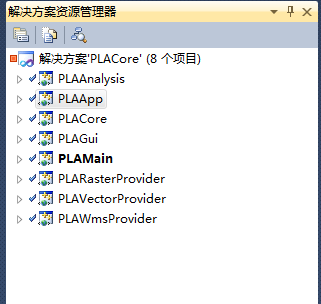
PLA基于开源GIS软件QGIS开发，采用插件式开发模式，程序代码分为两块：

1. 内核
2. 插件
3. 内核





* 1. 启动项目：PLAMain

1. 普通QT界面程序

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

qtApplication w;

w.show();

return a.exec();

}

2) PLA主程序

int main(int argc, char \*argv[])

{

QgsApplication myApp(argc, argv, myUseGuiFlag, configpath);

QgisApp \*qgis = new QgisApp( mypSplash, myRestorePlugins );

Qgis->show();

return myApp.exec();

}

应用程序类：QgsApplication myApp( argc, argv, myUseGuiFlag, configpath );

class CORE\_EXPORT QgsApplication : public QApplication

主窗体类：QgisApp \*qgis = new QgisApp( mypSplash, myRestorePlugins );

class APP\_EXPORT QgisApp : public QMainWindow, private Ui::MainWindow

* 1. PLA主窗体（一个完整的GIS系统）：PLAApp

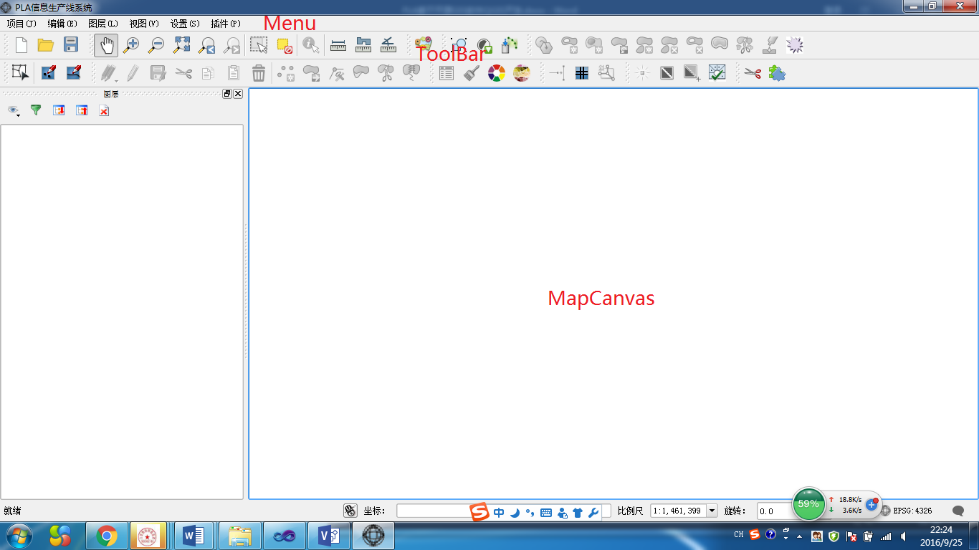
1. 创建界面（菜单、工具条、窗口）
2. 加载数据驱动

PLARasterProvider：读取栅格数据驱动

PLAVectorProvider：读取矢量数据驱动

PLAWmsProvider：加载网络服务驱动

1. 画布 mapCanvas：public QGraphicsView



1. 一个重要成员变量 QgisAppInterface mQgisInterface



QgisAppInterface是QgisApp的友元。

作用：插件访问QgisApp的接口

插件中只要操作mQgisInterface，就可以操作整个界面

1. 地图工具

地图工具都是都画布中的内容进行操作，mQgisInterface->mapCanvas()就可以得到画布，然后根据选择的MapTool,就可以对数据进行呢操作了。



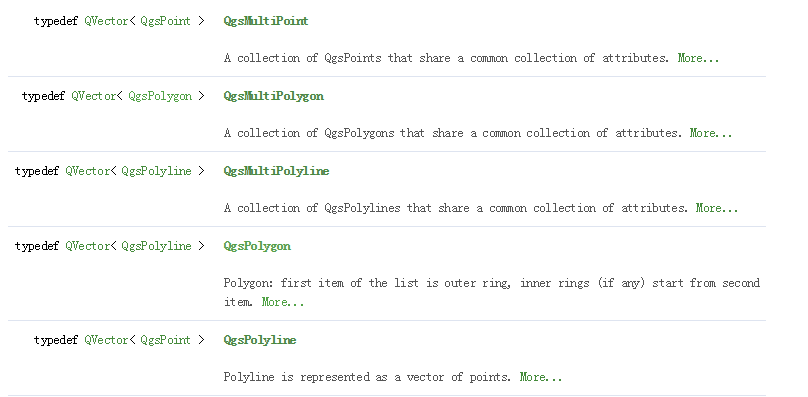
3. 数据模型

1）图层



1. 矢量图形





4. 数据精度

矢量图层中添加一个成员变量：dataPrecision，该变量的值从系统设置中读取，该值就是数据处理的精度。

5. 智能卡

跟踪器（边界法类）：鼠标拾取一个点，则该点与上一点作为样本，对矢量或影像进行实时计算，将两点间的连线（橡皮条）吸附到矢量边界上或影像地物边界上。

处理器（样本法类）：创建完一个要素，该要素的顶点作为样本，对影像进行计算，利用漫水填充类算法提取与样本相似的区域，矢量化为要素。

1. QgisApp的InitGUI（）中将卡加入列表中
2. 在鼠标助手窗口中选择智能卡
3. 根据规定的操作方法在画布中画要素（橡皮条），通过智能卡重写的Process（）或Track（）方法进行后台计算；
4. 将计算的结果作为橡皮条，写入。

问题：画样本标记和画要素没有区分开，只是截取了橡皮条点进行修改，所以都需要借助于已有的要素创建工具。 生成面，就只能借助于创建面的工具；生成线，就只能借助于生成线的工具。不能实现复杂且统一的操作逻辑

1. 插件
2. mQgisInterface
3. MapTool