一、main函数

1.main函数在myutil/mymain.h中

2.在appExec函数中：

（1）首先加载language目录下的qm翻译文件；

（2）然后加载resource目录下的rcc资源文件（资源文件由）；

（3）解析传入的默认参数，包括账号，密码，打开的模块名称等

（4）执行run函数，同时传入解析的默认参数

3.在run函数中

（1）加载plugins目录下的模组插件：使用QPluginLoader来获取插件，然后保存到变量中，后面直接通过module中配置的插件名称就可以找到插件

（2）加载源码module目录下的文件，通过一个cfg文件内容来拿到module目录。接着读取每个module.conf里的内容，然后报错到变量中，后面直接通过模块名获取module配置

（3）加载脚本扩展

（4）展示登录界面，并传递默认参数

（5）如果登录成功则隐藏登录界面，并显示主界面；否则弹出提示

（6）打开主界面：实现通过模块名（例如:my-main）获取插件，将插件转化为接口，然后调用接口的newClass函数，传入类名（用于new一个对象）和module名称（用于initModule）

二、主界面

1.初始化主界面，调用initModule函数，这里用到了myclassabs类。初始过程包括：（1）初始化module内容；（2）初始化uim；（3）初始化action配置；（4）初始化action内容；（5）初始化按钮

2.初始化界面布局

3.左侧的按钮关联槽函数，当点击时，打开新模块

4.如果左侧是menu，那么展示菜单，当点击菜单项的时候，打开新模块

5.每个模块只能打开一次，如果再次点击按钮，那么会跳转到已经打开的界面

6.主界面实现了拉伸边框功能

7.移动是通过WindowDragger类来实现的。

三、mybaseutil

1.myscriptengine继承自QScriptEngine，增加了importPackage和availablePackages函数，暂时没有找到使用到地方

四、myclassabs

1.继承自QMainWindow

2.是所有业务类的父类，也是主界面的父类。

3.initModule函数，所有业务类在构造函数中第一个调用的函数，实现了如下功能：

（1）initModuleMap函数，将模块名和module.conf的内容对应起来.主要是为了config函数能够根据key获取到内容。

（2）initUimMap函数，将uim.conf的内容保存起来。主要是initButtons时使用

（3）initActionMap函数，包括initActionConfMap和initActionFunctionMap。前者将action的配置和action的名称对应起来；后者将action的内容和action的名称对应起来。主要时为了initButtons使用。

（4）initButtons函数，从uimMap中找到TOOLBAR，然后创建toolbar和action按钮，action按钮的配置通过actionConfMap拿到（比如按钮显示文字），action中通过setData将按钮的实际内容存放起来，当点击时，调用onBtnClicked函数。

4.onBtnClicked，首先获取当前模块的指针（方法是：从按钮开始，一直找父类，直到父类等于打开的某一个界面时，那么这个类就是当前的模块）；接着或者action的脚本内容，接着调用脚本，并传入当前模块指针。

5.在获取当前模块指针的时候，用到了APP->getAllClass()，里面存放了所有打开的模块。

五、MyClassPuginAbs

1.该接口在业务插件处使用。

2.newClass通过类名创建新模块，并将模块名传进去，用于初始化

六、MyCore

1.addPlugin函数，找到插件目录，并将目录下的所有插件通过QPluginLoader加载，然后存放到pluginMap里，由插件名称对应插件类

2.addModule函数，将module.conf的内容存放起来，由module名称对应内容

3.openModuleUrl函数，通过module名称，获取到对应的内容，然后获取到plugin和class，接着调用openModule函数。

4. openModule函数，使用通过模块名获取到插件，然后插件调用newClass，传入class和模块名，即可获取到模块，将模块进行show，并返回模块指针。调用处获取到模块指针，即可继续处理。

5.每次打开一个模块后，就会将模块存储起来，每次关闭一个模块，就会将模块删除掉