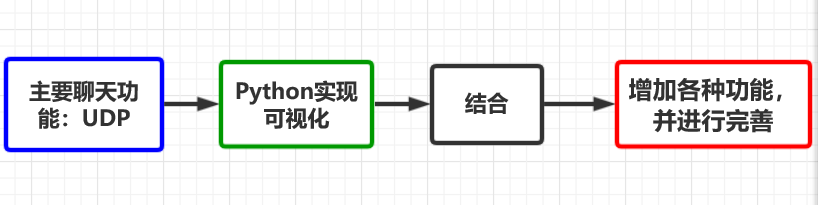
**QQ开发文档**

**一 .开发思路：**

****

根据实习指导中的提示，一步一步从小到大来不断完善程序。主要分两个步骤，一个是计算机网络领域的UDP，另一个是Python的可视化程序。

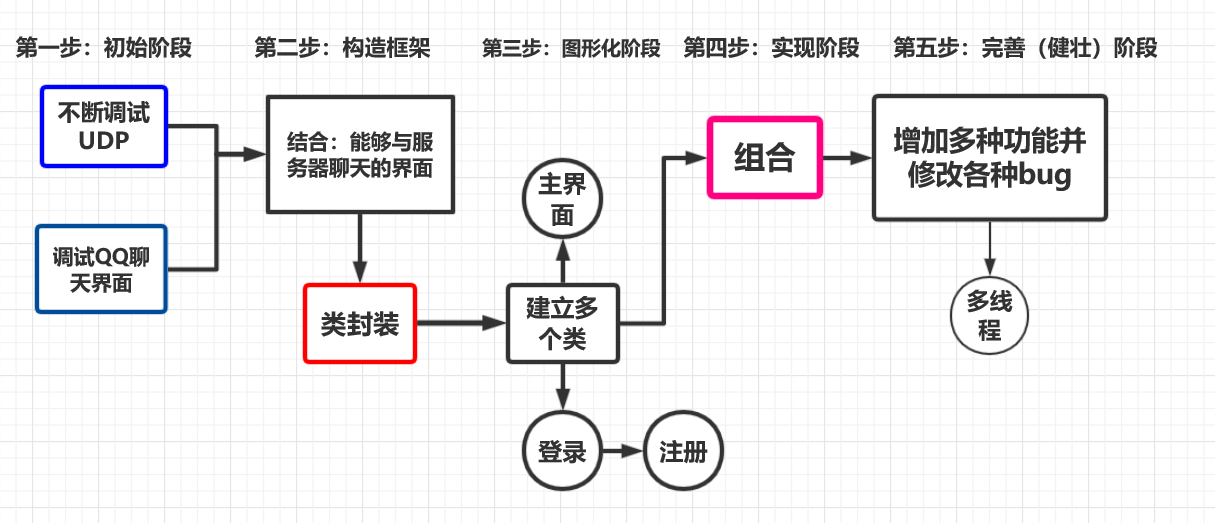
我们课上学的UDP很简单，对于本次实习来说，主要是通过与老师所给的服务器进行交互信息来实现QQ的与好友收发信息。

而对于可视化来说，需要掌握老师所给的两个文件。通过修改每行代码，了解这些代码的功能。之后再根据自己的需求，通过老师和网上查阅的代码，完成可视化的对话框程序。

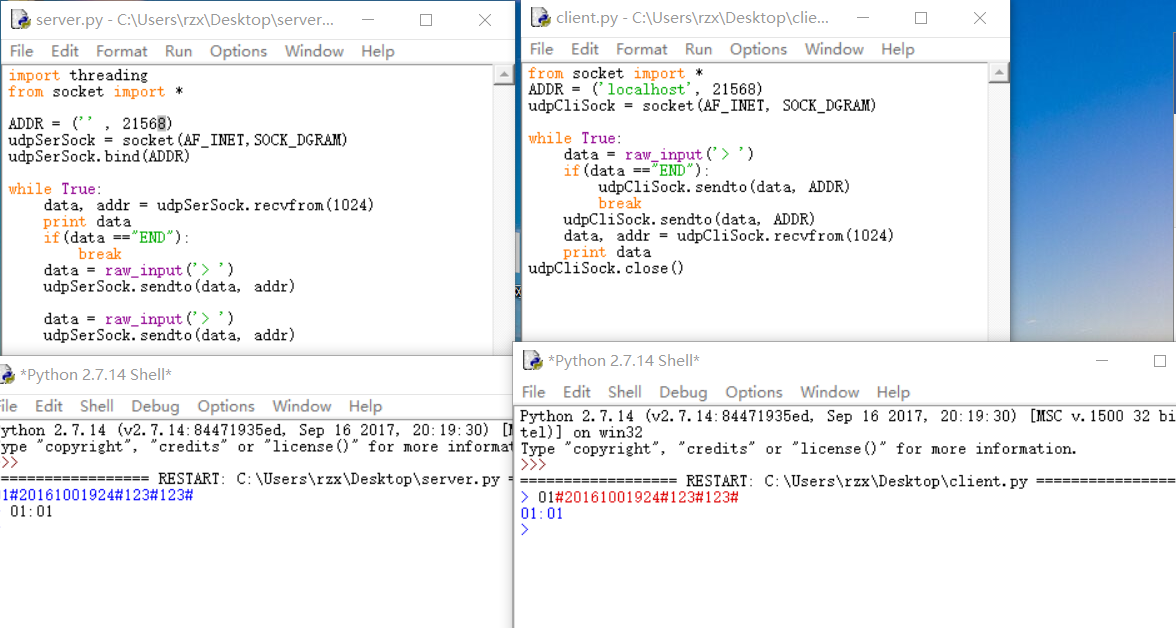
最后再将对话框程序里面加入UDP的代码。最后形成QQ程序。

之后不断与服务器测试，来微调一些BUG，错误等。

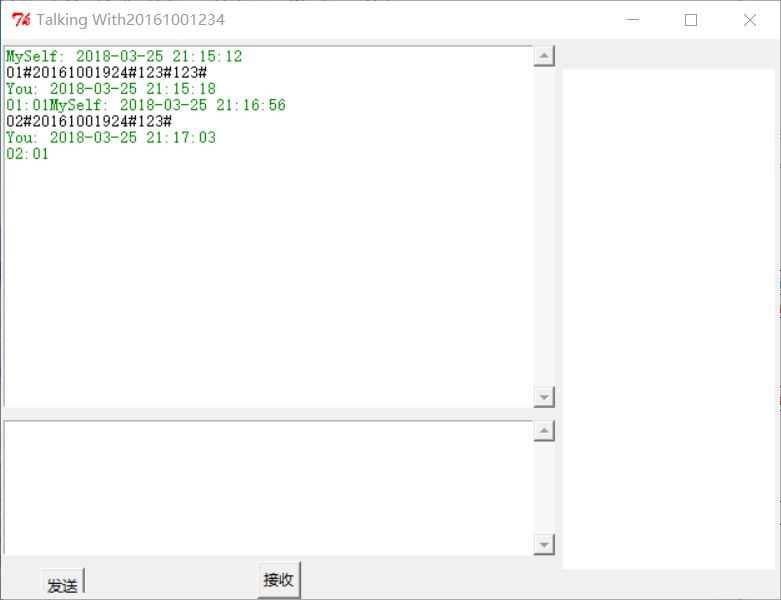
**二 .过程：**



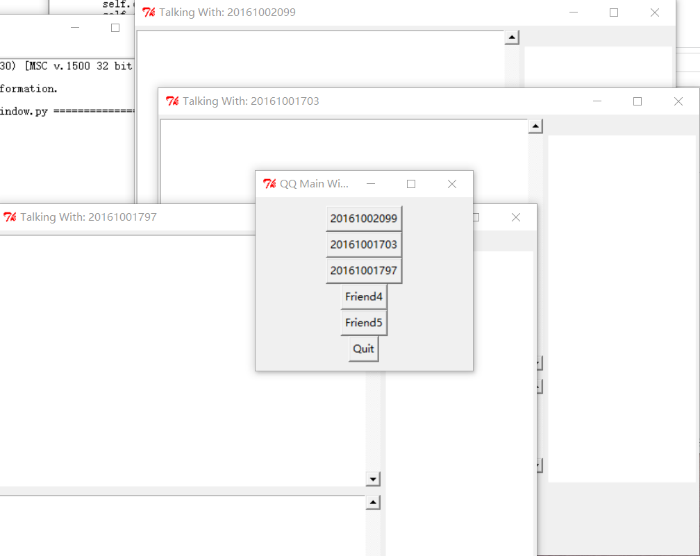
1.首先根据所给的UDP，设置成循坏接受，一发一收的 server与client。



2.之后开始在测试老师给的client 的窗口。并且将之前写的UDP与主窗口结合。从而可以实现通过主窗口进行聊天。于是可以通过所给的格式，与同学（好友）收发消息。



3.之后开始进行类封装上面的主聊天窗口。因为注意到还有一个给出的主界面，每点一个friend就会弹出一个主聊天界面。因此，需要将其进行类封装，再通过面向对象，生成多个对象，从而与不同好友聊天。

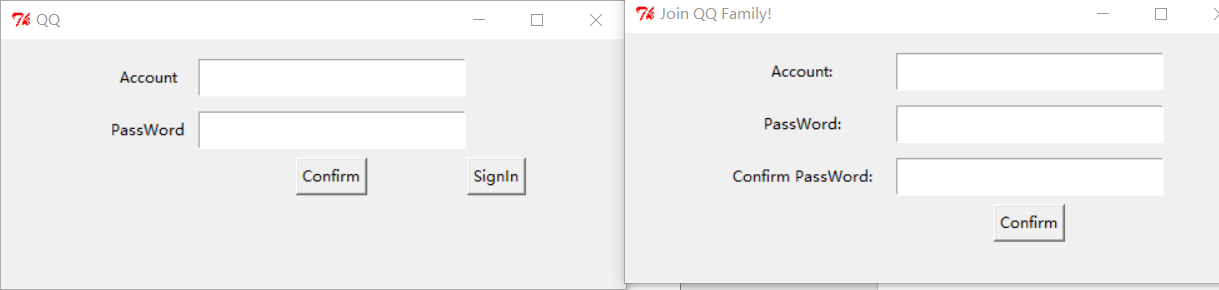


这一步需要两个类，但是根据以前的代码格式，不想将两个类写在一个文件中，于是，查阅了 import 。类似于include。

需要注意的是，如果打开对话窗口时，会弹出一个空的Tk。之后找到了解决方案。因为里面之前是用框架frame来写的，需要在框架的构造函数里传一个master。如果只有self时，可以用self.master，来调用。我理解的是，一开始root=Tk（），这是建立一个空的Tk。之后App=Cilent（root），是将Tk这个“灵魂”传给Cilent的“实体”。于是调用mainloop后就可以明确知道哪个窗口正在运行。 如果不给frame传入master，则不知道Tk控制哪个窗口。于是会和父窗口的master发生冲突。于是可以看出新生成的窗口的标题与父窗口相同。

但是当我们不用frame时，可以直接默认传入的是哪一master。

4.之后开始仿照聊天界面，写登录界面与注册界面。登录窗口上有一个按钮，可以打开注册按钮。两个类是连在一起的。



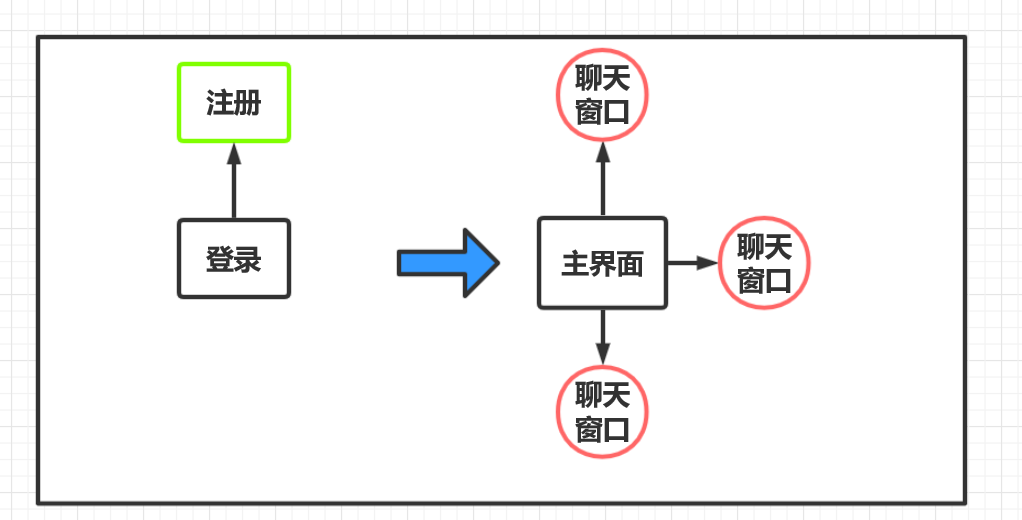
里面需要用UDP根据格式发送信息到服务器中，之后接受返回值，根据不同类型，反馈不同信息。

5.将所有界面逐层import，可以按照QQ的流程使用。但是出现了许多错误。

首先是越权。返回07:01，之后上网查阅并与同学讨论分析的出，需要只能用一个UDP来与服务器收发消息，即之前在每个类中都写了一个clisock= socket(AF\_INET, SOCK\_DGRAM)。 不然服务器不知道登陆账号的与发送信息的是不是一个端口。

其次这样逐层import的方式比较繁琐，如果关闭最一开始的对话框后，之后的子对话框（对象写在父类中）会报错。于是对其进行修改。方式类似于main主函数。将主代码写在一个文件中，里面import所有的类。这样先打开登录窗口如果登陆成功直接自动关闭这个窗口，继续运行，打开主界面,进行之后的操作。

对于如何判断是否登陆成功，想到通过向对象中传入引用的参数，如果登陆成功就将其值修改，这样删除对象后，此值还是改变了。关闭对象后，通过判断值是否改变，来判断是否登录成功。但python中所有的参数都不是引用的，而是值改变。但是如果传入一个列表，只改变list【0】后，类似于改指针的地址。这样可以实现类似于引用参数。



6. 不断测试，向服务器发送各种信息，并且修改BUG。同时加入动态调整对方账号，从而可以更加灵活的聊天。

7. 加入多线程，但是只能调用一次，通过上网查找，发现需要多次将这个函数每隔一段时间进行回调。这样可以实现多线程。即同时收发信息。

global timer

timer=threading.Timer(0.5,self.OnlineRecvMessage)

timer.start()

8.再次进行微调，不断测试。