嗅探器制作流程：

.嗅探器部分

获取网络接口设备[网卡设备]

想要捕捉网络数据包，首先要得到电脑的网络接口设备，也就是所谓的网卡设备．Jpcap包向我们提供了这一个类方法［静态方法］:JpcapCaptor.getDeviceList( )  ，从字面见就知道返回的是网络接口设备的列表，NetworkInterface类，网络接口类。其封装了相应网络设口信息，比如：网卡名，网卡描述，网卡所处的数据链路层的名称，描述等等。

打开网络接口设备 当得到网络接口设备列表后,你所要做的事就是打开一个网络接口设备.得到一个捕获者的实例,JpcapCaptor类的实例.如下的代码就是实现上述目的.

使用JpcapCaptor实例来捕捉网络数据包

实现一个方法处理器接口的类PacketReceiver，并将该类的一个对象，注册到捕获器中。（回调方法的实现, 其实就是监听器模型）

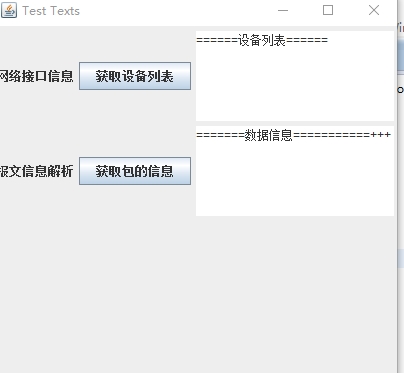
将消息处理器（PacketReceiver） 注册到捕获器（JpcapCaptor）有两种方式：

captor.processPacket(int number,PacketReceiver reciver);//1

loopPacket(int number,PacketReceiver reciver);//2

最后对抓取数据进行解析

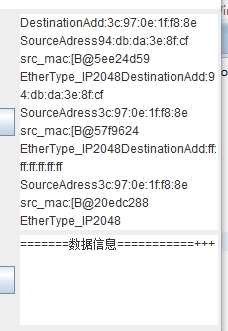
1.**初始化图形界面**



**点击获取设备列表**：得到如下部分：



**点击获取包的信息**得到如下信息，包括目的地址，源地址，包类型等 数据信息



实验总结：这个项目做下来感觉收获很大：

1. 最重要的是学会了通过收集整理资料来自主学习，因为我对java相对熟悉，（很少用c）所以选择了基于libcap和winpcap的Jpcap，但是这方面的资料在网上很少，所以只能零碎的去学习，所幸的是终于写出来了
2. 其实巩固了图形界面的编写，以前总觉得java没必要深究图形界面，但这次写代码我发现自己基本的图形界面编写方法都不太熟练了，这次虽然写的简单，但也让我理清了各组件使用方法。
3. 解决问题的能力，看上去量很大的小项目，但也是会遇到很多小问题，在不断的纠错与查阅资料中，锻炼了解决问题的能力。