NetSender文档（代码说明）

这篇文档的结构是：下面的每一点都是上一点的概括；往前是概览，往后则涉及到更多实现的细节。

1. 功能简介
   1. 多文件同时发送和接收，并且显示传输速度和进度条。
   2. 聊天功能。
   3. 还有一个歌曲同步播放器。制作这个播放器的想法在于，网络两端的两个人可以一起听歌，和对方分享自己的音乐。这两个人正在听的音乐是同步的，也就是说，一方调整播放进度、增加删除歌曲、开始/停止播放都会影响到另一方。
2. 软件结构

写这个程序时发现，虽然上面的想法简单，但实现起来却远比想象中的复杂。后来我尝试把这个软件按外壳到内核分成4层。这个想法来自于计算机处处可见的层次结构。如网络传输数据要先经过中有应用层、链路层什么的，最后才到物理层；操作系统中使用硬件先从外壳到内核，到硬件抽象层，然后才是真正的硬件。所以这次不如我也来试试分它几层。

* 最上层是GUI——NewJFrame。一个继承 JFrame的类。
* 先说第三层。是3个传输器（xxxTransmitter）。对应这个软件的3个功能。
* 第4层是两个类BaseSender,BaseReciever，为数据发送/接受提供底层实现。由于第三层的传输器有很多共性，于是我把这些共性抽象出来，做成两个抽象基类BaseSender,BaseReceiver。第三层的类都继承它们。
* 第二层是TransmitterManager，传输器管理器。这个类作为图形界面和底层实现（各种数据传输器）之间的中间层，图形界面和底层通过它来交换信息。

加入这个中间层为的是将交换信息的代码与图形界面中各种对控件的繁杂操作分离开。使程序的结构更清晰。

1. 关于结构的更多细节

* 第三层，每个传输器其实是两个类，xxxSender，xxxReceiver。
* 第二层的更多信息：TransmitterManager作为一个中介把NewJFrame和xxxTransmitter隔开。
  + NewJFrame从来不使用第三层的类，而是调用TransmitterManager里相应的函数，比如发送文件时，只需一句TransmitterManager. SendFile(String ip,int port,FileSenderListItem item)；
  + 同样，第三层也从来不直接使用NewJFrame，比如接收文件，这个事件发生在第三层，却需要修改GUI上的控件来提示接收百分比和传输速度。这时文件接收器也只需一句TransmitterManager.tellFileReceivingProgress(double percent, int speed, File f)，将修改GUI的操作交给第二层完成。
  + TransmitterManager里就有大量这样的，供第三层或第一层调用的函数，来“代理”两层间的信息交流，让第一层和第三层代码简洁和结构清晰。
* 第四层和第三层的关系：其实发送和接收数据都有个共性——就是新建线程、新建Socket对象等繁杂操作的最终目的，都是获得一个输入流，一个输出流来供数据传输。第4层的BaseSender和BaseReceiver的作用就是完成这个工作——获得这两个流，把它们传给第三层。

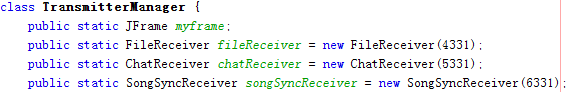
不同的传输器传输不同的数据。所以BaseSender和BaseReceiver保留了一个抽象方法Work给第三层的类，实现具体数据的传输。（下面的in和out,dataDescription都是第四层提供的）

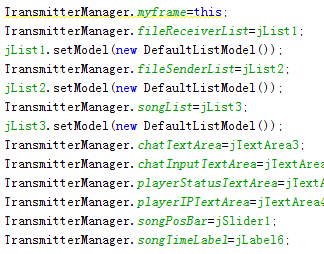
实际上，第三层的各种传输器只需实现这个Work函数即可，再没其它工作要做，比不分层的代码简洁了许多。

1. 关于结构的更多更多细节

* 各个类的用法都力求简洁：
  + xxxReceiver用法：一句new xxxReceiver (int port)即可开始工作。

它们在TransmitterManager.java里：



* + xxxSender用法：一句new xxxSender(ip,port).Send(Object dataDescription)即可。
  + TransmitterManager用法：在使用这个类时，要先告诉它各种GUI控件的引用。于是它的用法是，在NewJFrame的构造函数中设置这些变量：然后TransmitterManager可开始正常工作。