ChatDome设计文档

#### 1 程序的运行环境、安装步骤

         （1）****运行环境****：.Net Framework 4.5.2

         （2）****程序的组成部份****：



ClientApp为客户端，ServerApp为服务器端，ClientMessage.cs里是客户端的提示信息，ServerMessage.cs里是服务器端的提示信息，这些信息用来在客户在输入不规范的命令，或不当的操作（直接关闭）时给出的提示信息。MyNetWorkLibrary是一个辅助项目，用来处理socket发送接受数据，以及IPEndPoint的获取。

         （3）****安装步骤****：

1）  安装 .Net Framework 4.5.2。

          2）将程序AChatServer文件夹复制到计算机上

          3）在资源管理器中，双击server.exe，运行服务器端，双击ChatDemo.exe，运行程序客户端

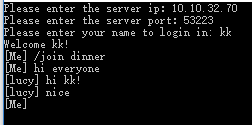
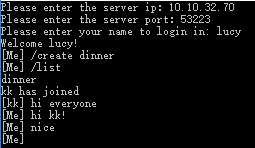
#### 2 程序开发平台

         （1）代码行数：400

         （2）开发环境：Visual studio community 2015

#### 3 程序功能说明：





主要功能参考自UCB计算机网络课程CS168的Project1：<https://github.com/NetSys/cs168_student/blob/master/projects/proj1_chat/spec.md>

如上图示例所示，首先启动server，在启动若干个client，输入server的ip和port，输入名字。

所有的输入分为2中，一种普通输入，会被广播到所有同频道的client，另一种命令以’/’开头，支持3中命令/join，/list，/create，所有命令都有格式要求，错误的格式会出现响应的错误提示信息。用户加入或退出频道会有提示信息。

#### 4 程序算法说明及面向对象实现技术方案

1. 对用到的算法的说明，简要介绍其算法步骤、时间空间复杂度等

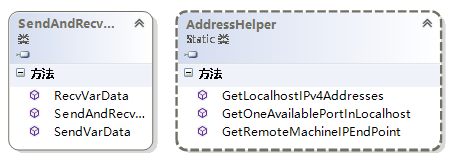
维护了一个connectionSocket，name，channel之间关系的数据结构，connectionSocket与name一一对应，一个channel可以有多个connectionSocket，同时如果某个connectionSocket没有channel，则默认channel为””，在此channel中信息不广播。这些对应关系都是使用Dictionary来实现的。

还有一个存放name的List。

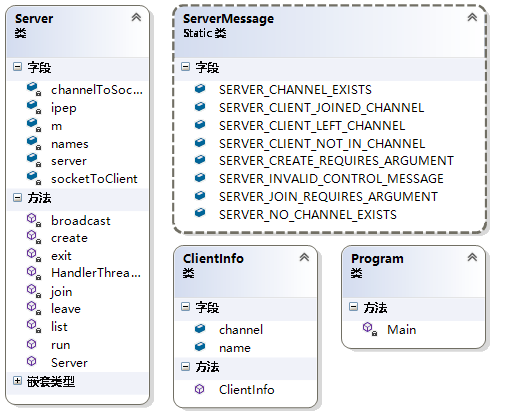
所有的对此数据结构的改动都只是对Dictionary的add和remove，故时间复杂度在O(1)。

1. 数据结构和算法的面向对象实现

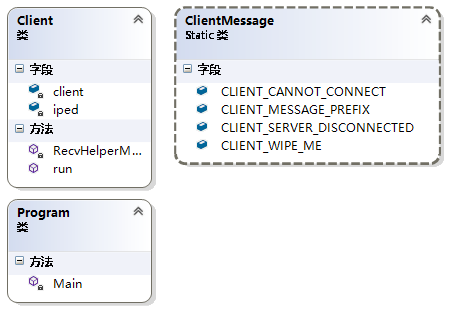
MyNetWorkLibrary项目：



Server项目：



Client项目：



#### 5 技术亮点、关键点及其解决方案

l  本程序的亮点：

我写的程序支持多个客户端同时连接，无阻塞！

程序支持发送和接受任意长数据

l  本程序的技术关键点

设计了MyNetWorkLibrary项目帮助发送和接收数据，规定发送和接收的数据开头4个字节为数据大小。

我采用了多线程技术实现多个客户端无阻塞的通信。

我使用Mutex来同步数据，使得一次只能有一个线程修改其中的数据结构。

l  遇到的技术难点及对应的解决方案：

处理客户端或者服务器端异常关闭的情况

****问题描述：****当用户直接关闭用户端时，辅助类MyNetWorkLibrary会抛出socketExpection异常，而关于此情况错误的信息提示在server中。

****最终的解决方案：****在MyNetWorkLibrary中捕获此异常，并重新抛出，在server中处理此异常。

#### 6 简要开发过程

5月24号      查找资料确定要使用的算法与数据结构

5月24号      完成系统设计

5月25号   进行MyNetWorkLibrary的发送和接受数据功能开发

5月25号     MyNetWorkLibrary的发送和接受数据功能开发完成，进入测试阶段

5月26号     程序主体功能开发完成

6月2号 程序完成处理各种错误信息的功能，增加程序的鲁棒性

6月3号   程序开发工作完毕，编写及整理文档