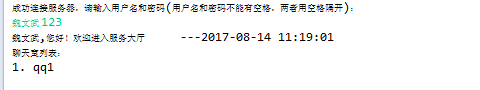
HappyChat2.0说明文档

### 使用说明

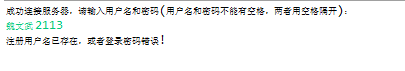
首先编译src目录下client和server包中的java源文件，再运行server.Server.class作为服务端，端口号为9999，运行多个client.Login.class作为多个客户端。注：单个用户可以同时在多个聊天室活动。

* 1. **注册/登录**

1. 客户端可以用“昵称+空格+密码”登陆进入聊天室大厅

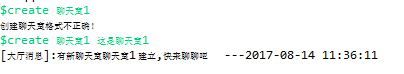


1. 输入的昵称不存在时，则为注册功能
2. 输入已存在昵称的密码错误时会提示：

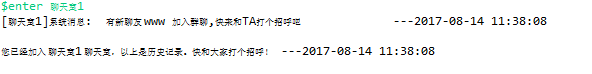


* 1. **创建/进入/退出聊天室**

1. 创建聊天室“$create + 空格 + 聊天室名字 + 空格 + 聊天室说明”



1. 进入聊天室“$enter + 空格 + 聊天室名字”



1. 退出聊天室：“$exit +空格 + 聊天室名字”



* 1. **群聊、私聊**

用户登录并进入某个聊天室后可以通过发送”$聊天室名字+空格+聊天内容”发送群消息:

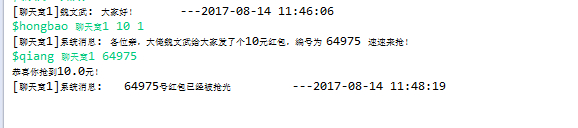


* 1. **发/抢红包**

1）使用命令“$hongbao +空格 + 房间号 + 空格 + 总金额+ 空格 + 个数”发红包

2）使用命令“$hongbao +空格 + 房间号 + 空格 + 总金额+ 空格 + 个数 + 空格 +拼手气”发拼手气的红包

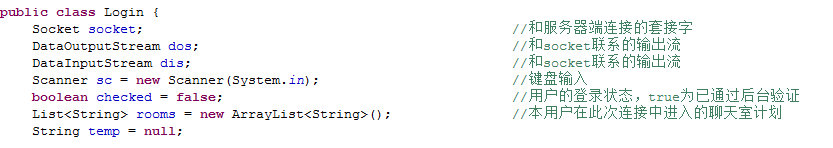
3）使用命令“$qiang + 空格 + 房间号 + 红包号码”进行抢红包



### 设计说明

* 1. **客户端设计：**

客户端信息处理由cliient.Login类处理。下图是Login类的相关属性:

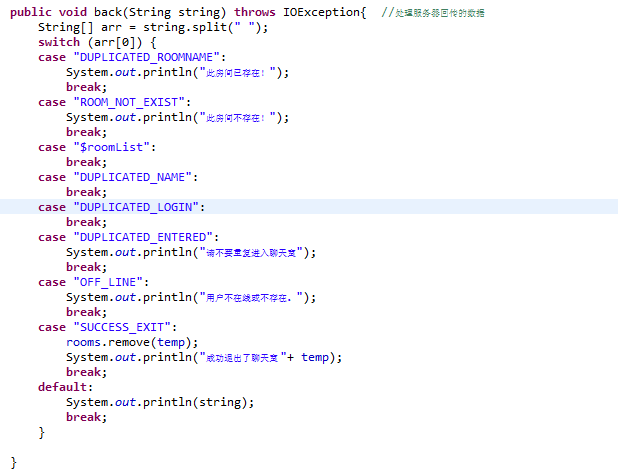


Login类中的主要函数：

Login.start() : main函数运行后直接进入start()函数进行相关初始化工作。该方法负责与后台服务器建立socket连接，然后再起一线程，用来监听键盘输入，同时自身来监听服务器发来的消息。



Login.back(): back()函数主要用来处理服务器端回传数据。



Login. check (): check()函数主要用来验证用户身份和注册用户信息的。



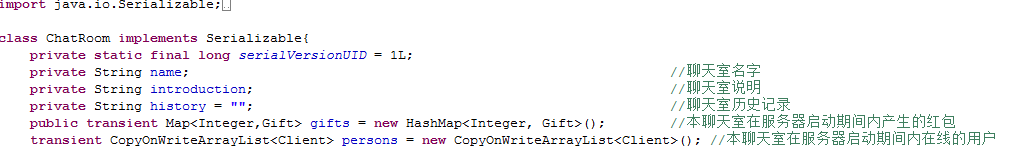
Login. handler ():handler ()函数主要用来验证各种用户输入信息的合法性，是Login类中最复杂的函数。只有通过handler函数验证的用户输入才能被传到服务器。

* 1. **服务端设计：**

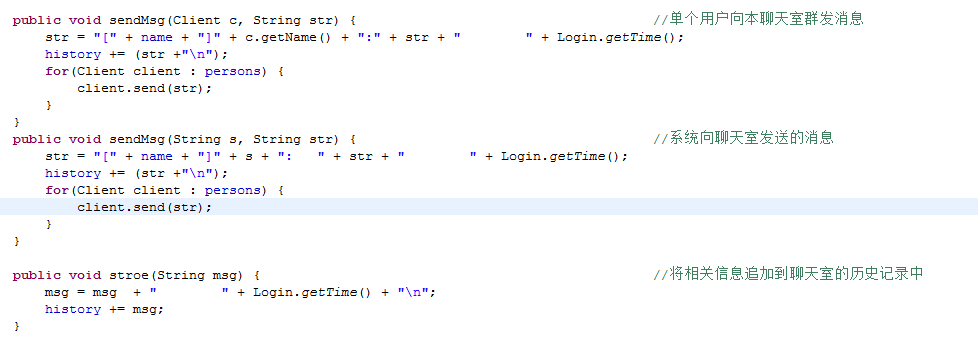
服务端主要由Server, ChatRoom, Gift三个类组成。Server类为后台服务启动类，ChatRoom 时聊天室类，Gift时红包类。

* + 1. **ChatRoom类**

ChatRoom主要负责聊天室的一些功能。下图是其相关属性及其说明。

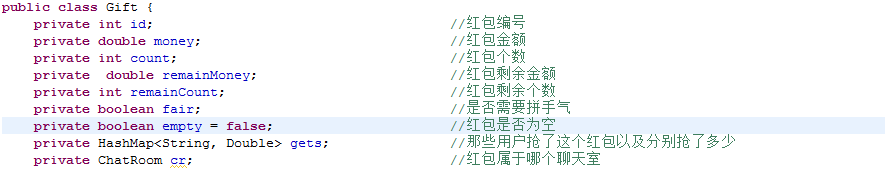


ChatRoom 主要函数有sendMsg和store。sendMsg由server端调用给聊天室群发消息，并将该条信息用store方法存入历史消息中。



* + 1. **Gift类**

Gift类是红包类，包含了红包相关属性信息和抢红包等动作。红包分为两类：公平的平分红包，大家抢到的金额都一样；拼手气红包，抢完后可以通过getMax函数计算最佳手气。

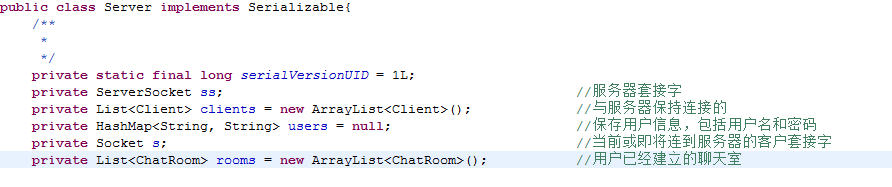
抢红包功能是由grabGift函数完成。对于公平红包，每个人抢到的金额是总金额除以人数，为了模拟实际情况，抢红包产生的金额都需要结果keep函数保留两位小数，因此最后一个抢红包的人将得到剩余金额，可能比其他人多一点点。拼手气红包策略：每个人最低抢0.01元，最多抢(剩余金额 – 剩余其他人\*0.01) 和

(剩余金额 / 剩余个数)\*2两者中较小者，这样会保证抢到的红包误差不至于特别大而影响玩家的体验。

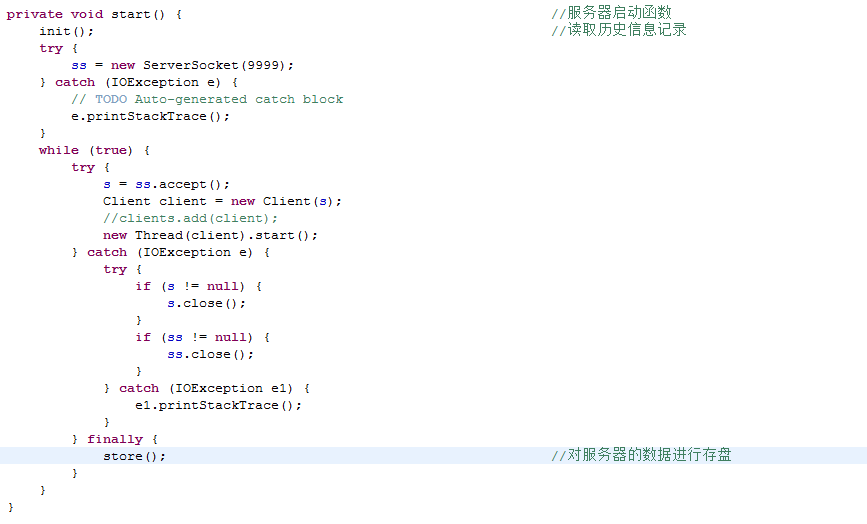


* + 1. **Server类**

Server类主要处理服务器端业务逻辑。



当服务器启动后首先执行start()函数。start()函数主要做三个工作：通过反序列化init函数读取服务器的历史记录，这里包含用户信息和聊天室记录信息；建立监听9999端口的服务器套接字，每当有一个客户连接到服务器就启动一个线程去单独处理客户信息。最后通过store函数将服务器的状态进行序列化存盘。



Server类有一个重要的内部类Server.Client类，这个类实现了Runnable类可以独立于Server主线程为单个用户提供服务。在run方法中调用handle函数处理客户端各种请求信息。

