Java11pro

对服务器改进（使之能读取信息的同时发送信息）：

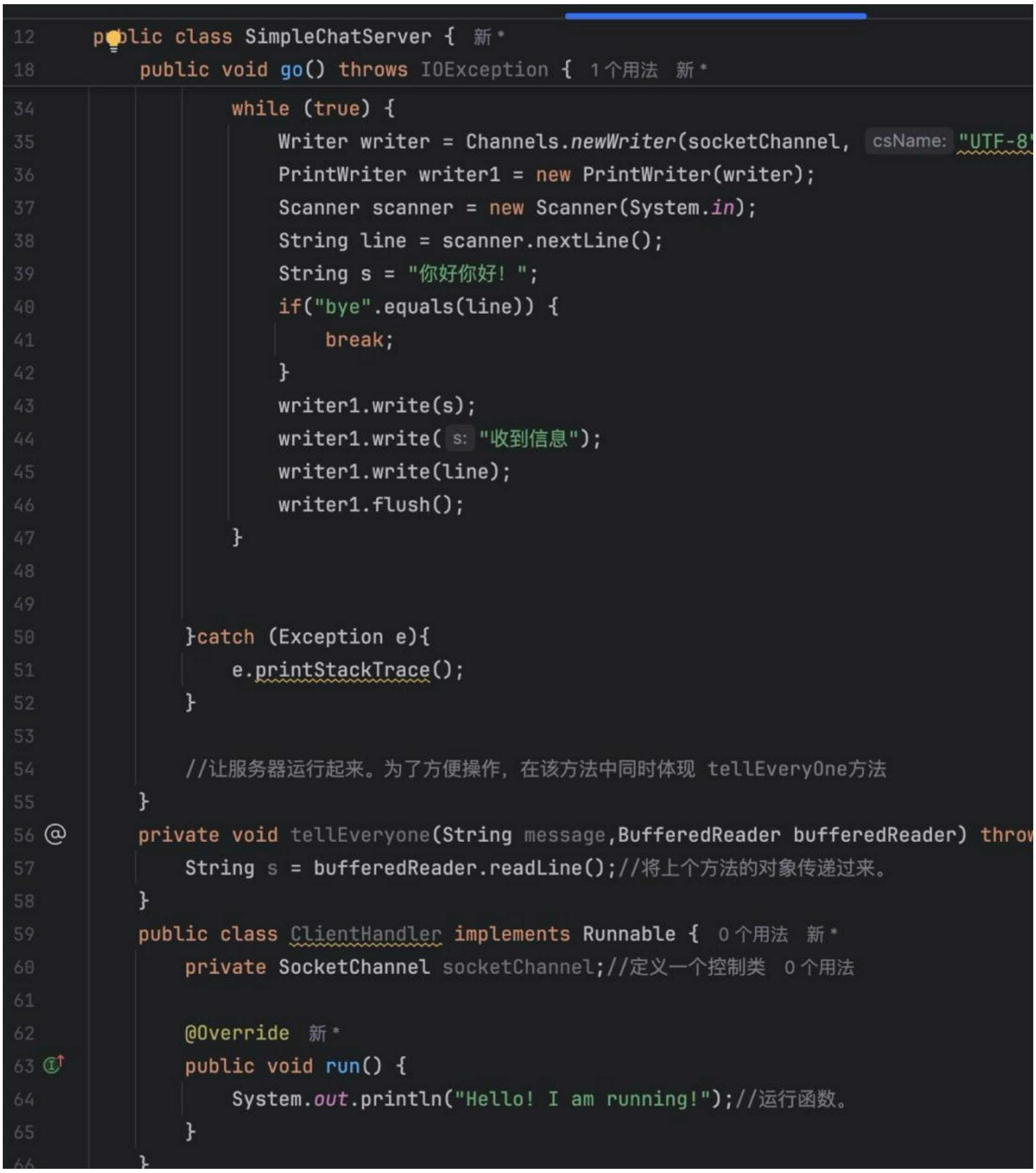
在改进的时候，在 server 中用 while 循环定义一个 PrintWriter，还是先通过 Writer 再构建 PrintWriter，然后把对应内容打印出来（记得在 while 循环插入 break 语句，然后记得关闭资源）

同理在客户端中，增加相应的 BufferedReader，读取到服务端传来的文本信息，然后可以将信息对应打印出来。

应该何时从服务器获取信息？

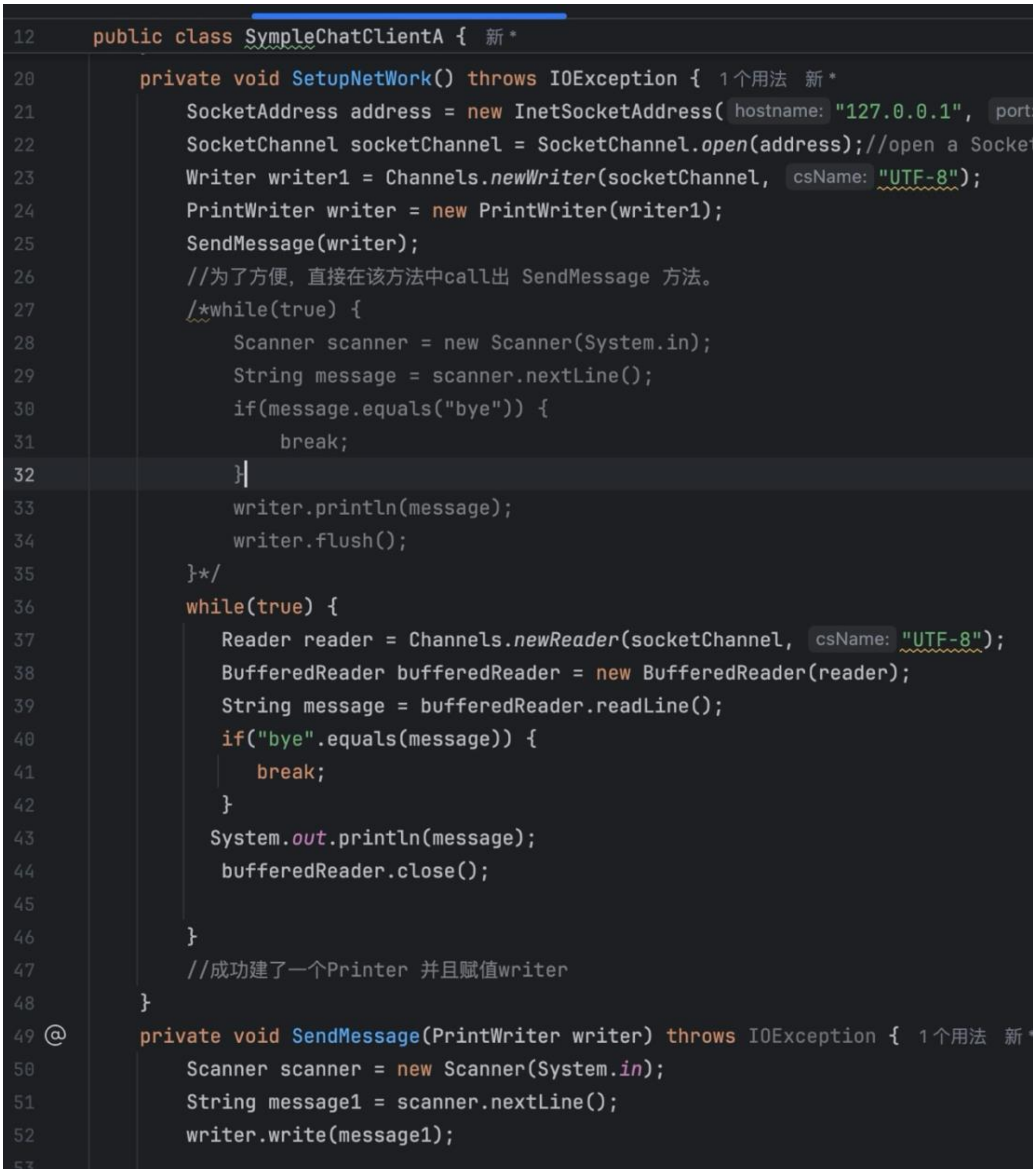
当需要实时更新信息或者与用户交互的时候。

改进代码截图：（服务端和客户端的顺序）



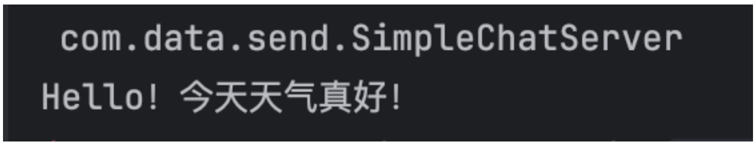
```
public class SimpleChatServer {  新 *
    public void go() throws IOException {  1个用法  新 *
            while (true) {
                Writer writer = Channels.newWriter(socketChannel,  csName: "UTF-8
                PrintWriter writer1 = new PrintWriter(writer);
                Scanner scanner = new Scanner(System.in);
                String line = scanner.nextLine();
                String s = "你好你好! ";
                if("bye".equals(line)) {
                    break;
                }
                writer1.write(s);
                writer1.write( s: "收到信息");
                writer1.write(line);
                writer1.flush();
            }


        }catch (Exception e){
            e.printStackTrace();
        }

        //让服务器运行起来。为了方便操作，在该方法中同时体现 tellEveryOne方法
    }
    private void tellEveryone(String message,BufferedReader bufferedReader) throw
        String s = bufferedReader.readLine();//将上个方法的对象传递过来。
    }
    public class ClientHandler implements Runnable {  0 个用法  新 *
        private SocketChannel socketChannel;//定义一个控制类  0 个用法

        @Override  新 *
        public void run() {
            System.out.println("Hello! I am running!");//运行函数。
        }
    }
```

下面是客户端代码截图：

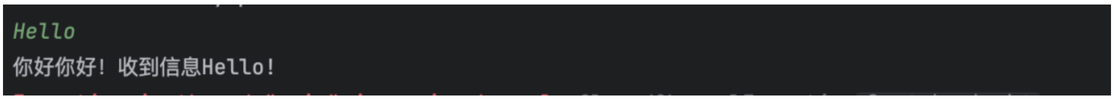


```
public class SympleChatClientA {  新 *

    private void SetupNetWork() throws IOException {  1个用法  新 *
        SocketAddress address = new InetSocketAddress( hostname: "127.0.0.1",  port
        SocketChannel socketChannel = SocketChannel.open(address);//open a Socke
        Writer writer1 = Channels.newWriter(socketChannel,  csName: "UTF-8");
        PrintWriter writer = new PrintWriter(writer1);
        SendMessage(writer);
        //为了方便，直接在该方法中call出 SendMessage 方法。
        /*while(true) {
            Scanner scanner = new Scanner(System.in);
            String message = scanner.nextLine();
            if(message.equals("bye")) {
                break;
            }
            writer.println(message);
            writer.flush();
        }*/
        while(true) {
           Reader reader = Channels.newReader(socketChannel,  csName: "UTF-8");
           BufferedReader bufferedReader = new BufferedReader(reader);
           String message = bufferedReader.readLine();
           if("bye".equals(message)) {
              break;
           }
          System.out.println(message);
           bufferedReader.close();

        }
        //成功建了一个Printer 并且赋值writer
    }
    private void SendMessage(PrintWriter writer) throws IOException {  1个用法  新
        Scanner scanner = new Scanner(System.in);
        String message1 = scanner.nextLine();
        writer.write(message1);
```

最后聊天室运行结果截图:



```
 com.data.send.SimpleChatServer
Hello! 今天天气真好!
```



```
Hello
你好你好! 收到信息Hello!
```