软件复用课程设计

client-server 应用程序

复用文档

小组： Team10

成员： 谢志杰 1352975

计鹏玥 1352914

王思尧 1352896

谢明玥 1352937

徐锦程 1353012

文档变更记录

|  |  |  |
| --- | --- | --- |
| 更改人 | 日期 | 更改内容 |
|  | 2016年3月28日 | 创建文档 |
|  |  |  |
|  |  |  |
|  |  |  |

**1.**应用程序简介

client-server 应用程序是一个java语言开发的客户端服务器连接应用程序，具备登录、客户端消息发送、服务端消息验证及消息转发、消息数量限制、日志记录等基本功能。

**2.**冗余检查及标示可复用构件

**2.1**冗余检查

2.1.1.由于应用程序中无论是客户端还是服务端均涉及到诸多的日志记录需求，频繁的io写操作会造成一定的代码冗余，因此在本应用程序中将io写操作标示为可复用构件。

2.1.2.由于应用程序需要在客户端与服务器之间传递消息，消息采用json格式，因而程序中涉及到很多对于json格式的操作，例如json的生成与解析，因此在本应用程序中将json相关操作标示为可复用构件。

2.1.3.由于应用程序涉及到数据库的存取操作，对于数据库的频繁查找操作是本应用程序的一个基本特点，因此在本应用程序中将数据库操作标示为可复用构件。

**2.2** 可复用构件列表

[2.2.1.io](http://2.2.1.io)写操作构件 － IOLog

2.2.2.消息构件 － Message

2.2.3.数据库构件 － DataSource

**3.** 可复用构件

**3.1 io**写操作构件－**IOLog**

**3.1.1.**构件介绍

这一构件的主要功能是针对于频繁的io写操作提供一个便利的接口使得io写操作的具体实现对外不可见，用户仅仅需要提供文件地址并使用提供的方法即可简单完成文件写操作。

**3.1.2.**功能说明

这一构件本质为一个方法类，用户使用构件时需要实例化一个类，并传入写文件地址、是否在文件尾写入作为参数，实例化后用户可以调用类中IOWrite的方法，传入写入的字符串作为参数，即可完成写操作的功能。

**3.1.3.**接口说明

1. **IOLog(String logAddr, boolean bAppend)**方法：

作用简述：

该方法实质为方法类的构造函数，用来指定写文件的地址以及是否在文件末尾执行写操作，使用该方法实例化方法类后可以使用后述的方法进行文件的写操作。

使用方法：

利用该方法实例化方法类，传入写文件地址、是否在文件尾写入作为参数，形如：

IOLog ioLog = new IOLog(“server.log”,true);

注意事项：

1.在指定是否在文件末尾执行写操作时，传入true表示在文件末尾执行写操作，传入false表示覆盖原文件执行写操作。

2.文件路径可以使用绝对路径也可以使用相对路径。

2. **IOWrite(String sWriten)**方法：

作用简述：

该方法用于将传入的字符串写入文件，通过上述构造函数产生的实例，调用该方法并传入想要写入的字符串作为参数，即可完成对于文件的写操作。

使用方法：

首先需要通过上述的方法实例化一个方法类，在上述方法中传入文件路径、是否在文件尾写入的参数后，即可调用该方法，该方法有一个string类型的参数，即传入想要写入指定文件的字符串，形如：

ioLog.IOWrite("forwarded message\n”);

注意事项：

1.只能向指定的文件执行写操作，而且只能以指定的方式（是否在文件尾写）执行写操作。

**3.2** 消息构件－ **Message**

**3.2.1.**构件介绍

这一构件的主要功能是构件与读取客户端与服务端通信时传递的消息，本质上是对于json生成与解析的方法的封装，并添加了线程标识符，用以表示消息的所属关系。

**3.2.2.**功能说明

用户可以通过该构件实现对消息的构造、消息所属线程的指定、消息内容的添加以及消息内容的解析等操作。

用户可以通过构造方法 Message指定消息所属的线程，同时也可以通过构造方法传递json格式的字符串用来生成消息的内容。用户也可以通过setValue方法添加消息内容，setValue方法通过重载可以接受不同类型的参数。用户可以通过getValue方法实现消息的解析。getOwner方法用于获取消息所属的线程，toString用于将JSONObject对象转换为json格式的字符串。

**3.2.3.**接口说明

1. **Message(String msg, long ownerThread)**方法：

作用简述：

该方法实质为方法类的构造函数，通过该方法可以实例化Message对象，并且可以指定消息所属的线程以及消息的内容。

使用方法：

利用该方法实例化Message方法类，其中需要传递两个参数，第一个参数为string类型，为json格式的字符串，即消息的内容，第二个参数为long类型，为消息所属线程的标识符，形如：

Message msg = new Message("{}", this.getId());

注意事项：

1.该方法中指定的消息的内容可以为空的json，即形如”{}”的字符串，但该参数不可为空。

2. **getOwner()**方法：

作用描述：

该方法用于获取消息所属消息的标识符，通过调用一个实例化的Message对象的该方法，即可获得这个实例化的Message对象的标识符。

使用方法：

对于一个实例化的Message对象，可以直接调用该方法，该方法无需传递任何参数，返回类型为long型，即线程标识符，形如：

Long ownerThread = msg.getOwner();

注意事项：

1.该方法返回的是其所属消息的线程的标识符，该标识符在上述构造函数中指定。

3. **setValue(String key, String value)**， **setValue(String key, long value)**方法：

作用描述：

该方法用于添加消息内容，通过传递两个参数作为key和value值，可以将这个键值对加入到消息内容中。该方法进行重载，支持不同类型的value值，包括string类型和long类型。

使用方法：

对于一个实例化的Message对象，可以直接调用该方法，该方法有两个参数，第一个参数为string类型的key，第二个参数为string类型或者long类型的value，调用该方法后，传入的键值对会添加到实例化的消息的内容中。形如：

msg.setValue("event", “login");

注意事项：

1.该方法通过重载可以接受两种不同类型的value值，即第二个参数可以接受不同类型的值，包括string类型与long类型。

4. **getValue(String key)**方法：

作用描述：

该方法用于获取消息内容，通过传入key值，可以获取到对应的value值。

使用方法：

对于一个实例化的Message对象，可以直接调用该方法，该方法唯一的参数是string类型的key值，返回类型为string类型的value值，形如：

String sValue = msg.getValue(“target”);

注意事项：

1.尽管上述setValue方法实现了重载，value类型可以接受不同类型的参数，包括string类型与long类型，但是本方法返回的value类型仅为string类型。

5. **toString()**方法：

作用描述：

该方法用于将JSONObject对象转换为json格式的字符串。

使用方法：

对于一个实例化的Message对象，可以直接调用该方法，该方法没有输入参数，返回类型为string类型，即json格式的字符串。形如：

String jsonMessage = msg.toString();

注意事项：

1.该方法会将实例对象中的内容转换为json格式的字符串，转换后若继续通过setValue方法添加内容，则需要在此调用本方法才能得到添加后的json格式的字符串。

**3.3** 数据库构件－ **DataSource**

**3.2.1.**构件介绍

这一构件的主要功能是对于数据库的查找操作，包括密码查找等基本数据库查找功能。

**3.2.2.**功能说明

用户可以根据自己的实际需要选择不同的数据库，并配置好数据库的用户名和密码，即可通过该构件连接到数据库。

使用密码查找功能需要在数据库中建立tb\_user表，表中包含username及password两个属性，配置好相应数据库表后即可使用该构件提供的密码查找功能。

**3.2.3.**接口说明

1.**getPassword(String username)**方法：

作用描述：

该方法用于根据指定的用户名在相应的数据库中查找用户名对应的密码，即执行相应的sql语句对数据库中对应的表执行select操作。

使用方法：

对于一个实例化的DataSource对象，可以直接调用该方法，该方法有一个string类型的参数，即想要查找的用户名，返回类型为string 类型，即输入的用户名对应的密码。形如：

DataSource dataSource = new DataSource();

String username = “Tom”;

String password = dataSource.getPassword(username);

注意事项：

1.该方法需要确保数据库中存在tb\_user表，并且表中存在username以及password属性。

2.若输入的用户在数据库tb\_user表中存在，则返回该用户对应的密码，若输入的用户不存在，则返回“rootpassword”。

**3.2.4.**注意事项

该构件功能会随着应用程序功能的添加而添加，应用程序中数据库相关的操作均有该构件支持。