- 考试二
  - 运行展示
    - 注册账号
    - 展示注册用户
    - 留言
    - 显示留言
  - 数据持久化
    - 存储方式
    - 读取文件
    - 存储文件

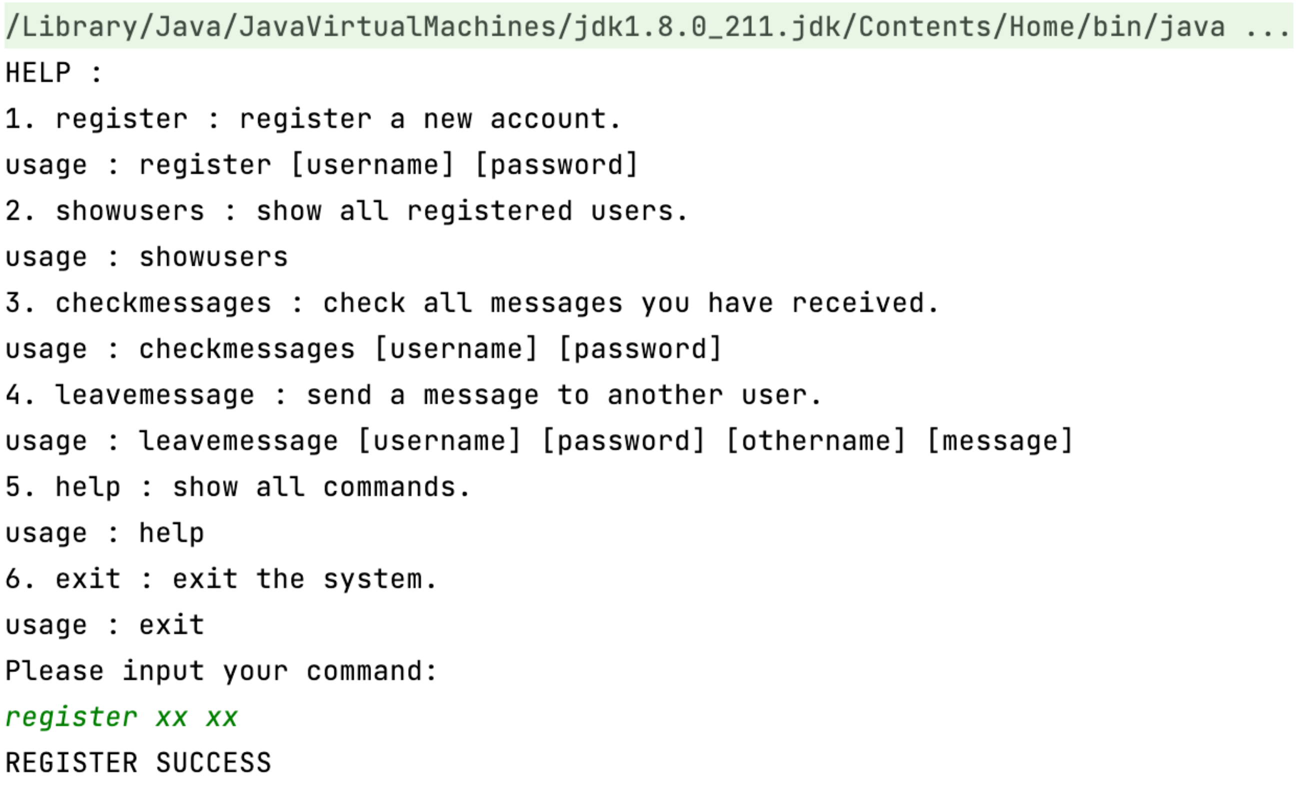
考试二

姓名：杨玺璇　　**学号**：2020302721　**班级**：14012005

运行展示

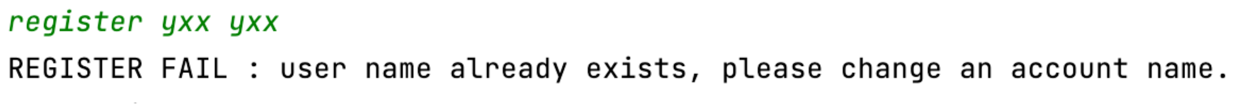
注册账号

1. 注册成功



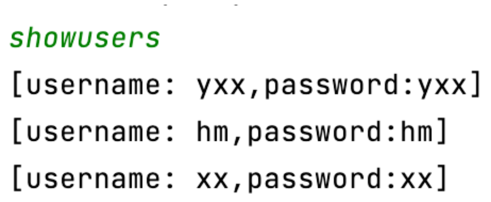
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home/bin/java ...
HELP :
1. register : register a new account.
usage : register [username] [password]
2. showusers : show all registered users.
usage : showusers
3. checkmessages : check all messages you have received.
usage : checkmessages [username] [password]
4. leavemessage : send a message to another user.
usage : leavemessage [username] [password] [othername] [message]
5. help : show all commands.
usage : help
6. exit : exit the system.
usage : exit
Please input your command:
register xx xx
REGISTER SUCCESS
```

2. 注册失败



```
register yxx yxx
REGISTER FAIL : user name already exists, please change an account name.
```

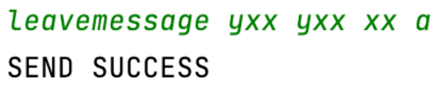
展示注册用户



```
showusers
[username: yxx,password:yxx]
[username: hm,password:hm]
[username: xx,password:xx]
```

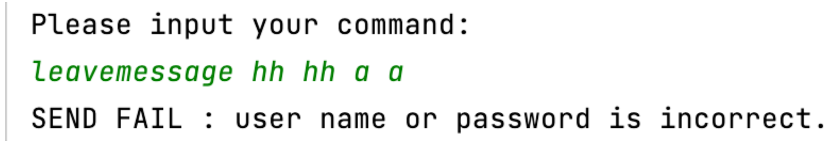
留言

1. 留言成功



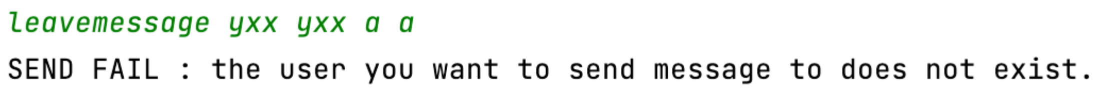
```
leavemessage yxx yxx xx a
SEND SUCCESS
```

2. 发送者用户错误



```
Please input your command:
leavemessage hh hh a a
SEND FAIL : user name or password is incorrect.
```

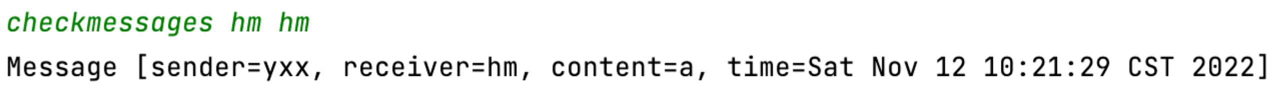
3. 接收者用户错误



```
leavemessage yxx yxx a a
SEND FAIL : the user you want to send message to does not exist.
```

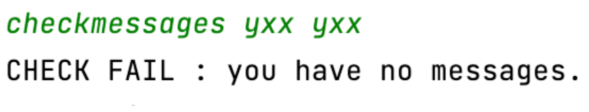
显示留言

1. 显示成功



```
checkmessages hm hm
Message [sender=yxx, receiver=hm, content=a, time=Sat Nov 12 10:21:29 CST 2022]
```

2. 显示失败

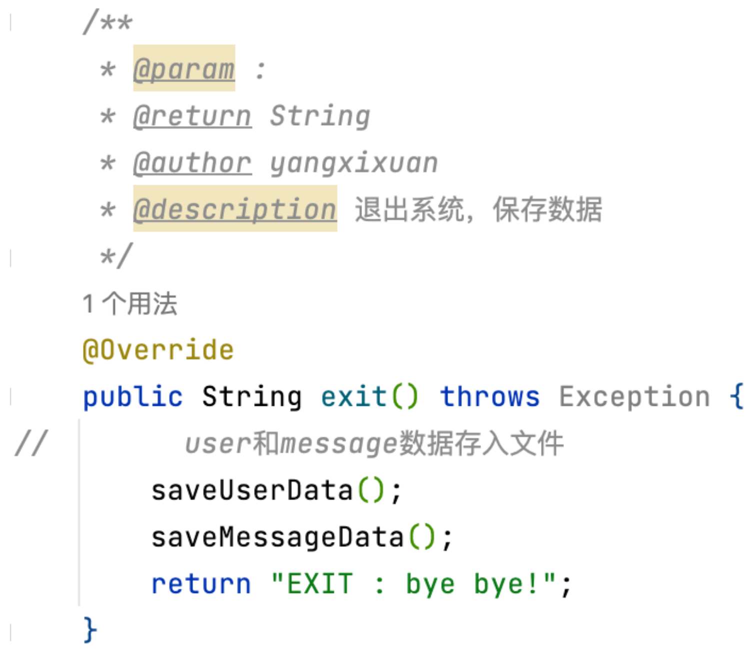


```
checkmessages yxx yxx
CHECK FAIL : you have no messages.
```

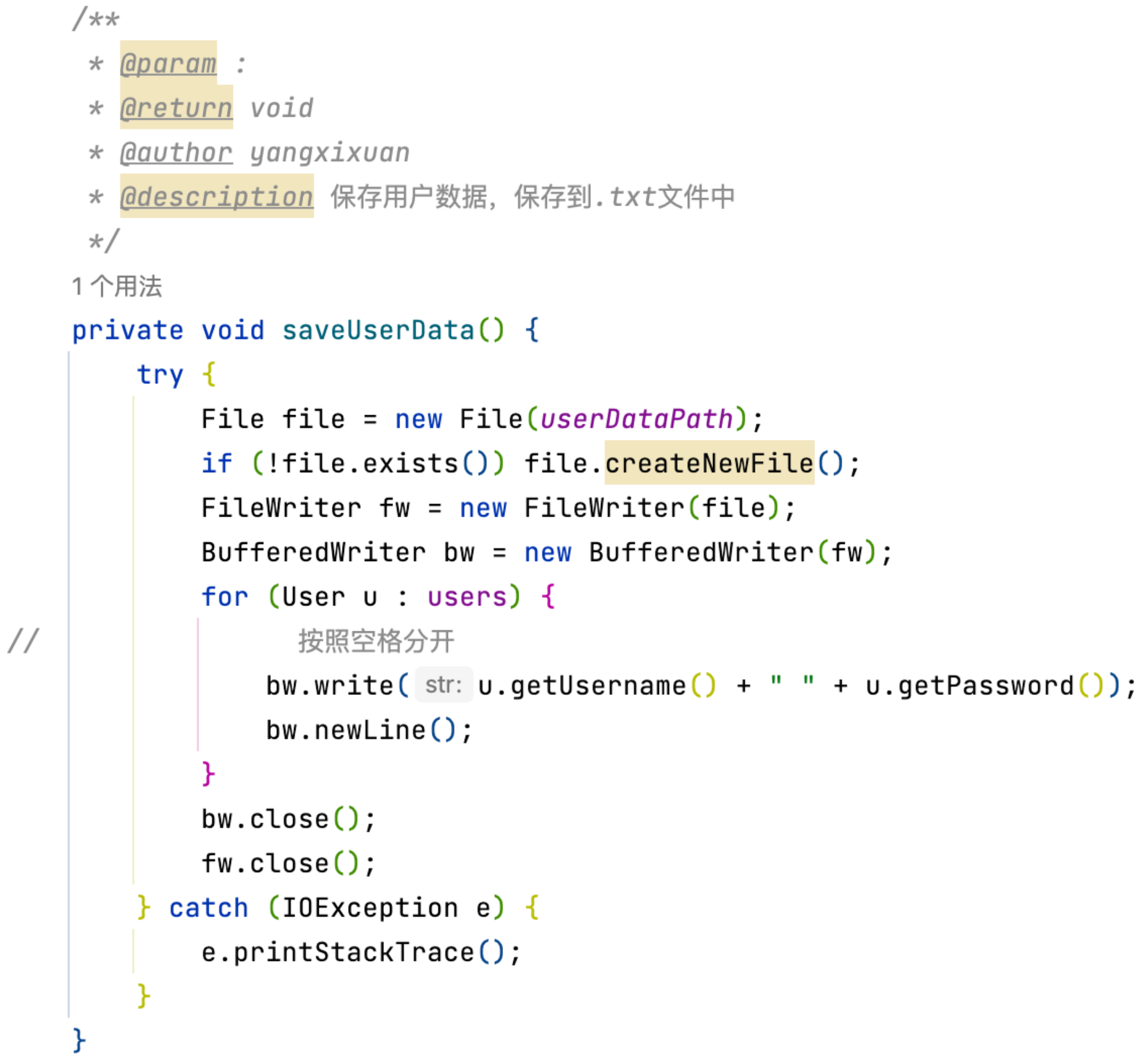
数据持久化

存储方式

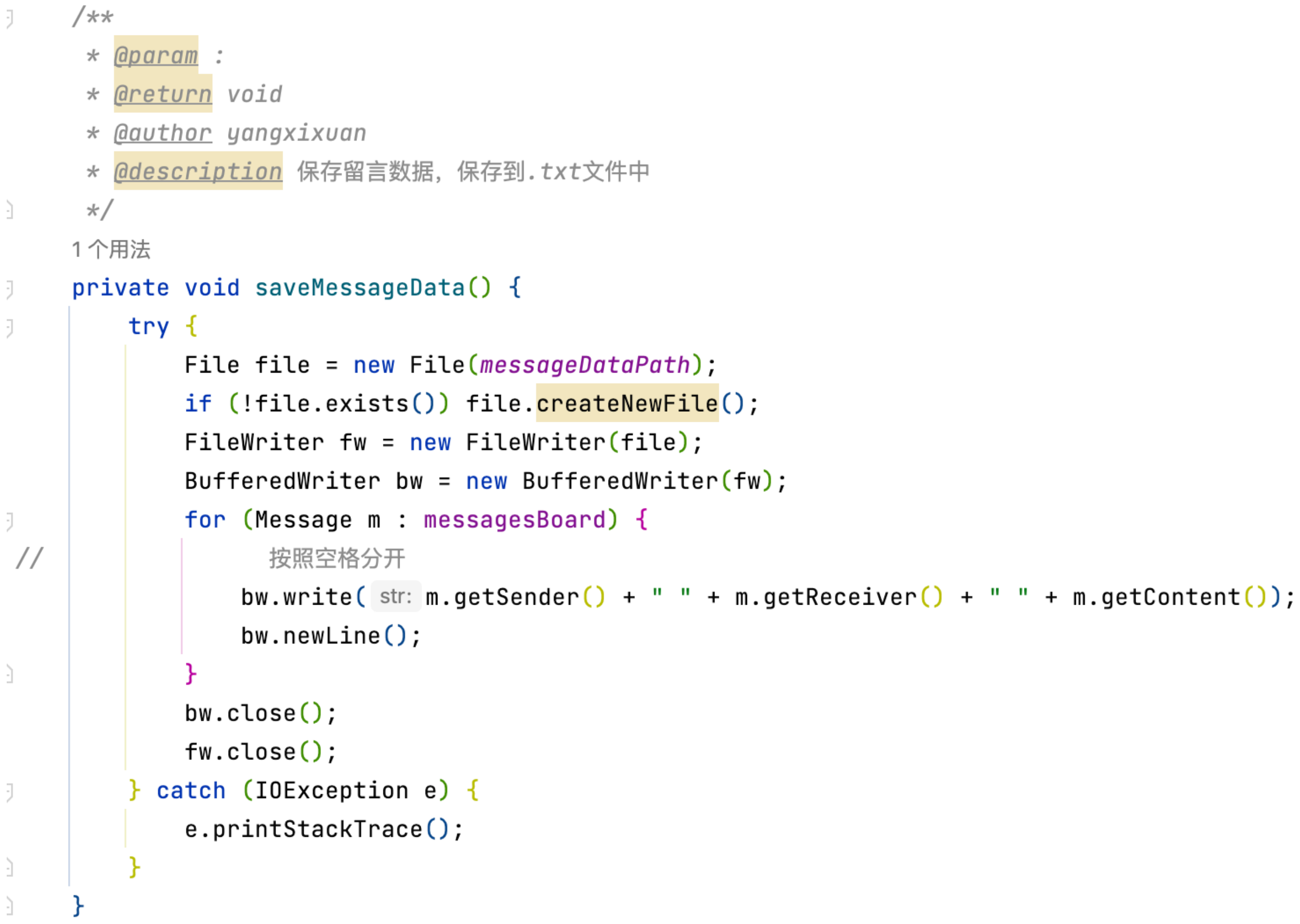
采用自定义的格式，进行对应的存储和读取，在每次exit系统时保存



```java
    /**
     * @param :
     * @return String
     * @author yangxixuan
     * @description 退出系统，保存数据
     */
    1 个用法
    @Override
    public String exit() throws Exception {
//        user和message数据存入文件
        saveUserData();
        saveMessageData();
        return "EXIT : bye bye!";
    }
```



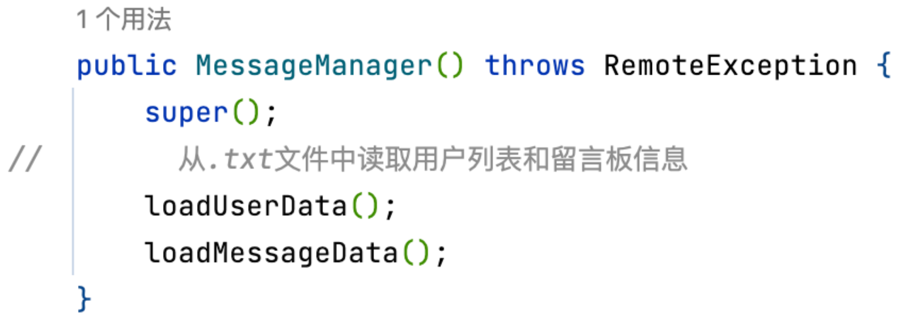
```java
    /**
     * @param :
     * @return void
     * @author yangxixuan
     * @description 保存用户数据，保存到.txt文件中
     */
    1 个用法
    private void saveUserData() {
        try {
            File file = new File(userDataPath);
            if (!file.exists()) file.createNewFile();
            FileWriter fw = new FileWriter(file);
            BufferedWriter bw = new BufferedWriter(fw);
            for (User u : users) {
//                按照空格分开
                bw.write( str: u.getUsername() + " " + u.getPassword());
                bw.newLine();
            }
            bw.close();
            fw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```



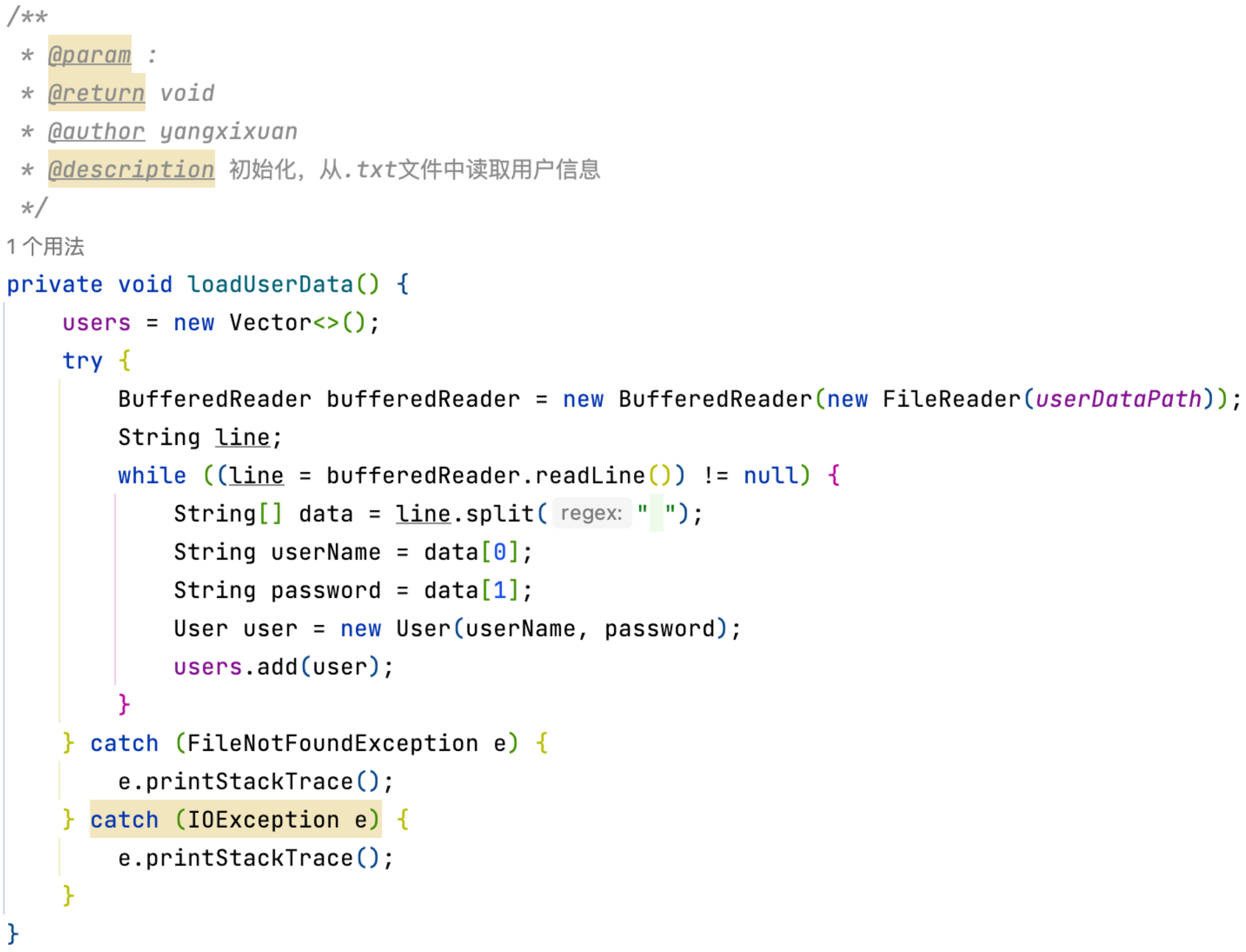
```
/**
 * @param :
 * @return void
 * @author yangxixuan
 * @description 保存留言数据，保存到.txt文件中
 */
1 个用法
private void saveMessageData() {
    try {
        File file = new File(messageDataPath);
        if (!file.exists()) file.createNewFile();
        FileWriter fw = new FileWriter(file);
        BufferedWriter bw = new BufferedWriter(fw);
        for (Message m : messagesBoard) {
//              按照空格分开
            bw.write( str: m.getSender() + " " + m.getReceiver() + " " + m.getContent());
            bw.newLine();
        }
        bw.close();
        fw.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

读取文件

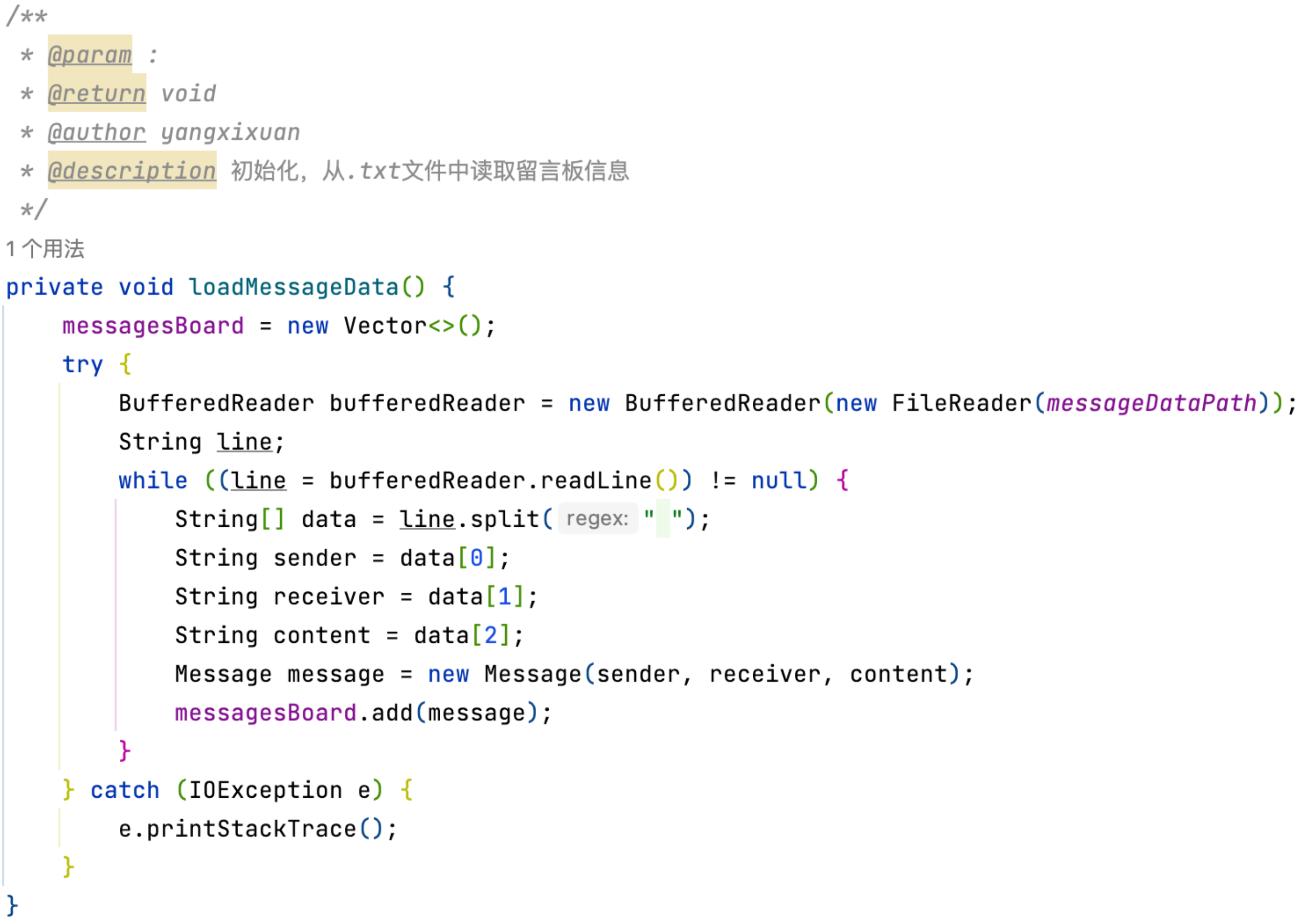
每次启动系统的时候读入



```
1 个用法
public MessageManager() throws RemoteException {
    super();
//      从.txt文件中读取用户列表和留言板信息
    loadUserData();
    loadMessageData();
}
```

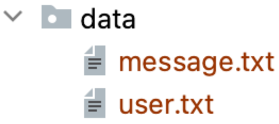


```
/**
 * @param :
 * @return void
 * @author yangxixuan
 * @description 初始化，从.txt文件中读取用户信息
 */
1 个用法
private void loadUserData() {
    users = new Vector<>();
    try {
        BufferedReader bufferedReader = new BufferedReader(new FileReader(userDataPath));
        String line;
        while ((line = bufferedReader.readLine()) != null) {
            String[] data = line.split( regex: " ");
            String userName = data[0];
            String password = data[1];
            User user = new User(userName, password);
            users.add(user);
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```



```
/**
 * @param :
 * @return void
 * @author yangxixuan
 * @description 初始化，从.txt文件中读取留言板信息
 */
1 个用法
private void loadMessageData() {
    messagesBoard = new Vector<>();
    try {
        BufferedReader bufferedReader = new BufferedReader(new FileReader(messageDataPath));
        String line;
        while ((line = bufferedReader.readLine()) != null) {
            String[] data = line.split( regex: " ");
            String sender = data[0];
            String receiver = data[1];
            String content = data[2];
            Message message = new Message(sender, receiver, content);
            messagesBoard.add(message);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

存储文件



```
data
    message.txt
    user.txt
```