

# 辽 宁 工 业 大 学

## 《Java 语言程序设计》课程设计（论文）

题目： 基于 AI 接口的人机交流

院（系）： 软件学院

专业班级： 软件工程 2101 班

学 号： 211305001

学生姓名： 李昊岩

指导教师： 翟宝峰

教师职称： 副教授

起止时间： 2021. 12. 13-2022. 02. 24

## 课程设计任务及评语

院（系）：软件学院

教研室：软件教研室

学 号	211305001	学生姓名	李昊岩	专业班级	软件工程 2101 班
课程设计题目	基于 AI 接口的人机交流				
课程设计（论文）任务	<p><b>任务与要求：</b></p> <p>（1）. 掌握面向对象程序设计语言 JAVA 的基本语法</p> <p>（2）. 掌握 JAVA 面向对象设计的思想</p> <p>（3）. 掌握 V JAVA 6.0 与 MyEclipse JAVA 开发工具的使用</p> <p>（4）. 运用标准 JAVA 规范设计应用程序</p> <p>设计过程中，要严格遵守设计的时间安排，听从指导教师的指导。</p> <p>正确地完成上述内容， 规范完整地撰写出设计报告。</p>				
指导教师评语及成绩	<p>成绩：_____</p> <p style="text-align: right;">指导教师签字：_____</p> <p style="text-align: right;">2022 年 02 月 27 日</p>				

## 目录

第 1 章课程设计目的与要求.....	1
1.1 设计目的 .....	1
1.2 实验环境 .....	1
1.3 预备知识 .....	1
1.4 设计要求 .....	2
第 2 章 课程设计内容.....	3
2.1 课题描述.....	3
2.2 系统设计.....	3
2.2.1 功能分析 .....	3
2.2.2 面向对象设计 .....	3
2.2.3 类成员描述 .....	5
2.3 程序实现 .....	6
2.3.1 源码分析 .....	6
2.3.2 运行结果截图 .....	11
第 3 章 总结.....	15
参考文献.....	17

## 第 1 章 课程设计目的与要求

### 1.1 设计目的

将理论教学中涉及到的知识点贯穿起来,对不同的数据类型、程序控制结构、数据结构作一比较和总结,结合设计题目进行综合性应用,对所学知识达到融会贯通的程度。

通过课程设计,学生在下述各方面的能力应该得到锻炼:

- (1) 进一步巩固、加深学生所学专业课程《JAVA 语言教程》的基本理论知识,理论联系实际,进一步培养学生综合分析问题,解决问题的能力。
- (2) 全面考核学生所掌握的基本理论知识及其实际业务能力,从而达到提高学生素质的最终目的。
- (3) 利用所学知识,开发小型应用系统,掌握运用 JAVA 语言编写调试应用系统程序,训练独立开发应用系统,进行数据处理的综合能力。
- (4) 对于给定的设计题目,如何进行分析,理清思路,并给出相应的数学模型。
- (5) 掌握面向对象程序设计的方法。
- (6) 熟练掌握 JAVA 语言的基本语法,灵活运用各种数据类型。
- (7) 进一步掌握在集成环境下如何调试程序和修改程序。

### 1.2 实验环境

硬件环境:

CPU: 11th Gen Intel(R) Core(TM) i5-11400H

内存: 16GB DDR4 内存

硬盘: 三星 512GB 固态硬盘

软件环境:

Java 环境: java version "11.0.12" 2021-07-20 LTS

IDE: IntelliJ IDEA 2021.3 (Ultimate Edition)

操作系统: Windows 11 专业工作站版

### 1.3 预备知识

熟悉 JAVA 语言程序设计的基本知识及 IDEA 编辑器的使用方法。

#### 1.4 设计要求

1. 仔细分析设计题目，画出程序流程图，编写程序源代码。
2. 积极上机调试源程序，增强编程技巧与调程能力。
3. 认真书写课程设计预习报告, 课程设计说明书。
4. 遵守课程设计要求 and 机房管理制度，服从指导教师的安排，确保课程设计的顺利完成课程设计内容。

## 第2章 课程设计内容

### 2.1 课题描述

人工智能(Artificial Intelligence), 英文缩写为AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

人工智能是计算机科学的一个分支, 它企图了解智能的实质, 并生产出一种新的能以人类智能相似的方式做出反应的智能机器, 该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。人工智能从诞生以来, 理论和技术日益成熟, 应用领域也不断扩大, 本系统旨在使用人工智能接口的自然语言处理功能来实现与电脑进行对话, 并加入了一些容错处理。

### 2.2 系统设计

多线程执行。

为实现聊天功能使用了图灵和腾讯云(容错处理)接口, 将系统分为10个类分别实现网络请求, 线程管理, 信息处理, Json解析, 图形界面展示等功能。

#### 2.2.1 功能分析

要完成用户界面的展示, 用户消息的录入, 数据的后台处理, 发送和接受网络请求, 调用接口等功能。接收到服务器发送来的消息后要展示在用户界面上以完成一轮对话。

#### 2.2.2 面向对象设计

设计的类分别为 ChatBot, TuLBot, Bean, ResultBean, TencentBean, Control, NetThread, ChatGui, Dispose, HttpClient。

Java 中数据类型的说明:

bean 包: 创建 Json 格式化模型, 将 Json 数据映射到 bean 中的类中以实现序列化和反序列化。

control 包: 控制器和线程控制器, 用户控制整个系统数据流的走向, 调用其他类和多线程控制。

gui 包: 界面展示, 显示系统交互界面, 按钮的监听, 模型的序列化和反



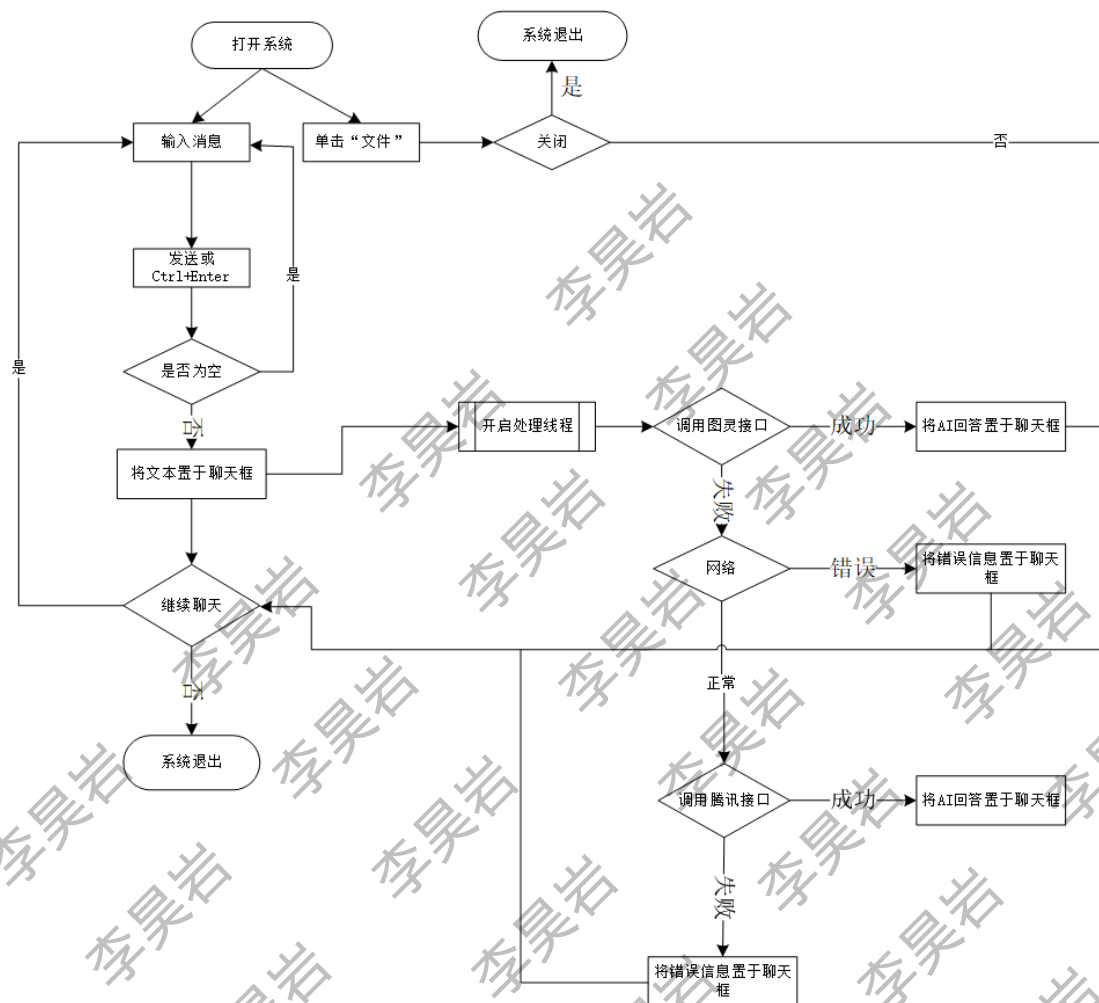


图 2-2 总体流程图

### 2.2.3 类成员描述

ChatBot 类：由腾讯云提供的类，配合腾讯 SDK 可实现自动调用 AI 接口，返回一条 Json 格式的响应。

TuLBot 类：图灵机器人处理消息类，通过构造网络请求，接收图灵接口的响应，返回值为 Json 格式。

Bean, ResultBean, TencentBean 类：Gson 将 Json 数据转换为 String 的模型

Control 类：控制类，构造请求，调用各个功能类实现响应功能。

NetThread 类：多线程类，网络线程为耗时线程，所有网络请求通过本类实现，保证程序不被阻塞。

ChatGui 类：程序主界面。

Dispose 类：监听事件类，监听用户所做的操作。

HttpClient 类：网络请求类，构造网络请求，发送和接受网络请求和响应。



## 2.3 程序实现

使用 Java 语言完成本程序的开发，发布和容错等处理。

### 2.3.1 源码分析

主要类：

ChatGui 类核心代码：

1. 界面使用空布局，需要计算各个组件的位置，并且保证在改变窗口大小时布局显示正常。

```
this.addComponentListener(new ComponentAdapter() { //自动窗口变化
@Override
public void componentResized(ComponentEvent e) {
    showPanel.setBounds(1, 1, getWidth() - 16, (int) (getHeight() *
0.65) - 25); //设置展示区域大小和位置
    areaScroll.setBounds(1, showPanel.getHeight() + 2,
showPanel.getWidth(), (int) (getHeight() * 0.25) - 25); //设置发送区
域大小和位置
    sendBt.setBounds(getWidth() - 100, showPanel.getHeight() +
areaScroll.getHeight() + 6, 85, 30); //设置发送按钮大小和位置
}
}); //设置窗口自动变化
```

2. 各个组件的监听

```
sendBt.addActionListener(new Dispose(sendArea, chatContent)); //设
置发送按钮单击时的监听
item.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
    System.exit(0);
}}); //设置菜单项的单击事件
```

## Dispose 类主要核心代码

1. 处理发送的按钮并且启动多线程处理网络请求，验证消息是否为空，在单击发送按钮后将发送区域清空，保证滚动条在发送完消息后更新。

```
String content = sendArea.getText();//获取发送区域的文本
content = content.trim().replaceAll("\n", "").replaceAll("\r",
 "").trim();//删除回车和空格。
if (content != null && !content.equals("")) { //判断消息是否为空
    date = new Date(System.currentTimeMillis());
    chatContent.append(format.format(date) + " ");
    chatContent.append("我:" + content + "\n");
    netThread = new NetThread(chatContent, sendArea, content); //使
    netThread.start();
} else {
    date = new Date(System.currentTimeMillis());
    chatContent.append(format.format(date) + " ");
    chatContent.append("系统:聊天内容不能为空" + "\n"); //如果为空输
    sendArea.setText(""); //设置文本区域为空
    DefaultCaret caret = (DefaultCaret) chatContent.getCaret(); //滚
    caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE); //滚动条更新
```

## NetThread 类核心代码

1. 日期时间的格式化，调用控制类实现信息的发送和接受，展示接收到的消息。

```
@Override
    public void run() {
        super.run();
        format = new SimpleDateFormat("HH:mm:ss");//日期时间的格式化
        control = new Control();//实例化一个控制对象
        result = control.parse(content);//调用控制类的格式化方法
        for (String res : result) {
            date = new Date(System.currentTimeMillis());
            chatContent.append(format.format(date) + " ");
            chatContent.append("AI:" + res + "\n");//在展示区域展示消息
        }
    }
```

## Control 类核心代码

1. Json 的格式化，内部类的封装，调用接口方法

```
public String formation(String text) {
    bean = new Bean();
    Bean.Perception.InputText inputText = new
    Bean.Perception.InputText();//Json 模型内部类的创建
    bean.setReqType("0");//设置请求的 Json 数据
    inputText.setText(text);//设置发送的信息
    Bean.Perception perception = new
    Bean.Perception();//Json 模型内部类的创建
    perception.setInputText(inputText);//信息的封装
    bean.setPerception(perception);//内部类的封装
    bean.setUserInfo(userInfo);//内部类的封装
    Gson gson = new Gson();//实例化 Gson 格式化数据
    TuLBot lBot = new TuLBot();
    return lBot.Respon(url, gson.toJson(bean));//调用机器人的方法发送 Json 数据，得到接口的响应
}
```

2. Json 的解析, Json 字符串的构造和拼接, 获取 Json 格式中的信息, 封装到 List 中, 判断调用的接口。

```

public List<String> parse(String text) {
    Random r = new Random(); //三个用户名密码随机使用一个
    code.add("6ce2fe*****");
    code.add("75d5be*****");
    code.add("1e1455*****");
    userInfo = new Bean.UserInfo();
    userInfo.setApiKey(code.get(r.nextInt(3))); //构造请求 Json 数
    据, 设置请求的用户名和密码
    userInfo.setUserId("628***");
    String str = formation(text); //格式化数据并得到结果
    Gson gson = new Gson();
    ResultBean resultBean = gson.fromJson(str, ResultBean.class); //
    得到 Json 数据后, 使用 Gson 将数据按照 ResultBean 的模型转换
    List<ResultBean.Results> list = resultBean.getResults(); //返回
    值为数组映射到 List 中
    List<String> result = new ArrayList<>();
    if (resultBean.getIntent().getCode() != 10004 ||
    resultBean.getIntent().getCode() != 10005) { //响应成功的代码为
    10004, 通过模型中封装的方法过去数据
    ChatBot chatBot = new ChatBot(); //当图灵接口失效后使用腾讯云接
    口
    try {
        result.add(TencentParse(chatBot.chatbot(text))); //调用 chatbot
        方法, 返回值为腾讯云接口的 Json 格式数据
    } catch (TencentCloudSDKException e) {
        result.add("所有系统失效, 请联系管理员解决错误!");
        return result;
    }
    return result;
    } else {
    for (ResultBean.Results rs : list) {
        if (rs.getResultType().equals("text")) //Json 中封装了返回的数
    
```

据类型，本系统只处理 text 和 url 类型

```

        result.add(rs.getValues().getText());
    else if (rs.getResultType().equals("url"))
        result.add(rs.getValues().getUrl());
    else
        result.add("此类型信息无法显示");
    }
}
return result; }

```

TuLBot 类核心代码

1. 发送数据，获取接口信息

```

public String Respon(String url, String json) {
    HttpClient httpClient;//创建 HttpClient 类
    httpClient = new HttpClient();
    String result = httpClient.sendPost(url, json);//与服务器
    交互，发送请求。
    return result;
}

```

HttpClient 类核心代码

1. 建立连接，发送数据，获取数据，容错处理

```

URL Url = new URL(url); // 打开和 URL 之间的连接
URLConnection conn = Url.openConnection();
// 发送 POST 请求必须设置如下两行
conn.setDoOutput(true);
conn.setDoInput(true);
out = new OutputStreamWriter(conn.getOutputStream(), "UTF-8");//
发送请求参数
out.write(param);// flush 输出流的缓冲
out.flush();// 定义 BufferedReader 输入流来读取 URL 的响应
in = new BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));
String line;

```

```
while ((line = in.readLine()) != null) {
    result += line;}
}
```

数据模型部分代码

ResultBean 类：类，属性的封装，模型的构建

```
private Intent intent;// 将内部类 Intent 作为本类的属性
private List<Results> results; //将内部类作为泛型

public static class Intent { //内部类
    @SerializedName("code")
    private int code;

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }
}
```

### 2.3.2 运行结果截图

1. 启动界面，输入内容

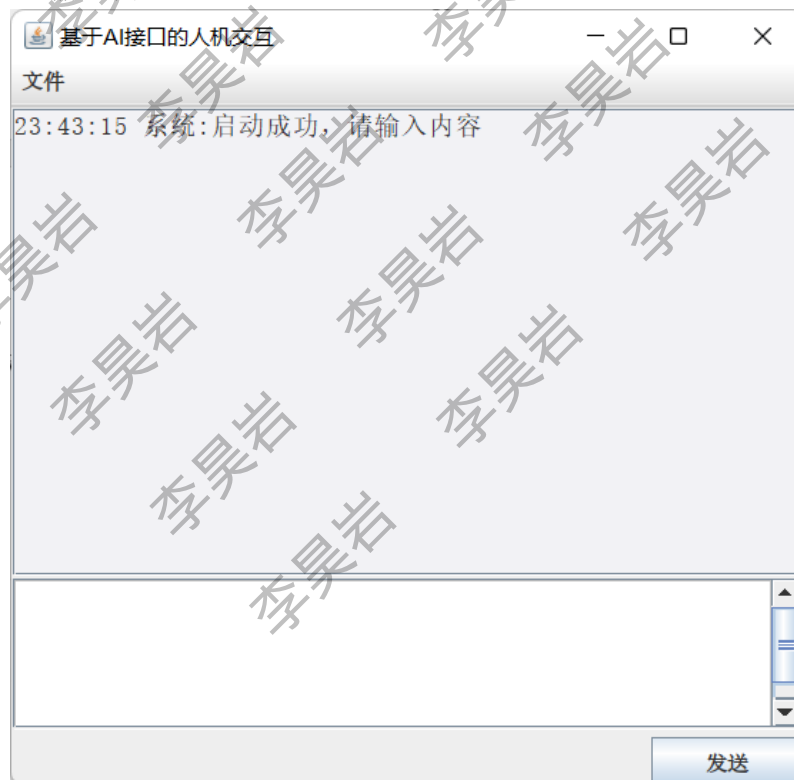


图 2-3 启动界面

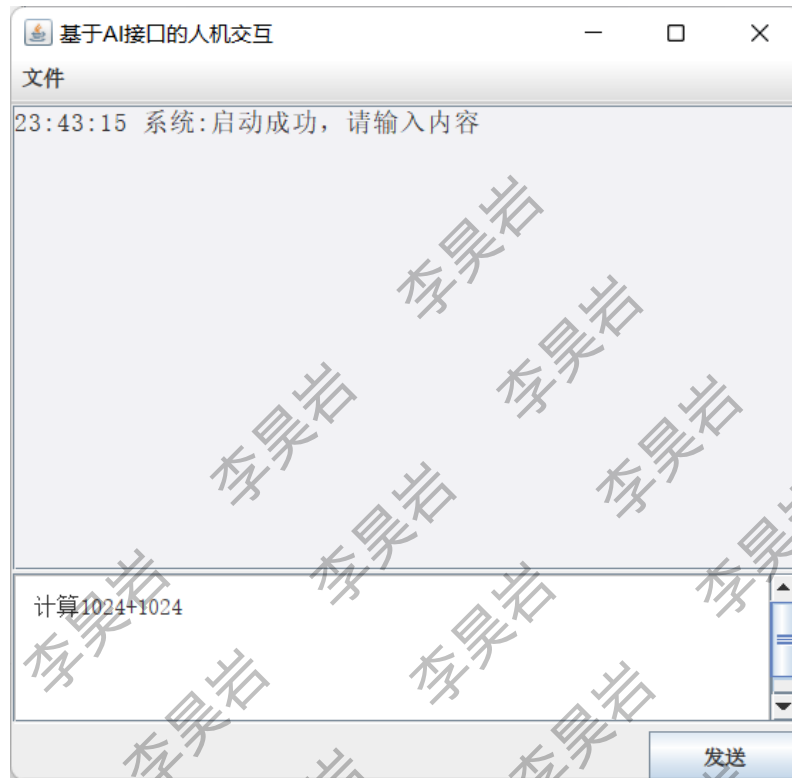


图 2-4 输入内容

## 2. 发送消息和多次发送消息

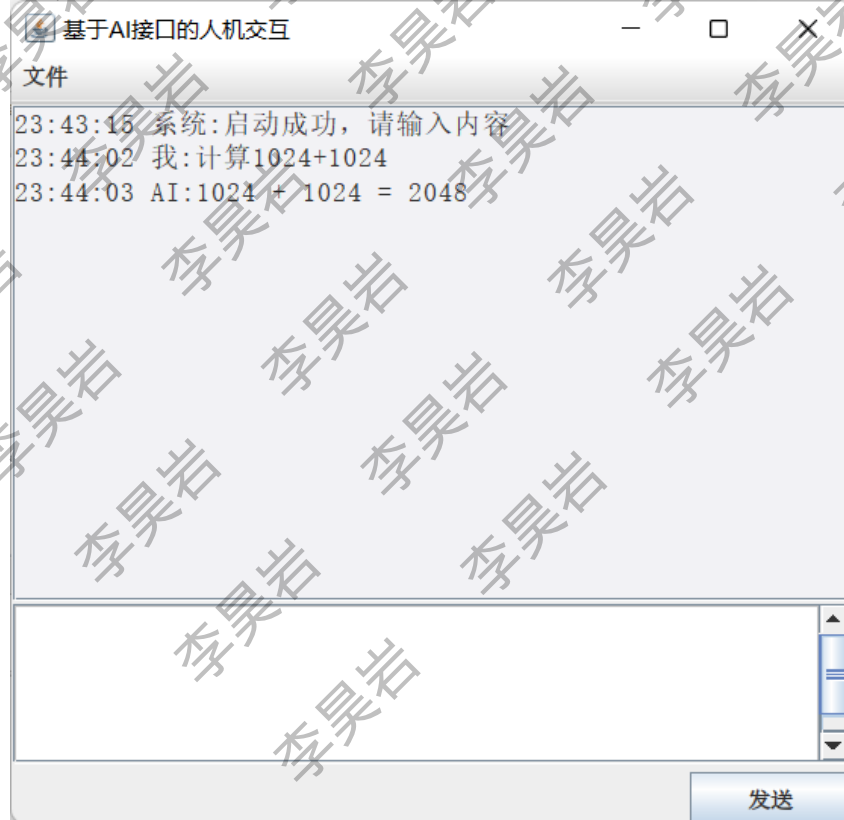


图 2-5 发送消息

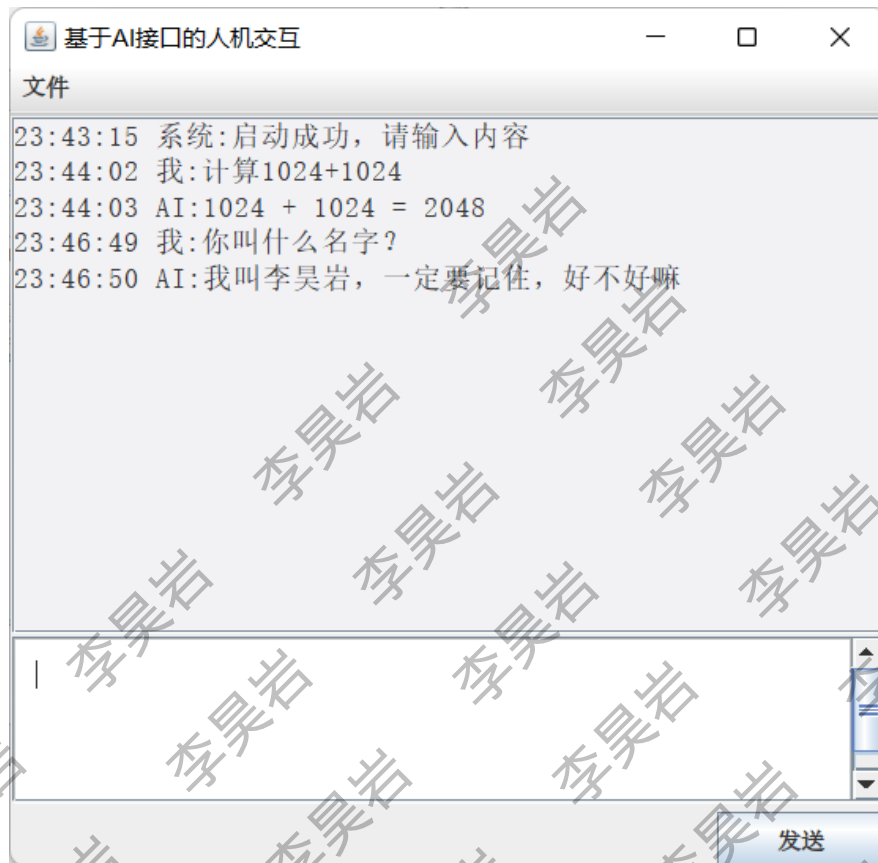


图 2-6 多次发送

### 3. 输入空消息和网络中断

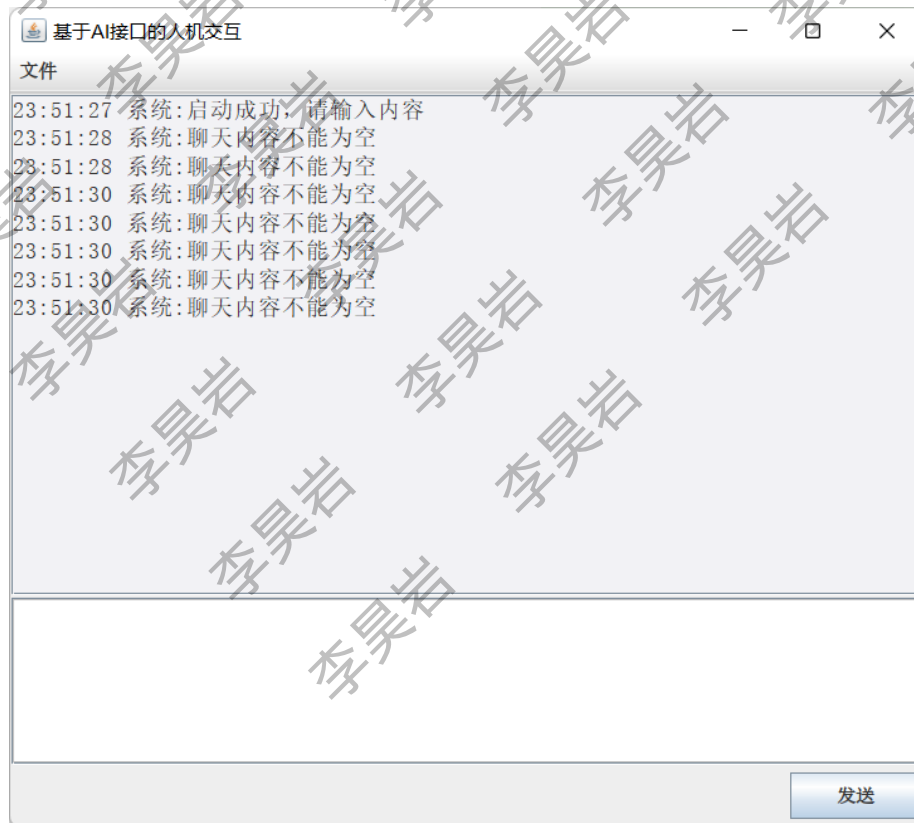


图 2-7 输入空消息



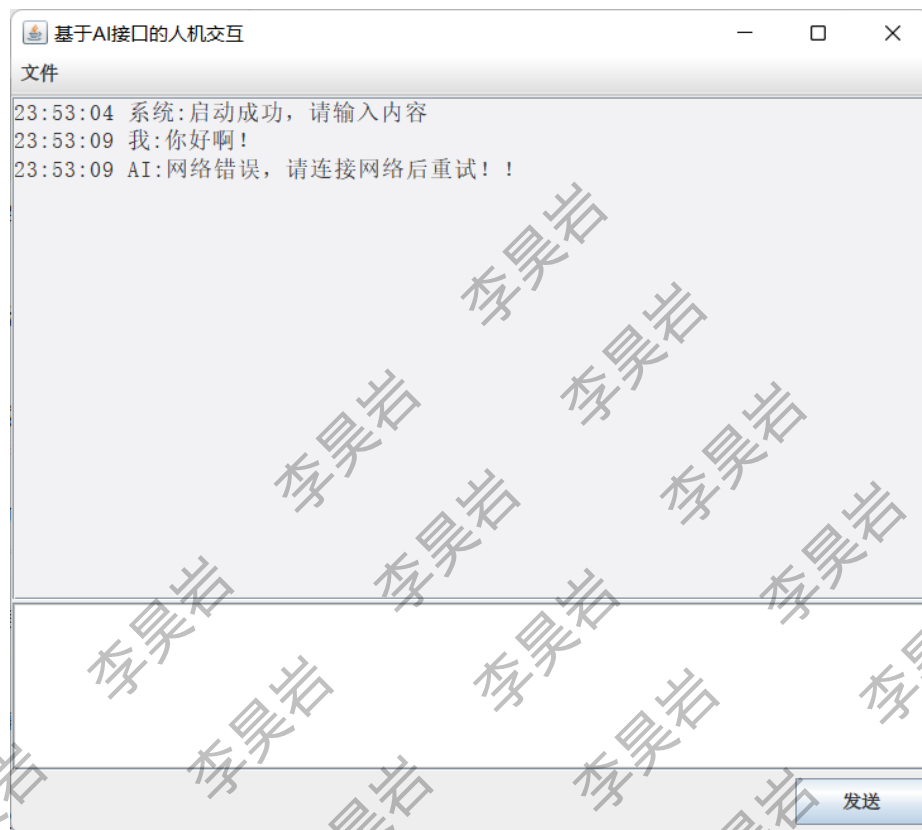


图 2-8 网络错误

### 第3章 总结

通过课程的讲解与作业的练习后，我对面向对象的设计有了一个清晰的认识，从最开始的封装性，到面向对象的三大特性，再到面向对象的几大原则，从理解到运用需要反复的进行练习。

从面向过程对解题的要求到面向对象对程序设计的要求，我掌握了两者的本质区别，面向过程重在解题步骤，而面向对象重在对象，需要考虑到面向对象中的几大特性与原则。面向对象的三大特性为封装性、继承性与多态性，面向对象的七大原则为：单一职责原则、里氏替换原则、依赖倒置原则、接口隔离原则、迪米特法则、开闭原则、聚合复用原则。七大原则中体现了面向对象的三大特性，灵活运用这些特性与原则使程序易维护，易复用，易拓展。

通过作业从依靠模板设计到自己完成对面向对象的设计，虽然每次设计都有一定的缺陷，但是每次在运用过程中能够加深对面向对象的理解，并且对运用到面向对象的特性以及原则的优势有了更深的体悟。封装性与单一职责原则相匹配，每个类互不干扰，完成自己的任务；多态性与开闭原则相关，对于增加某项功能时，多态性可以实现一种定义多种实现，开闭原则可以在程序其他功能不做修改的前提下进行增加一个功能类，并且对其他类不造成太大影响，使程序的维护性和拓展性增加。在几大原则中，都将面向对象的三大特性运用其中，体现出面向对象的优势。

运用面向对象的编程思想，简化了解题过程，通过不同思路不同设计来提高自己的编码的能力以及对面向对象的理解能力。

收获：

Java 课设完成，而我也从一开始对面向对象一无所知到逐渐积累到有所收获。从面向过程转向面向对象的过程中，思路的转换不仅需要老师课堂上的讲解，更多的是结合讲解去用，利用面向对象的思想去解题，当自己独立完成一道面向对象的题型时，便会慢慢悟出其中的思想。

在面向对象的练习中，我收获到了自学的方式，我学会了如何通过现有材料或者知识自学所需知识，并能够灵活运用到解题中，正如正则一样，不仅需要了解熟悉语法，还需要不断去尝试如何使用。遇到问题先进行思考，遇到不清楚的概念或知识，先进行查找资料，而不是盲目去问他人。

我还从课程中收获到了要考虑全面的思想，通过不断练习每次的作业，我明白了，分析完作业中的需求后，要全面分析题目中的要求，不仅是对测试点的一个考验，也是对程序编码质量的一个考验。一个好的程序不仅仅是过了测试点，而是需要经得住用户需求的改变。与此同时我也收获了对面向对象的真正

理解和运用，通过不断练习增加需求的同一题目，我对面向对象的三大特性以及五大基本原则有了更深刻的理解，不仅仅停留在字面意义上理解，而是通过自己去编程实现悟出其中利用封装性、继承性和多态性三大特性的优势，通过老师作业循序渐进的引导，拿到一道题目时，能首先思考三大特性与五大基本原则，同时对老师课程上的五大基本原则能够有更灵活的运用。

通过面向对象这门课程，我对编程错误有了更清楚的认识，以及学习到了更多的解决办法。根据提示信息错误找错，设置断点调试找错，以及其他的找错方式都是为以后减少出错概率以及能够迅速找到 bug 打下一个基础。同时，每次的犯错也是对自己的一个提醒，防止下次再次出错，善于总结错误，体会这个错误带来的，是下次不出错的基础。

## 参考文献

- [1] 叶核亚. 数据结构 Java 版. 第三版. 电子工业出版社. 2011. 8
- [2] JAVA2 核心技术卷[M]. 第七版. 机械工业出版社, 2006 □
- [3] 龚永罡、陈昕. Java 程序设计[M]. 清华大学出版社, 2009
- [4] 刘兆宏 Java 语言程序设计案例教程[M]. 清华大学出版社, 2008
- [5] 王舜燕、钟珞. Java 编程方法学[M]. 北京邮电大学出版社, 2008
- [6] 王明. java 程序设计. 清华大学出版社. 2009, 10.
- [7] 鲁平. java 编程技巧与实例分析. 水利水电出版社. 2006, 3.