

深圳大学课程作业报告

课程名称：Java 程序设计

课程作业名称：结合大模型的 Java 知识问答机器人

学院：计算机与软件学院

专业：数计班

指导教师：潘微科

报告人：詹耿羽 学号：2023193026 班级：数计

作业时间：2024 年 11 月 22 日（周五）~2024 年 12 月 25 日（周三）

课程作业报告提交时间：2024.12.15

教务部制

作业目的与要求:

课程作业目的: Java 程序设计综合练习, 含 GUI、多线程、I/O、网络通信等。

课程作业要求:

(1). 编写 Java 应用程序, 实现“结合大模型的 Java 知识问答机器人”。

- 人机交互界面: 使用命令行窗口(不得分)或图形用户界面(10分)
- 大模型: 允许用户从腾讯混元大模型、百度文心一言大模型、阿里通义千问大模型或其他大模型中(至少支持两个大模型)选择使用一个大模型(20分)
- 多线程&网络通信: 能够并行处理多个用户的网络请求, 每个用户请求都应该在一个新的线程中处理, 以避免阻塞主线程(20分)
- 测试案例: 要求测试 Java 程序设计方面的选择题、判断题、程序填空题和编程题(每个题型至少测试两个例子), 并就测试效果做出简单分析(20分)
- 日志记录: 在本地文本文件(.txt)中记录人机对话过程(20分)
- 特色功能: 可以列出自己实现的其他特色功能(每个特色功能5分, 最多记两个特色功能共10分)

在报告中附上程序截图、完整的运行结果截图和详细的文字说明。

报告写作。要求: 主要思路有明确的说明, 重点代码有详细的注释, 行文逻辑清晰可读性强, 报告整体写作较为专业。(20分)

说明:

(1) 本次课程作业满分为100分, 占总成绩的比例(待定)。

(2) 本次课程作业截至时间 2024 年 12 月 25 日(周**) 21:59。

(3) 报告正文: 请在**指定位置填写**, 本次课程作业**需要单独提交源程序文件**(源程序单独打包在 Blackboard 中上传, 不要包含外部导入的包)。

(4) 个人信息: WORD 文件名中的“姓名”、“学号”, 请改为你的**姓名和学号**; 课程作业报告的首页, 请**准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“课程作业报告提交时间”**等信息。

(5) 提交方式: 截至时间前, 请在 Blackboard 平台中提交。

(6) 发现抄袭(包括复制&粘贴整句话、整张图), **抄袭者和被抄袭者的成绩记零分**。

(7) 延迟提交, 不得分; 如有特殊情况, 请于截至日期之后的**48小时内**发邮件到 panweike@szu.edu.cn, 并在邮件中注明课程名称、作业名称、姓名、学号等信息, 以及特殊情况的说明, 我收到后会及时回复。

(8) 期末考试阶段补交无效。

(1). 编写 Java 应用程序，实现“结合大模型的 Java 知识问答机器人”。

- 人机交互界面：使用命令行窗口（不得分）或图形用户界面（10 分）
- 大模型：允许用户从腾讯混元大模型、百度文心一言大模型、阿里通义千问大模型或其他大模型中（至少支持两个大模型）选择使用一个大模型（20 分）
- 多线程&网络通信：能够并行处理多个用户的网络请求，每个用户请求都应该在一个新的线程中处理，以避免阻塞主线程（20 分）
- 测试案例：要求测试 Java 程序设计方面的选择题、判断题、程序填空题和编程题（每个题型至少测试两个例子），并就测试效果做出简单分析（20 分）
- 日志记录：在本地文本文件（.txt）中记录人机对话过程（20 分）
- 特色功能：可以列出自己实现的其他特色功能（每个特色功能 5 分，最多记两个特色功能共 10 分）

在报告中附上程序截图、完整的运行结果截图和详细的文字说明。

一、产品展示

产品名称：结合大模型的 Java 知识问答机器人

产品介绍：

结合大模型的 Java 知识问答机器人是一款集成人工智能技术与高效学习功能的创新型产品，旨在为开发者、学生以及技术爱好者提供专业、即时且个性化的 Java 知识问答支持。依托阿里通义千问和百度文心一言两大领先语言模型的强大能力，本产品在技术深度和用户体验上实现了全面升级。

核心功能：

- 大模型智能支持：

内置 阿里通义千问 和 百度文心一言 大模型，提供精准的 Java 知识解答，涵盖基础概念、框架应用、代码优化等多层次需求。

自然语言处理能力卓越，支持复杂问题解析和实时解决方案生成。

- 用户专属交互：

提供安全的**用户登录机制（特色功能 1）**：每位用户通过专属用户名和密码登录，确保数据和学习记录的私密性与安全性。

错误登录提示：用户名或密码输入错误时会即时反馈，防止非法访问。

- 个性化学习存档：

每个用户拥有独立的聊天记录存储功能，分开存储，充分保护用户隐私（特色功能 2），所有对话内容以 TXT 格式 保存，可用于后续学习和复盘，打造个性化学习档案。

- 便捷操作体验：

简洁的用户界面设计，**支持登录、使用和退出等操作（特色功能 3）**，用户可随时在菜单中点击退出以结束当前会话。

- 高并发能力：

采用多线程架构，每个用户的请求均在独立线程中处理，有效避免主线程阻塞，确保高效并行响应多用户请求。

- **对话中止功能：（特色功能 4）**

新增对话中止选项：在与大模型进行对话的过程中，用户可随时点击“中止”按钮中断当前对话，及时停止大模型的运行。

不中断会话窗口：中止后，用户的窗口将保持激活状态，可立即开始下一个对话，提供更加灵活的交互体验。

技术优势:

- 高效性: 强大的多线程并行处理架构, 支持同时处理大量用户请求, 响应速度快且稳定。
- 专业性: 大模型精准回答, 助力用户掌握 Java 知识, 从初学者到专家均可受益。
- 易用性: 界面友好, 操作简单, 无需繁琐配置即可享受顶尖 AI 服务。

应用场景:

- 开发者学习辅助: 解决编码疑问, 优化开发流程。
- 教育培训工具: 作为课堂或在线教育平台的知识补充。
- 企业内训支持: 为员工提供即时技术支持和学习资料。

二、问题设计

• 选择题

1. 以下关于 Java 的描述中, 哪一项是正确的?

- A. Java 是一种面向过程的编程语言
- B. Java 代码需要编译成机器码直接运行
- C. Java 支持多继承
- D. Java 使用字节码运行在虚拟机上

答案: D

解析: Java 是一种面向对象的编程语言 (排除 A), 它通过将代码编译为字节码 (.class 文件) 运行在 Java 虚拟机 (JVM) 上, 而不是直接编译为机器码 (排除 B)。Java 不支持类的多继承, 但可以通过接口实现多继承的效果 (排除 C)。D 是正确答案。

2. 以下关于 String 的描述中, 哪一个错误的?

- A. String 是不可变的类
- B. String 可以使用 + 进行连接
- C. String 是 Java 的基本数据类型
- D. String 对象存储在字符串池中

答案: C

解析: String 是一个不可变类 (A 正确), 可以使用 + 操作符进行字符串连接 (B 正确)。String 是 Java 中的引用数据类型, 而非基本数据类型 (C 错误)。字符串对象存储在字符串池中以节约内存 (D 正确)。

• 判断题

1. Java 中, int 和 Integer 是相同的数据类型。

答案: 错误

解析: int 是 Java 的基本数据类型, 而 Integer 是包装类, 是 int 的对象封装类型, 两者虽然可以互相转换, 但并不相同。

2. 在 Java 中, 所有类的父类是 Object。

答案: 正确

解析: Java 中所有类 (包括用户自定义类) 直接或间接继承自 Object 类, 这是 Java 面向对象的核心设计。

• 编程填空题 (答案合理即可)

1. 计算两个数的最大公约数

请完成以下 Java 代码, 计算两个整数的最大公约数 (GCD)。

```
public class GCD {  
    public static void main(String[] args) {
```

```
        int num1 = 56;
        int num2 = 98;
        System.out.println("GCD of " + num1 + " and " + num2 + " is: " +
findGCD(num1, num2));
    }
}
```

```
    public static int findGCD(int a, int b) {
        while (b != 0) {
            // 填空部分
        }
        return a;
    }
}
```

正确答案

```
int temp = b;
```

```
b = a % b;
```

```
a = temp;
```

2.数组反转

请完成以下 Java 代码，反转一个整数数组。

```
public class ReverseArray {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};
        reverse(array);
        for (int i : array) {
            System.out.print(i + " ");
        }
    }
}
```

```
    public static void reverse(int[] arr) {
        int left = 0;
        int right = arr.length - 1;
        while (left < right) {
            // 填空部分
        }
    }
}
```

正确答案:

```
int temp = arr[left];
```

```
arr[left] = arr[right];
```

```
arr[right] = temp;
```

```
left++;
```

```
right--;
```

• 编程题（答案合理即可）

编写一个 Java 程序，计算并输出数组中的最大值。

```

1.  public class MaxFinder {
2.      public static void main(String[] args) {
3.          int[] numbers = {3, 5, 7, 2, 8};
4.          int max = numbers[0];
5.          for (int num : numbers) {
6.              if (num > max) {
7.                  max = num;
8.              }
9.          }
10.         System.out.println("Maximum value: " + max);
11.     }
12. }

```

解析：

定义一个数组 `numbers` 并初始化。

通过 `for-each` 循环遍历数组元素，逐一比较当前元素与变量 `max` 的值，将更大的值赋给 `max`。

最终，`max` 中保存的是数组中的最大值。

2. 编写一个程序，检查输入的字符串是否是回文。

```

1.  import java.util.Scanner;
2.
3.  public class PalindromeChecker {
4.      public static void main(String[] args) {
5.          Scanner scanner = new Scanner(System.in);
6.          System.out.print("Enter a string: ");
7.          String input = scanner.nextLine();
8.          String reversed = new StringBuilder(input).reverse().toString();
9.          if (input.equals(reversed)) {
10.             System.out.println("The string is a palindrome.");
11.         } else {
12.             System.out.println("The string is not a palindrome.");
13.         }
14.         scanner.close();
15.     }
16. }

```

解析：

使用 `Scanner` 获取用户输入字符串。

通过 `StringBuilder` 的 `reverse` 方法获取反转后的字符串并与原字符串进行比较。

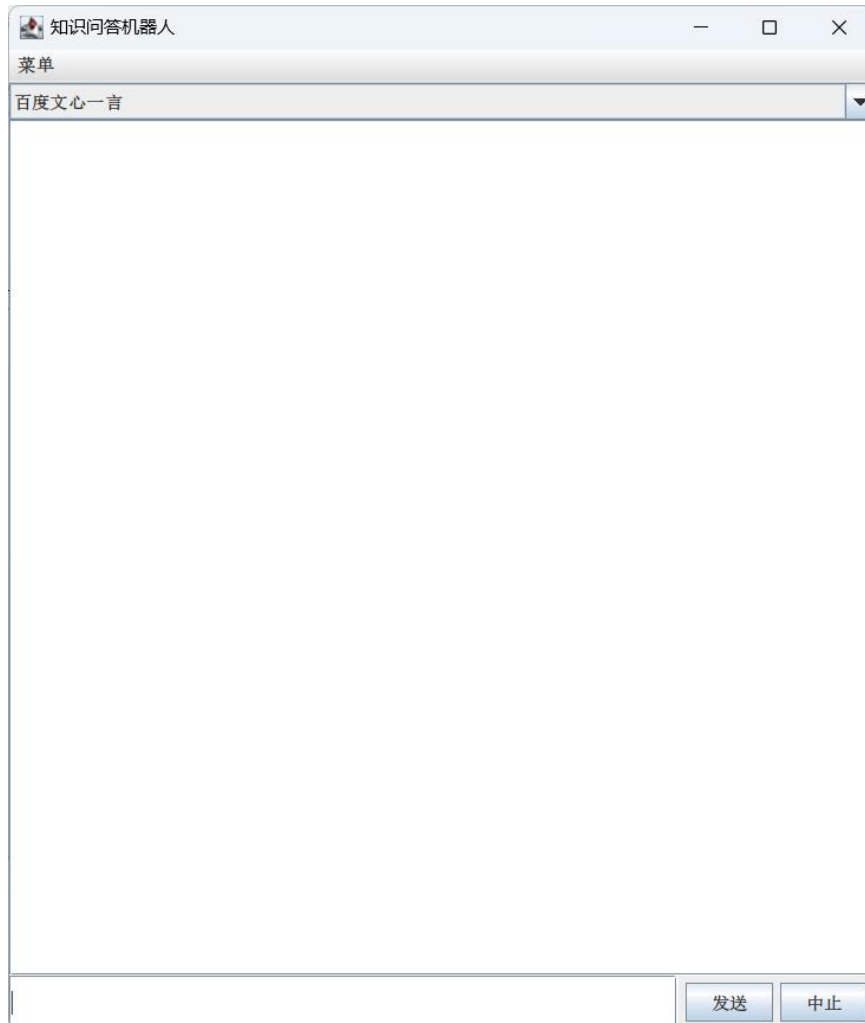
如果两者相等，则为回文字符串；否则，不是回文。

三、使用说明

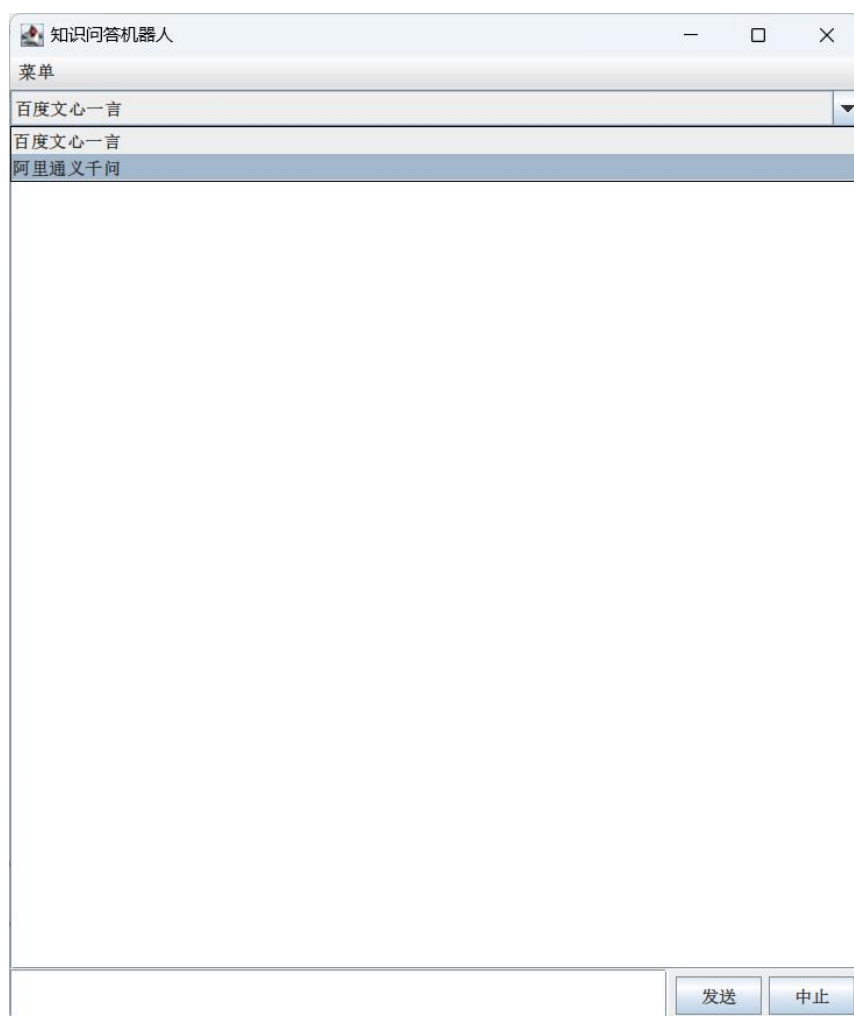
1.运行代码，进入登录界面：



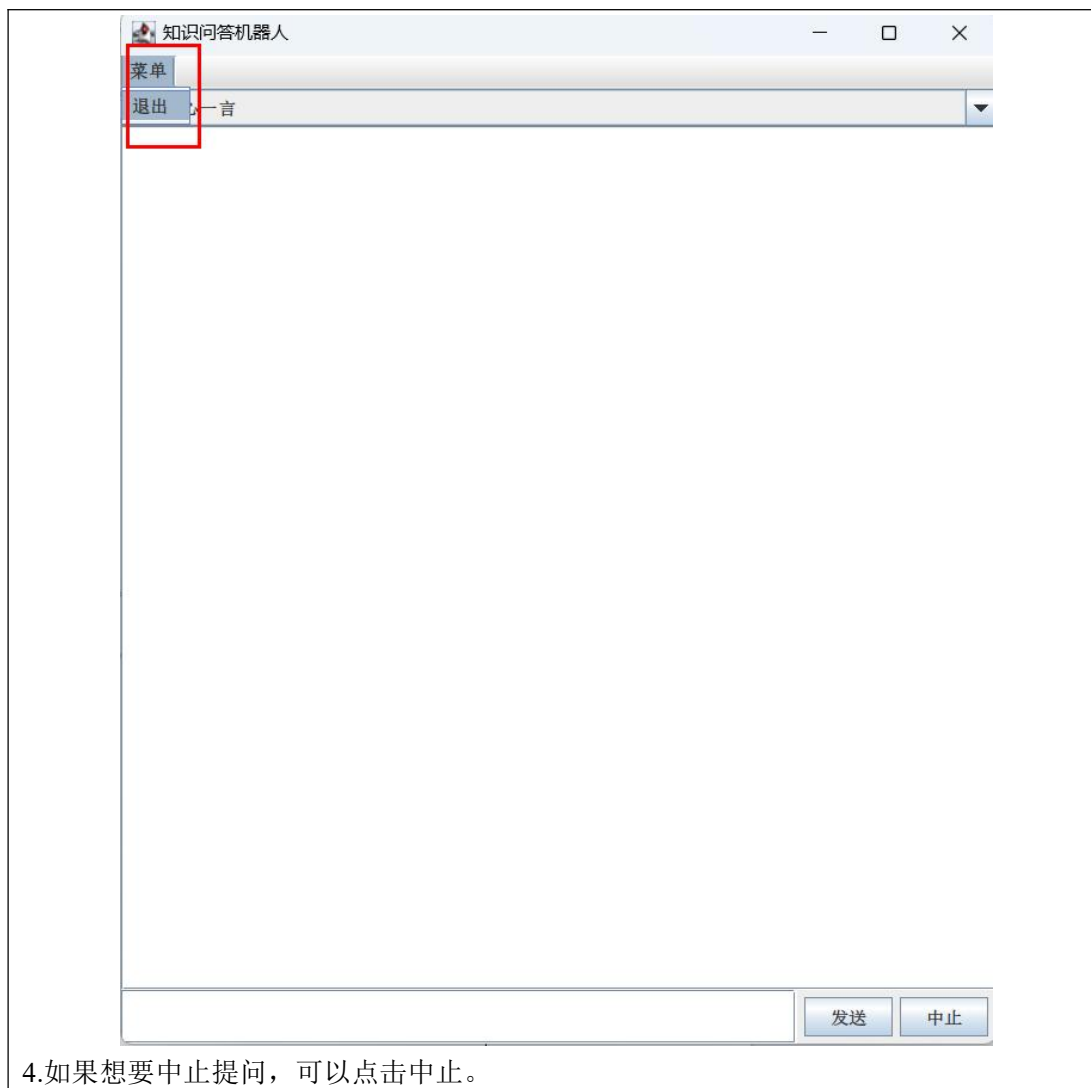
2.进入使用界面，可以在聊天框提问问题：



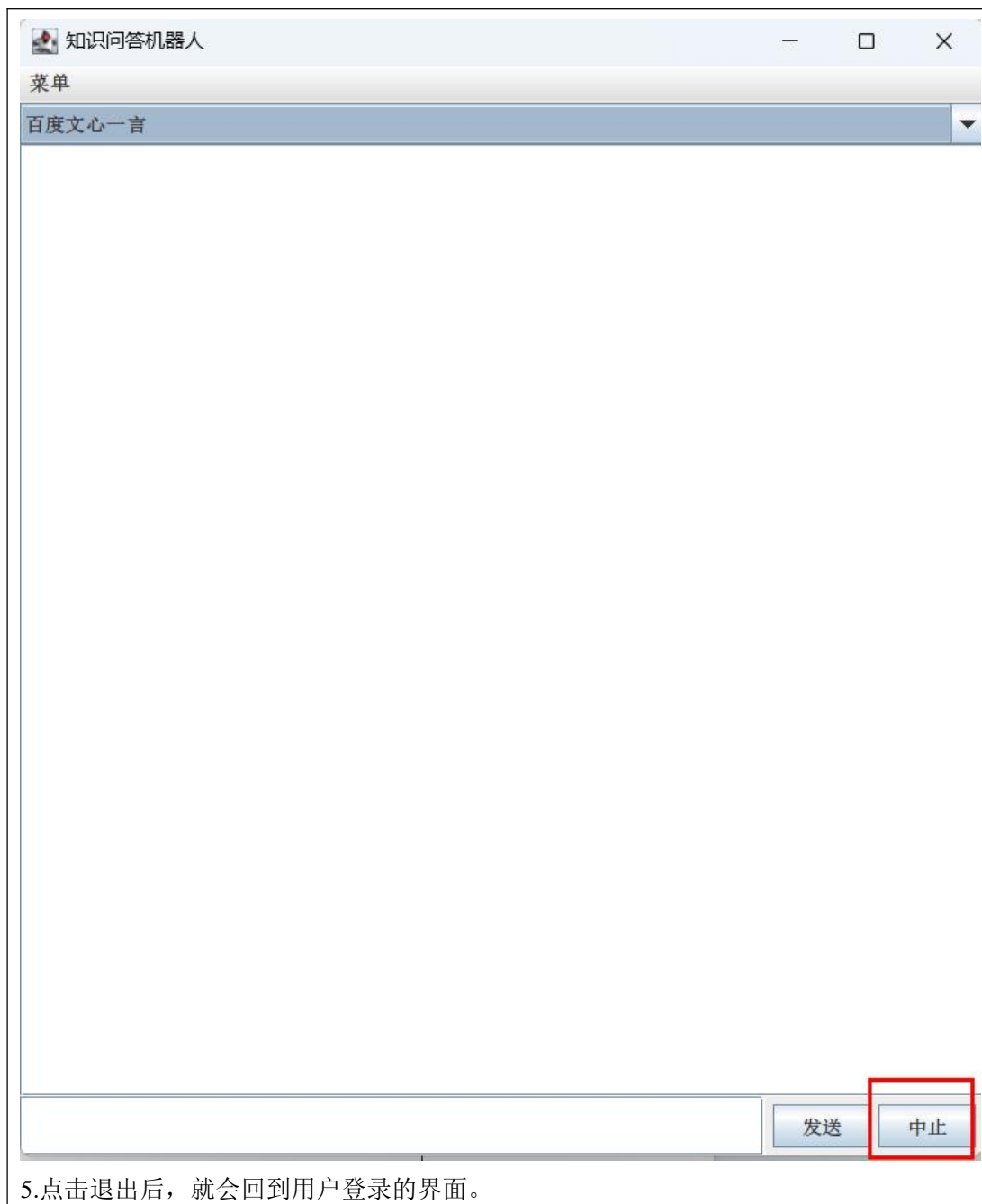
3.进入界面后，可以更换大模型。



4.点击菜单，可以选择退出。



4.如果想要中止提问，可以点击中止。



5.点击退出后，就会回到用户登录的界面。

 问答机器人

用户名

1

密码

•

登录

重置

Me: 选择题 1.以下关于 Java 的描述中，哪一项是正确的？ A. Java 是一种面向过程的编程语言 B. Java 代码需要编译成机器码直接运行 C. Java 支持多继承 D. Java 使用字节码运行在虚拟机上； 2.以下关于 String 的描述中，哪一个错误的？ A. String 是不可变的类 B. String 可以使用 + 进行连接 C. String 是 Java 的基本数据类型 D. String 对象存储在字符串池中；做一下上面两道Java知识选择题

机器人: 1. 以下关于 Java 的描述中，哪一项是正确的？

答案：D. Java 使用字节码运行在虚拟机上。

解释：

A. Java 是一种面向对象的编程语言，而不是面向过程的。

B. Java 代码需要编译成字节码，然后由 Java 虚拟机（JVM）解释或编译成机器码运行，而不是直接运行机器码。

C. Java 不支持多继承，但支持接口的多实现。

D. 正确，Java 的字节码是在 JVM 上运行的。

2. 以下关于 String 的描述中，哪一个错误的？

答案：C. String 是 Java 的基本数据类型。

解释：

A. String 是不可变的类，这意味着一旦一个字符串对象被创建并初始化，它的内容就不能被改变。

B. String 确实可以使用 + 进行连接来创建新的字符串。

C. 错误，String 不是 Java 的基本数据类型。Java 的基本数据类型包括 int、float、char 等。String 是一个类，用于表示和操作字符串。

D. String 对象通常存储在字符串池中，这样可以有效地重复使用相同的字符串对象。

因此，两个问题的正确答案分别是 D 和 C。

发送

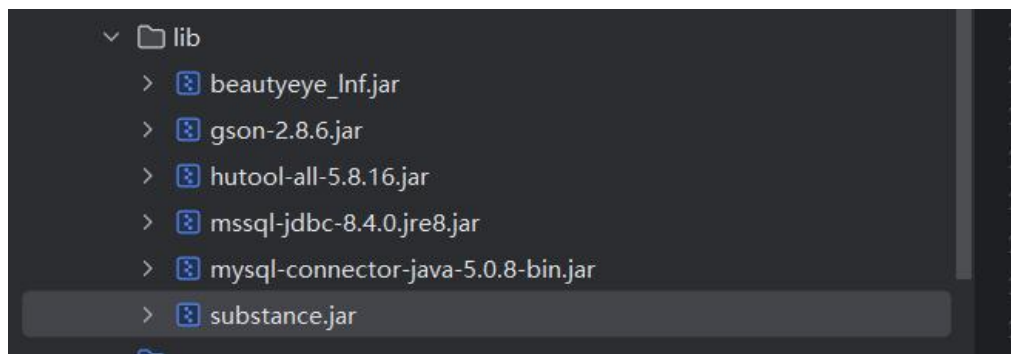
中止



四、代码解析

1. 环境配置

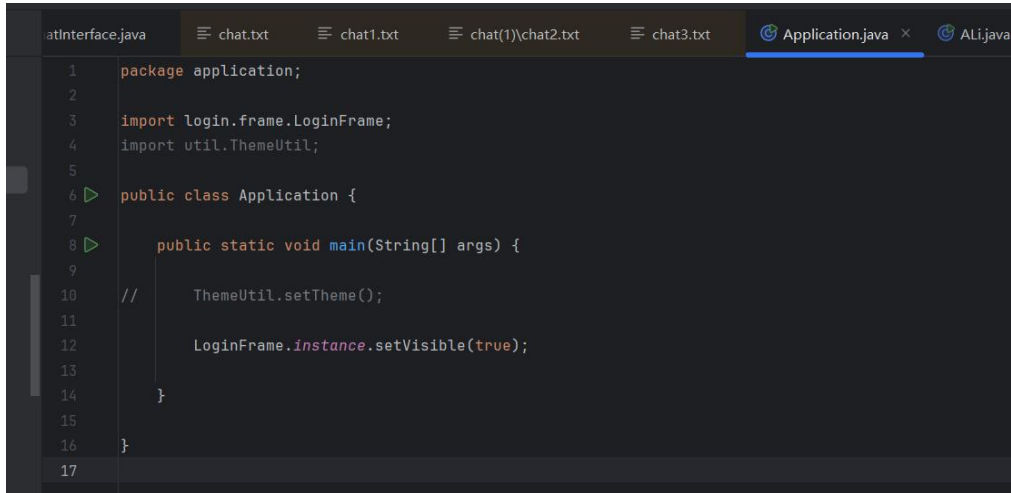
在 lib 的目录下，把下面的 jar 包导入到项目中：



2. 代码剖析

1) Application.java

- 程序截图



```
1 package application;
2
3 import login.frame.LoginFrame;
4 import util.ThemeUtil;
5
6 public class Application {
7
8     public static void main(String[] args) {
9
10         // ThemeUtil.setTheme();
11
12         LoginFrame.instance.setVisible(true);
13     }
14 }
15
16 }
17
```

- 文字说明

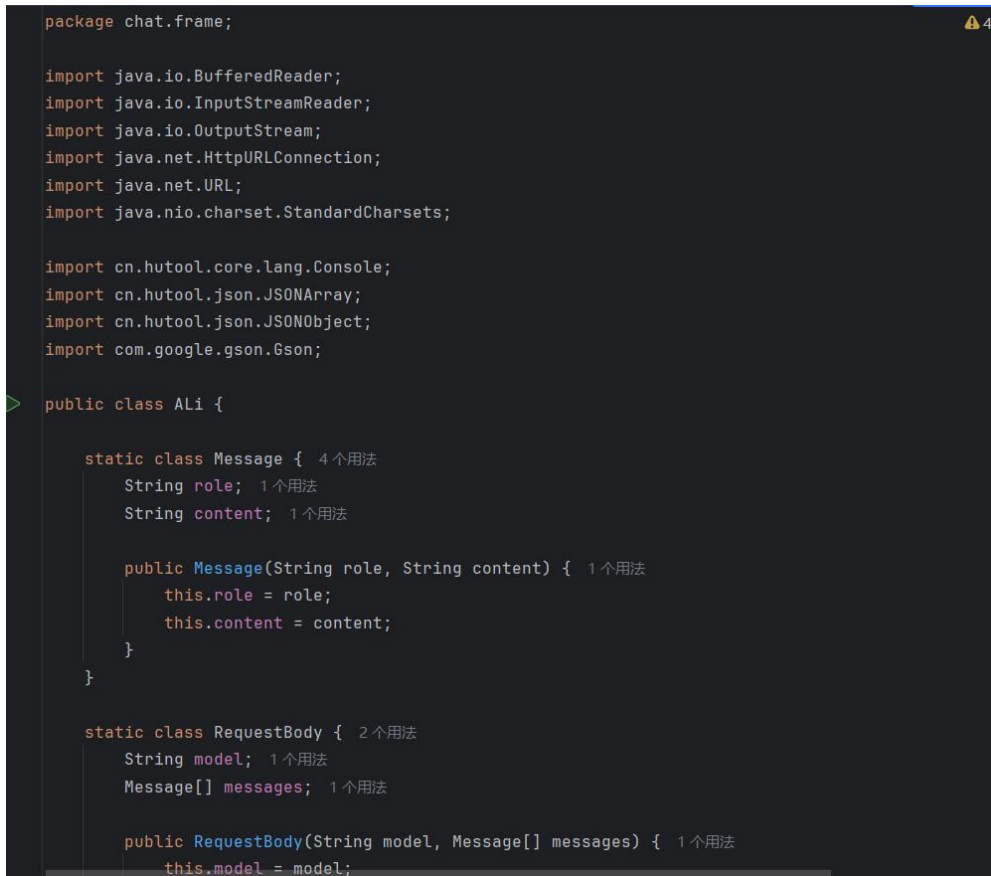
功能：

作为程序的入口，初始化用户界面。

通过 LoginFrame 实例调用登录窗口。

2) ALi.java

- 程序截图



```
package chat.frame;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;

import cn.hutool.core.lang.Console;
import cn.hutool.json.JSONArray;
import cn.hutool.json.JSONObject;
import com.google.gson.Gson;

public class ALi {

    static class Message { 4个用法
        String role; 1个用法
        String content; 1个用法

        public Message(String role, String content) { 1个用法
            this.role = role;
            this.content = content;
        }
    }

    static class RequestBody { 2个用法
        String model; 1个用法
        Message[] messages; 1个用法

        public RequestBody(String model, Message[] messages) { 1个用法
            this.model = model;
        }
    }
}
```

• 完整代码

```
1. package chat.frame;
2.
3. import java.io.BufferedReader;
4. import java.io.InputStreamReader;
5. import java.io.OutputStream;
6. import java.net.HttpURLConnection;
7. import java.net.URL;
8. import java.nio.charset.StandardCharsets;
9.
10. import cn.hutool.core.lang.Console;
11. import cn.hutool.json.JSONArray;
12. import cn.hutool.json.JSONObject;
13. import com.google.gson.Gson;
14.
15. public class ALi {
16.
17.     static class Message {
18.         String role;
19.         String content;
20.
21.         public Message(String role, String content) {
22.             this.role = role;
23.             this.content = content;
24.         }
25.     }
26.
27.     static class RequestBody {
28.         String model;
29.         Message[] messages;
30.
31.         public RequestBody(String model, Message[] messages) {
32.             this.model = model;
33.             this.messages = messages;
34.         }
35.     }
36.
37.     public static String query(String answer){
38.         try {
39.             // 2222222222
40.             RequestBody requestBody = new RequestBody(
41.                 "qwen-plus",
42.                 new Message[] {
43.                     new Message("user", answer)
```

```

44.         }
45.     );
46.
47.     // 创建JSON
48.     Gson gson = new Gson();
49.     String jsonString = gson.toJson(requestBody);
50.     Console.log("jsonInputString : {}", jsonString);
51.
52.     // 创建URL
53.     URL url = new URL("https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions");
54.     HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
55.
56.     // 设置POST
57.     httpURLConnection.setRequestMethod("POST");
58.     httpURLConnection.setRequestProperty("Content-Type", "application/json; utf-8");
59.     httpURLConnection.setRequestProperty("Accept", "application/json");
60.
61.     // 设置API Key
62.     String apiKey = "sk-d335350e318d4b5ea0989e26602f8c8b";
63.     String auth = "Bearer " + apiKey;
64.     httpURLConnection.setRequestProperty("Authorization", auth);
65.
66.     // 设置输出
67.     httpURLConnection.setDoOutput(true);
68.
69.     // 尝试输出
70.     try (OutputStream os = httpURLConnection.getOutputStream()) {
71.         byte[] input = jsonString.getBytes(StandardCharsets.UTF_8);
72.         os.write(input, 0, input.length);
73.     }
74.
75.     // 获取响应码
76.     int responseCode = httpURLConnection.getResponseCode();

```



```

77.         System.out.println("Response Code: " + responseCode
78.     );
79.         // 乱码
80.         try (BufferedReader br = new BufferedReader(new Inp
81. utStreamReader(httpURLConnection.getInputStream(), StandardChars
82. ets.UTF_8))) {
83.             StringBuilder response = new StringBuilder();
84.             String responseLine;
85.             while ((responseLine = br.readLine()) != null)
86.             {
87.                 response.append(responseLine.trim());
88.             }
89.             System.out.println("Response Body: " + response
90.     );
91.             JSONObject jo = new JSONObject(response);
92.             JSONArray ja = jo.getJSONArray("choices");
93.             JSONObject oo = ja.getJSONObject(0);
94.             JSONObject mm = oo.getJSONObject("message");
95.             String cc = mm.getStr("content");
96.             Console.log("cc : {}", cc);
97.             return cc;
98.         }
99.     } catch (Exception e) {
100.         e.printStackTrace();
101.     } finally {
102.         // System.exit(0);
103.     }
104.     return "";
105. }
106.
107. public static void main(String[] args) {
108.     String answer = ALi.query("北京市天气如何");
109.     Console.log("answer : {}", answer);
110. }
111. }

```

• 文字说明

功能：

实现与百度文心一言模型的通信。

构造请求并解析响应。

内部类 **Message**：封装了一条对话消息，role 表示角色（如 user 或 assistant），content 是具体的对话内容。

内部类 **RequestBody**：封装了完整的请求体，包括：模型名称（model，如 qwen-plus）。

消息数组（messages），存储多条消息，支持上下文对话。

请求流程：

创建 URL：使用阿里大模型的 RESTful 接口地址。

设置请求方式和头部：POST 请求。

Authorization 头部中设置了 Bearer 令牌（API Key）。

Content-Type 指定为 application/json。

发送请求体：使用 Gson 库将 RequestBody 对象序列化为 JSON。

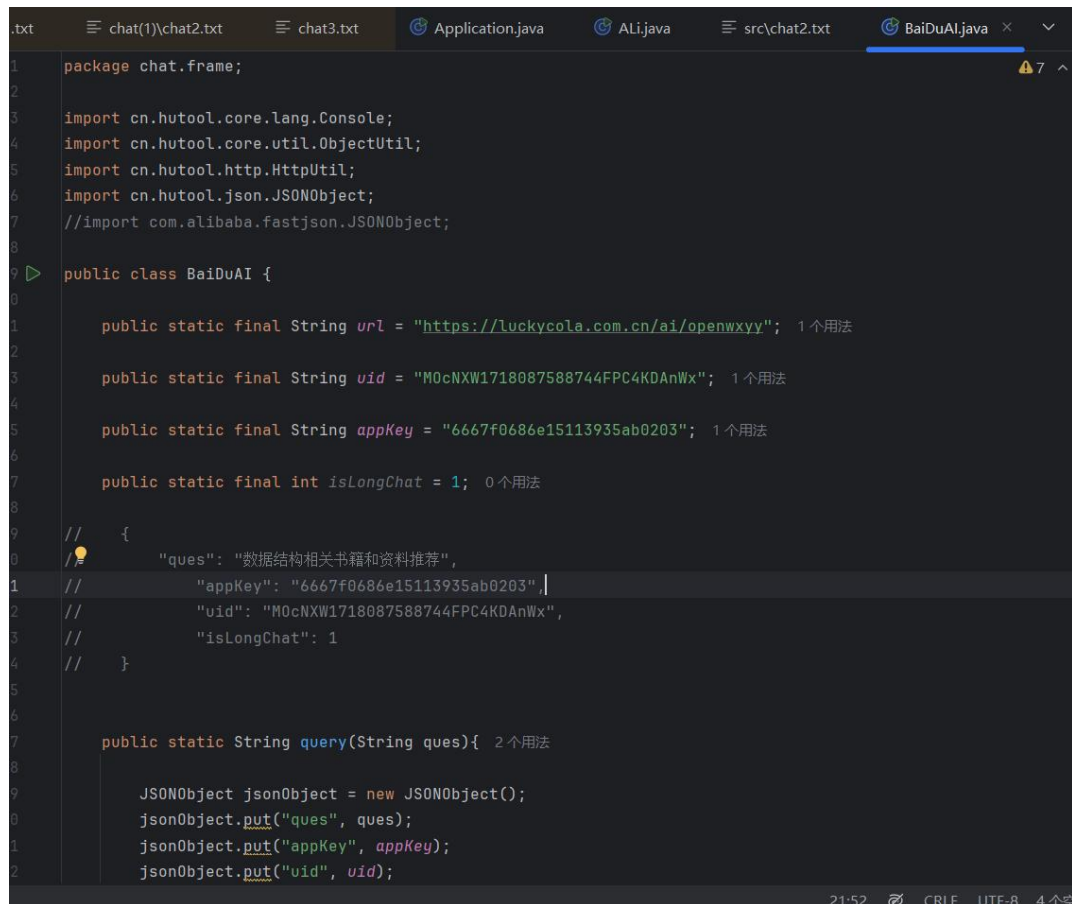
通过 `URLConnection.getOutputStream()` 将 JSON 写入请求体。

解析响应：读取服务器返回的 JSON 响应，提取回答内容，并返回给调用者。

本段代码实现了阿里通义千问 API 的完整调用，包括封装请求体、发送 HTTP 请求和解析响应。

3) BaiDuAI.java

- 程序截图



```
1 package chat.frame;
2
3 import cn.hutool.core.lang.Console;
4 import cn.hutool.core.util.ObjectUtil;
5 import cn.hutool.http.HttpUtil;
6 import cn.hutool.json.JSONObject;
7 //import com.alibaba.fastjson.JSONObject;
8
9 public class BaiDuAI {
10
11     public static final String url = "https://luckycola.com.cn/ai/openwxyy"; 1个用法
12
13     public static final String uid = "M0cNXW1718087588744FPC4KdAnWx"; 1个用法
14
15     public static final String appKey = "6667f0686e15113935ab0203"; 1个用法
16
17     public static final int isLongChat = 1; 0个用法
18
19     // {
20     //     "ques": "数据结构相关书籍和资料推荐",
21     //     "appKey": "6667f0686e15113935ab0203",
22     //     "uid": "M0cNXW1718087588744FPC4KdAnWx",
23     //     "isLongChat": 1
24     // }
25
26     public static String query(String ques){ 2个用法
27
28         JSONObject jsonObject = new JSONObject();
29         jsonObject.put("ques", ques);
30         jsonObject.put("appKey", appKey);
31         jsonObject.put("uid", uid);
32     }
```

- 完整代码

```
1. package chat.frame;
2.
3. import cn.hutool.core.lang.Console;
4. import cn.hutool.core.util.ObjectUtil;
5. import cn.hutool.http.HttpUtil;
```

```
6. import cn.hutool.json.JSONObject;
7. //import com.alibaba.fastjson.JSONObject;
8.
9. public class BaiDuAI {
10.
11.     public static final String url = "https://luckycola.com.cn/
    ai/openwxyy";
12.
13.     public static final String uid = "MOcNXW1718087588744FPC4KD
    AnWx";
14.
15.     public static final String appKey = "6667f0686e15113935ab02
    03";
16.
17.     public static final int isLongChat = 1;
18.
19.     // {
20.     //     "ques": "数据结构相关书籍和资料推荐",
21.     //     "appKey": "6667f0686e15113935ab0203",
22.     //     "uid": "MOcNXW1718087588744FPC4KDAnWx",
23.     //     "isLongChat": 1
24.     // }
25.
26.
27.     public static String query(String ques){
28.
29.         JSONObject jsonObject = new JSONObject();
30.         jsonObject.put("ques", ques);
31.         jsonObject.put("appKey", appKey);
32.         jsonObject.put("uid", uid);
33.         jsonObject.put("isLongChat", 1);
34.
35.         String body = HttpUtil.post(url, jsonObject.toString())
            ;
36.         Console.log("body : {}", body);
37.         JSONObject jo = new JSONObject(body);
38.         Integer i = jo.getInt("code");
39.         if(ObjectUtil.equals(i, 0)){
40.             return jo.getJSONObject("data").getStr("result");
41.         }else{
42.             return "暂时无法解答，你可以问些其他问题！";
43.         }
44.     }
45.
```

```

46.     public static void main (String args[]){
47.         query("你是谁呀");
48.     }
49.
50. }

```

- 文字说明

功能：

封装百度文心一言（Wenxin Yiyan）大模型的 API 调用，向其发送问题并获取回答。

内容：

请求参数封装：构建 JSON 请求对象，包含：ques：用户提问内容。

appKey 和 uid：模型的用户标识和授权凭据。

isLongChat：是否开启长对话模式。

发送请求：使用 HttpUtil.post() 发送 HTTP POST 请求，将 JSON 数据提交到指定 URL。

解析响应：将服务器返回的 JSON 响应解析为 Java 对象。

提取回答内容，如果响应码不是 0，返回默认提示。

4) ChatInterface.java

- 程序截图

```

1  package chat.frame;
2
3  import cn.hutool.core.io.FileUtil;
4  import cn.hutool.core.util.StrUtil;
5  //import com.sun.deploy.panel.JPanel;
6  import login.frame.LoginFrame;
7  import login.panel.LoginPanel;
8
9  import javax.swing.*;
10 import java.awt.*;
11 import java.awt.event.ActionEvent;
12 import java.awt.event.ActionListener;
13 import java.nio.charset.Charset;
14
15 public class ChatInterface extends JFrame {
16
17     public static String path = "C:\\Users\\詹歌羽\\Desktop\\chat(1)\\chat\\src\\"; 1 个用法
18     public static String path2 = "C:\\Users\\詹歌羽\\Desktop\\chat(1)\\chat\\src\\chat2.txt"; 0 个用法
19
20     private JTextArea chatArea; 8 个用法
21     private JTextField inputField; 6 个用法
22     private JButton sendButton; 3 个用法
23     private JButton zhongZhi; 3 个用法
24     private JComboBox<String> comboBox; 3 个用法
25
26     private static volatile boolean zz = false; 3 个用法
27
28
29     public ChatInterface() { createUI(); }
30
31
32
33     private void createUI() { 1 个用法

```

- 完整代码

```

1.     package chat.frame;
2.
3.     import cn.hutool.core.io.FileUtil;

```

```
4. import cn.hutool.core.util.StrUtil;
5. //import com.sun.deploy.panel.JPanel;
6. import login.frame.LoginFrame;
7. import login.panel.LoginPanel;
8.
9. import javax.swing.*;
10. import java.awt.*;
11. import java.awt.event.ActionEvent;
12. import java.awt.event.ActionListener;
13. import java.nio.charset.Charset;
14.
15. public class ChatInterface extends JFrame {
16.
17.     public static String path = "C:\\Users\\詹耿羽
        \\Desktop\\chat(1)\\chat\\src\\";
18.     public static String path2 = "C:\\Users\\詹耿羽
        \\Desktop\\chat(1)\\chat\\src\\chat2.txt";
19.
20.     private JTextArea chatArea;
21.     private JTextField inputField;
22.     private JButton sendButton;
23.     private JButton zhongZhi;
24.     private JComboBox<String> comboBox;
25.
26.     private static volatile boolean zz = false;
27.
28.
29.     public ChatInterface() {
30.         createUI();
31.     }
32.
33.     private void createUI() {
34.         //
35.         setTitle("知识问答机器人");
36.         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
37.         setSize(600, 700);
38.         setLocationRelativeTo(null); // 居中
39.
40.         //
41.         JPanel panel = new JPanel(new BorderLayout());
42.
43.         //
44.         chatArea = new JTextArea(12, 30);
45.         chatArea.setLineWrap(true);
```

```

46. chatArea.setWrapStyleWord(true);
47. chatArea.setEditable(false); //
48.
49. String[] items = {"百度文心一言", "阿里通义千问"};
50. comboBox = new JComboBox<>(items);
51.
52. panel.add(comboBox, BorderLayout.NORTH);
53.
54. //
55. JScrollPane scrollPane = new JScrollPane(chatArea);
56. panel.add(scrollPane, BorderLayout.CENTER);
57.
58. //
59. inputField = new JTextField(30);
60.
61. JPanel p = new JPanel();
62. p.setLayout(new BorderLayout());
63. p.add(inputField, BorderLayout.CENTER);
64.
65.
66. panel.add(p, BorderLayout.SOUTH);
67.
68. JPanel p2 = new JPanel();
69.
70. zhongZhi = new JButton("中止");
71. zhongZhi.addActionListener(new ActionListener() {
72.     @Override
73.     public void actionPerformed(ActionEvent e) {
74.         ChatInterface.zz = true;
75.     }
76. });
77.
78. //
79. sendButton = new JButton("发送");
80. sendButton.addActionListener(new ActionListener() {
81.     public void actionPerformed(ActionEvent e) {
82.         sendMessage();
83.     }
84. });
85.
86. p2.add(sendButton);
87. p2.add(zhongZhi);
88. //

```

```

89.         p.add(p2, BorderLayout.EAST);
90.         // 设置窗口标题
91.         add(panel);
92.
93.
94.
95.
96.         JMenuBar mb = new JMenuBar();
97.
98.         //
99.         JMenu heSuan = new JMenu("菜单");
100.
101.         JMenuItem jMenuItem = new JMenuItem("退出");
102.         heSuan.add(jMenuItem);
103.
104.         mb.add(heSuan);
105.         setJMenuBar(mb);
106.
107.         ChatInterface c = this;
108.
109.         jMenuItem.addActionListener(new ActionListener() {
110.             @Override
111.             public void actionPerformed(ActionEvent e) {
112.
113.                 c.setVisible(false);
114.                 LoginFrame.instance.setVisible(true);
115.             }
116.
117.         });
118.
119.         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
120.
121.         // 设置窗口大小
122.         // setVisible(true);
123.     }
124.
125.     // 发送消息
126.     private void sendMessage() {
127.
128.         String p = path;
129.         String fileName = "chat{}.txt";
130.         fileName = StrUtil.format(fileName, LoginPanel.userId);
131.         p += fileName;
132.

```

```

133.         final String pp = p;
134.         String s = (String)comboBox.getSelectedItem();
135.
136.         String message = inputField.getText();
137.
138.         String wen = "";
139.
140.         if (!message.trim().isEmpty()) {
141.             wen = "Me: " + message + "\n";
142.             chatArea.append(wen);
143.             inputField.setText(""); // ?????????
144.         }
145.
146.         FileUtil.appendString(wen, pp, Charset.defaultCharset()
147.             );
148.         new Thread(new Runnable() {
149.             @Override
150.             public void run() {
151.                 String huiDa= "";
152.                 if(StrUtil.equals(s, "阿里通义千问")){
153.                     huiDa = ALi.query(message);
154.                 }else{
155.                     huiDa = BaiDuAI.query(message);
156.                 }
157.
158.                 String da = "";
159. //         String huiDa = BaiDuAI.query(message);
160.                 if (!huiDa.trim().isEmpty()) {
161.                     da = "机器人: " + huiDa + "\n";
162.                     inputField.setText("");
163.                     for(int i=0; i<da.length(); i++){
164.                         try{
165.                             Thread.sleep(20);
166.                         }catch (Exception e){
167.                             e.printStackTrace();
168.                         }
169.                         if(ChatInterface.zz){
170.                             ChatInterface.zz = false;
171.                             chatArea.append("\n");
172.                             return;
173.                         }
174.                         chatArea.append(da.charAt(i)+"");
175.                         inputField.repaint();

```



```

176.         }
177.     }
178.
179.         FileUtils.appendString(da, pp, Charset.defaultCh
        arset());
180.     }
181. }).start();
182.
183. }
184.
185. public static void main(String[] args) {
186.     SwingUtilities.invokeLater(new Runnable() {
187.         public void run() {
188.             new ChatInterface();
189.         }
190.     });
191. }
192. }

```

- 文字说明

功能：

实现用户与知识问答机器人交互的图形界面，包括选择模型、发送提问、查看回答等功能。

代码分析：

界面布局：

标题：通过 setTitle("知识问答机器人") 设置窗口标题。

聊天显示区域：使用 JTextArea 实现，显示用户与机器人的对话内容。

用户输入区域：使用 JTextField 实现，允许用户输入提问。

模型选择：通过 JComboBox 提供多个模型选项（如阿里、百度）。

发送按钮：绑定事件监听器，处理用户提问。

功能实现：

```

new Thread(new Runnable() {
    @Override
    public void run() {
        String huiDa = "";
        if (StrUtil.equals(s, "阿里通义千问")) {
            huiDa = ALi.query(message);
        } else {
            huiDa = BaiDuAI.query(message);
        }
        chatArea.append("Bot: " + huiDa + "\n");
    }
}).start();

```

使用新线程处理提问，避免阻塞主线程。

根据用户选择调用对应大模型接口（ALi.query 或 BaiDuAI.query）。

将返回结果显示在聊天区域。

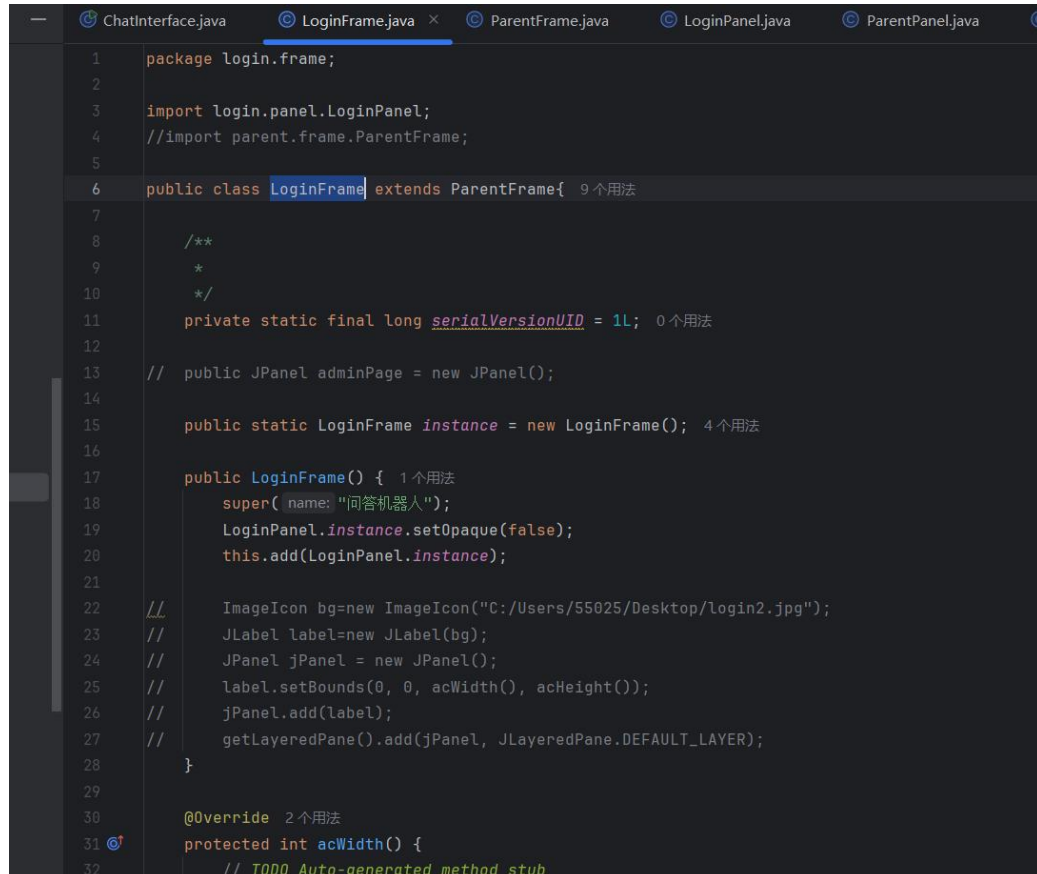
日志记录：

FileUtil.appendString(wen, pp, Charset.defaultCharset());

将用户与机器人的对话追加到本地文本文件，便于后续分析。

5) LoginFrame 类

- 程序截图



```
1 package login.frame;
2
3 import login.panel.LoginPanel;
4 //import parent.frame.ParentFrame;
5
6 public class LoginFrame extends ParentFrame{ 9个用法
7
8     /**
9     *
10    */
11    private static final long serialVersionUID = 1L; 0个用法
12
13    // public JPanel adminPage = new JPanel();
14
15    public static LoginFrame instance = new LoginFrame(); 4个用法
16
17    public LoginFrame() { 1个用法
18        super( name: "问答机器人");
19        LoginPanel.instance.setOpaque(false);
20        this.add(LoginPanel.instance);
21
22    // ImageIcon bg=new ImageIcon("C:/Users/55025/Desktop/login2.jpg");
23    // JLabel label=new JLabel(bg);
24    // JPanel jPanel = new JPanel();
25    // label.setBounds(0, 0, acWidth(), acHeight());
26    // jPanel.add(label);
27    // getLayeredPane().add(jPanel, JLayeredPane.DEFAULT_LAYER);
28    }
29
30    @Override 2个用法
31    protected int acWidth() {
32        // TODO Auto-generated method stub
```

- 完整代码

```
1. package login.frame;
2.
3. import login.panel.LoginPanel;
4. //import parent.frame.ParentFrame;
5.
6. public class LoginFrame extends ParentFrame{
7.
8.     /**
9.     *
10.    */
11.    private static final long serialVersionUID = 1L;
12.
```

```

13. // public JPanel adminPage = new JPanel();
14.
15. public static LoginFrame instance = new LoginFrame();
16.
17. public LoginFrame() {
18.     super("问答机器人");
19.     LoginPanel.instance.setOpaque(false);
20.     this.add(LoginPanel.instance);
21.
22. // ImageIcon bg=new ImageIcon("C:/Users/55025/Desktop/Login2.j
    pg");
23. // JLabel label=new JLabel(bg);
24. // JPanel jPanel = new JPanel();
25. // label.setBounds(0, 0, acWidth(), acHeight());
26. // jPanel.add(label);
27. // getLayeredPane().add(jPanel, JLayeredPane.DEFAULT_LAYER);
28. }
29.
30. @Override
31. protected int acWidth() {
32.     // TODO Auto-generated method stub
33.     return 400;
34. }
35.
36. @Override
37. protected int acHeight() {
38.     // TODO Auto-generated method stub
39.     return 300;
40. }
41.
42. @Override
43. public void xianShi() {
44.     LoginPanel.instance.userNameInput.setText("");
45.     LoginPanel.instance.passwordInput.setText("");
46.     super.xianShi();
47. }
48.
49. }

```

- 文字说明

功能：

LoginFrame 是登录窗口的主要框架类，继承自 ParentFrame。它加载了一个登录面板 LoginPanel，并定义了窗口的宽度、高度以及显示逻辑。

详细解析：

继承与实例化：

```
public static LoginFrame instance = new LoginFrame();
```

使用单例模式，确保登录框架在整个应用中只有一个实例。

界面加载：

```
LoginPanel.instance.setOpaque(false);
```

```
this.add(LoginPanel.instance);
```

加载 LoginPanel 作为主内容面板，并设置其透明背景。

窗口尺寸：重写了 `acWidth()` 和 `acHeight()` 方法，返回窗口的宽度和高度（400x300），方便在父类 `ParentFrame` 中设置窗口大小。

显示逻辑：

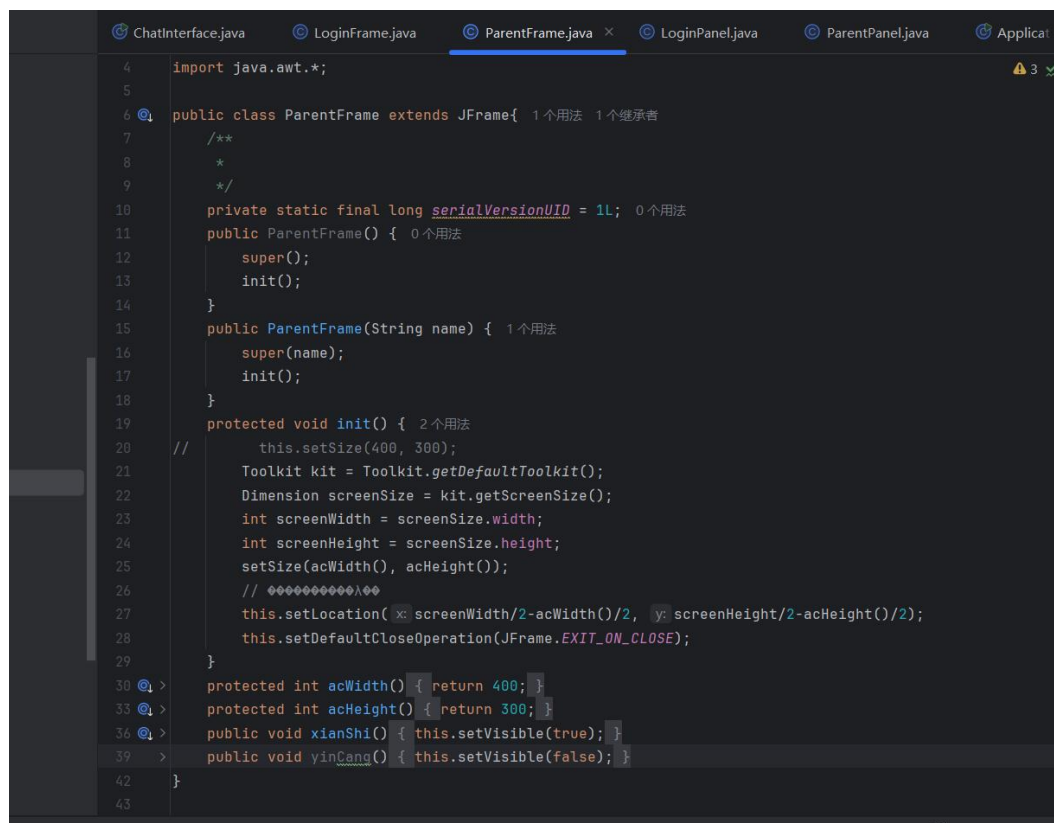
@Override

```
public void xianShi() {  
    LoginPanel.instance.userNameInput.setText("");  
    LoginPanel.instance.passwordInput.setText("");  
    super.xianShi();  
}
```

每次显示登录窗口时，都会清空用户名和密码的输入框。

6) ParentFrame.java

- 程序截图



```
import java.awt.*;  
  
public class ParentFrame extends JFrame {  
    /**  
     *  
     */  
    private static final long serialVersionUID = 1L;  
    public ParentFrame() {  
        super();  
        init();  
    }  
    public ParentFrame(String name) {  
        super(name);  
        init();  
    }  
    protected void init() {  
        // this.setSize(400, 300);  
        Toolkit kit = Toolkit.getDefaultToolkit();  
        Dimension screenSize = kit.getScreenSize();  
        int screenWidth = screenSize.width;  
        int screenHeight = screenSize.height;  
        setSize(acWidth(), acHeight());  
        // *****  
        this.setLocation(x: screenWidth/2-acWidth()/2, y: screenHeight/2-acHeight()/2);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
    protected int acWidth() { return 400; }  
    protected int acHeight() { return 300; }  
    public void xianShi() { this.setVisible(true); }  
    public void yinCang() { this.setVisible(false); }  
}
```

- 文字说明

功能：

`ParentFrame` 是一个基础框架类，提供了窗口初始化、显示和隐藏的通用方法，为 `LoginFrame` 等子类提供继承支持。

详细解析：

窗口初始化:

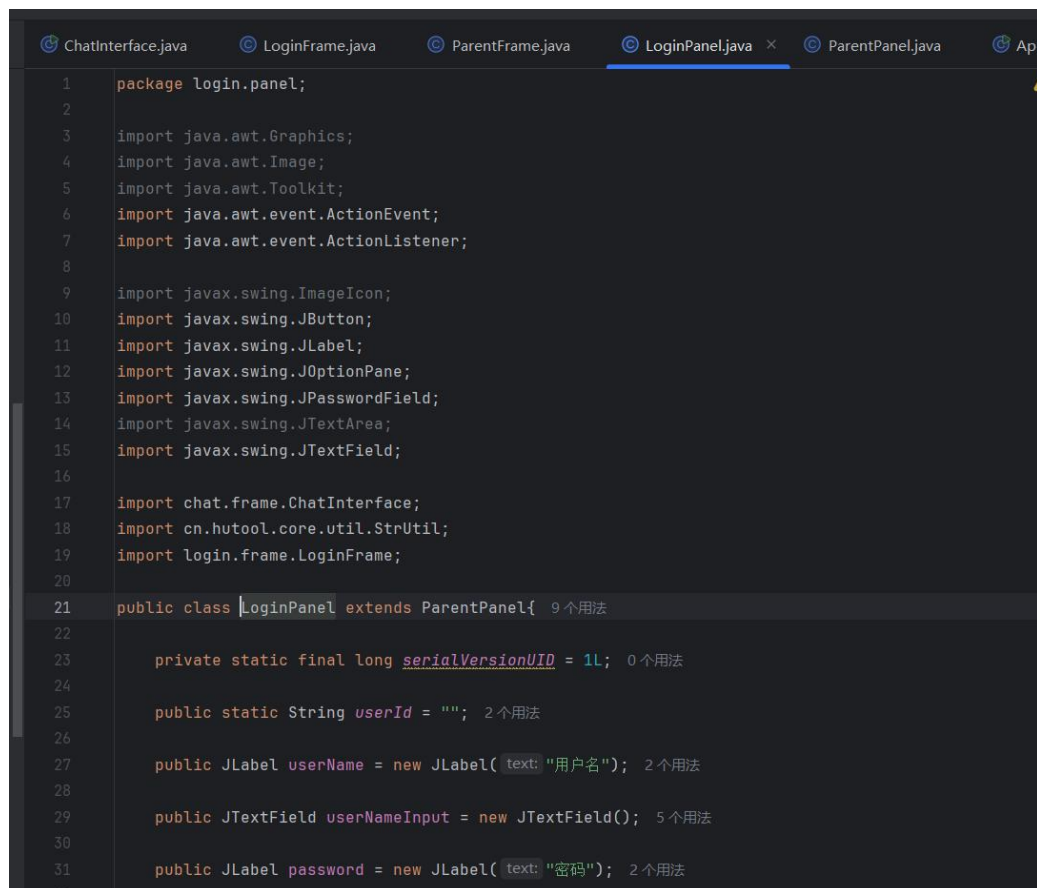
```
protected void init() {  
    Toolkit kit = Toolkit.getDefaultToolkit();  
    Dimension screenSize = kit.getScreenSize();  
    int screenWidth = screenSize.width;  
    int screenHeight = screenSize.height;  
  
    setSize(acWidth(), acHeight());  
    this.setLocation(screenWidth / 2 - acWidth() / 2, screenHeight / 2 - acHeight() / 2);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

自动将窗口设置为屏幕中心位置, 窗口尺寸由子类通过 `acWidth()` 和 `acHeight()` 方法定义。

显示和隐藏: 提供了简单的 `xianShi()` 和 `yinCang()` 方法, 用于控制窗口的可见性。

7) LoginPanel.java

- 程序截图



```
1 package login.panel;  
2  
3 import java.awt.Graphics;  
4 import java.awt.Image;  
5 import java.awt.Toolkit;  
6 import java.awt.event.ActionEvent;  
7 import java.awt.event.ActionListener;  
8  
9 import javax.swing.ImageIcon;  
10 import javax.swing.JButton;  
11 import javax.swing.JLabel;  
12 import javax.swing.JOptionPane;  
13 import javax.swing.JPasswordField;  
14 import javax.swing.JTextArea;  
15 import javax.swing.JTextField;  
16  
17 import chat.frame.ChatInterface;  
18 import cn.hutool.core.util.StrUtil;  
19 import login.frame.LoginFrame;  
20  
21 public class LoginPanel extends ParentPanel{ 9个用法  
22  
23     private static final long serialVersionUID = 1L; 0个用法  
24  
25     public static String userId = ""; 2个用法  
26  
27     public JLabel userName = new JLabel( text: "用户名"); 2个用法  
28  
29     public JTextField userNameInput = new JTextField(); 5个用法  
30  
31     public JLabel password = new JLabel( text: "密码"); 2个用法
```

- 完整代码

```
1. package login.panel;  
2.  
3. import java.awt.Graphics;  
4. import java.awt.Image;
```

```
5.  import java.awt.Toolkit;
6.  import java.awt.event.ActionEvent;
7.  import java.awt.event.ActionListener;
8.
9.  import javax.swing.ImageIcon;
10. import javax.swing.JButton;
11. import javax.swing.JLabel;
12. import javax.swing.JOptionPane;
13. import javax.swing.JPasswordField;
14. import javax.swing.JTextArea;
15. import javax.swing.JTextField;
16.
17. import chat.frame.ChatInterface;
18. import cn.hutool.core.util.StrUtil;
19. import login.frame.LoginFrame;
20.
21. public class LoginPanel extends ParentPanel{
22.
23.     private static final long serialVersionUID = 1L;
24.
25.     public static String userId = "";
26.
27.     public JLabel userName = new JLabel("用户名");
28.
29.     public JTextField userNameInput = new JTextField();
30.
31.     public JLabel password = new JLabel("密码");
32.
33.     public JPasswordField passwordInput = new JPasswordField();
34.
35.     public JButton loginButton = new JButton("登录");
36.
37.     public JButton reset = new JButton("重置");
38.
39.     public static LoginPanel instance = new LoginPanel();
40.
41.     public LoginPanel() {
42.
43.         super();
44.
45.         this.setLayout(null);
46.         this.add(userName);
47.         this.add(userNameInput);
48.         this.add(password);
```

```
49.     this.add(passwordInput);
50.     this.add(loginButton);
51.     this.add(reset);
52.
53.     userName.setBounds(50,40, 150, 30);
54.     userNameInput.setBounds(100,40, 180, 30);
55.     password.setBounds(50,80, 150, 30);
56.     passwordInput.setBounds(100,80, 180, 30);
57.     loginButton.setBounds(80,120, 70, 30);
58.     reset.setBounds(180,120, 70, 30);
59.
60.     loginButton.addActionListener(new ActionListener() {
61.
62.         @Override
63.         public void actionPerformed(ActionEvent e) {
64.             // TODO Auto-generated method stub
65.             String userName = userNameInput.getText();
66.             String password = passwordInput.getText();
67.             // User dbUser = new UserDAO().queryByUserNameAndPassword(us
erName, password);
68.             if((StringUtil.equals(userName, "1") && StringUtil.equals(password, "1"))
69.                 ||(StringUtil.equals(userName, "2") && StringUtil.equals(password,
"2"))
70.                 ||(StringUtil.equals(userName, "3") && StringUtil.equals(password, "3"))
71.                 ||(StringUtil.equals(userName, "4") && StringUtil.equals(password, "4"))
72.                 ||(StringUtil.equals(userName, "5") && StringUtil.equals(password, "5"))
73.             ){
74.                 // UserManager.instance.setUser(dbUser);
75.                 // if(UserManager.instance.isAdmin()){
76.                 //     LoginFrame.instance.setVisible(false);
77.                 //     // AdminFrame.instance.xianShi();
78.                 // }else{
79.                 //     LoginFrame.instance.setVisible(false);
80.                 //     // SearchFrame.instance.xianShi();
81.                 // }
82.                 userId = userName;
83.                 LoginFrame.instance.setVisible(false);
84.                 // AdminFrame.instance.xianShi();
85.                 // ChatFrame.instance.xianShi();
86.                 new ChatInterface().setVisible(true);
```

```

87. // JOptionPane.showMessageDialog(LoginFrame.instance, "用户名
    或密码错误!");
88. }else{
89.     JOptionPane.showMessageDialog(LoginFrame.instance, "用户名
    或密码错误!");
90. }
91. }
92.
93. });
94.
95. reset.addActionListener(new ActionListener() {
96.
97.     @Override
98.     public void actionPerformed(ActionEvent e) {
99.         userNameInput.setText("");
100.        passwordInput.setText("");
101.    }
102.
103. });
104.
105. }
106.
107. // public void paintComponent(Graphics g) {
108. // super.paintComponent(g);
109. //
110. // ImageIcon icon=new ImageIcon(getClass().getResource("C:/Use
    rs/55025/Desktop/login.jpg"));
111. // Image img=icon.getImage();
112. // // 设置图像的初始位置
    // 设置图像的初始位置
    // 设置图像的初始位置
113. // g.drawImage(img, 0, 0, this.getWidth(), this.getHeight(), th
    is);
114. // }
115. //
116. }

```

• 文字说明

功能:

LoginPanel 是主要的登录面板, 包含用户名、密码输入框, 以及“登录”和“重置”按钮, 负责用户输入处理与登录逻辑。

详细解析:

界面布局:

```
this.setLayout(null);
```

```
this.add(userName);
```

```
this.add(userNameInput);
```



```

this.add(password);
this.add(passwordInput);
this.add(loginButton);
this.add(reset);
使用空布局（null），手动设置每个组件的位置和大小。
登录逻辑：
loginButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
String userName = userNameInput.getText();
String password = passwordInput.getText();
if ((StringUtil.equals(userName, "1") && StringUtil.equals(password, "1"))
|| ...) {
userId = userName;
LoginFrame.instance.setVisible(false);
new ChatInterface().setVisible(true);
} else {
JOptionPane.showMessageDialog(LoginFrame.instance, "用户名或密码错误！");
}
}
});

```

使用简单的用户名和密码匹配逻辑（硬编码）。

登录成功后，打开新的聊天界面（ChatInterface）。

登录失败时，弹出提示信息。

重置功能：

```

reset.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
userNameInput.setText("");
passwordInput.setText("");
}
});

```

清空用户名和密码输入框。

8) ParentPanel.java

- 程序截图

```
ChatInterface.java LoginFrame.java ParentFrame.java LoginPanel.java ParentPanel.java x Applica
1 package login.panel;
2
3 import javax.swing.*;
4
5 public class ParentPanel extends JPanel { 1个用法 1个继承者
6
7     /**
8      *
9      */
10    private static final long serialVersionUID = 1L; 0个用法
11
12    public ParentPanel() { 1个用法
13        // TODO Auto-generated constructor stub
14        super();
15    }
16
17 }
18
```

- 文字说明

功能:

ParentPanel 是一个简单的基础面板类，提供了面板组件的基本功能，供子类（如 LoginPanel）继承。

9) ThemeUtil.java

- 程序截图

```
terface.java LoginFrame.java ParentFrame.java LoginPanel.java ParentPanel.java ThemeUtil.java x Application.java
1 package util;
2
3 import org.jb2011.lnf.beautyeye.BeautyEyeLNFHelper;
4
5 import javax.swing.*;
6
7 public class ThemeUtil { 1个用法
8
9     public static void setTheme() { 0个用法
10         try {
11             BeautyEyeLNFHelper.frameBorderStyle = BeautyEyeLNFHelper.FrameBorderStyle.generalNoTranslucencyShadow;
12             UIManager.put("RootPane.setupButtonVisible", false);
13             // ****
14             BeautyEyeLNFHelper.launchBeautyEyeLNF();
15         } catch (Exception e) {
16             // TODO: handle exception
17             e.printStackTrace();
18         }
19     }
20
21 }
22
```

- 文字说明

功能

setTheme() 方法主要作用：设置 BeautyEye 主题（一个美化 Java Swing 界面的第三方库）。

实现细节：使用 BeautyEyeLNFHelper 类调整主题样式。

关闭某些默认设置（如 RootPane 的设置按钮可见性）。

启动并应用 BeautyEye 主题。

详解

`BeautyEyeLNFHelper.frameBorderStyle`

设置窗口边框样式。

使用 `FrameBorderStyle.generalNoTranslucencyShadow`, 意味着 普通无透明阴影的边框样式。

`UIManager.put("RootPane.setupButtonVisible", false)`

通过修改 Swing 的 UI 属性, 隐藏窗口右上角的 "设置按钮"。

BeautyEye 默认界面可能包含不需要的按钮, 这一步禁用它。

`BeautyEyeLNFHelper.launchBeautyEyeLNF()`

启动并应用 BeautyEye 主题。

这是调用 BeautyEye 外观的核心方法, 用于替换 Swing 的默认外观 (Look and Feel, L&F)。

try-catch 块

使用异常捕获来防止由于主题设置问题导致程序崩溃。

如果出现问题, 调用 `e.printStackTrace()` 打印错误堆栈, 方便调试。

作用

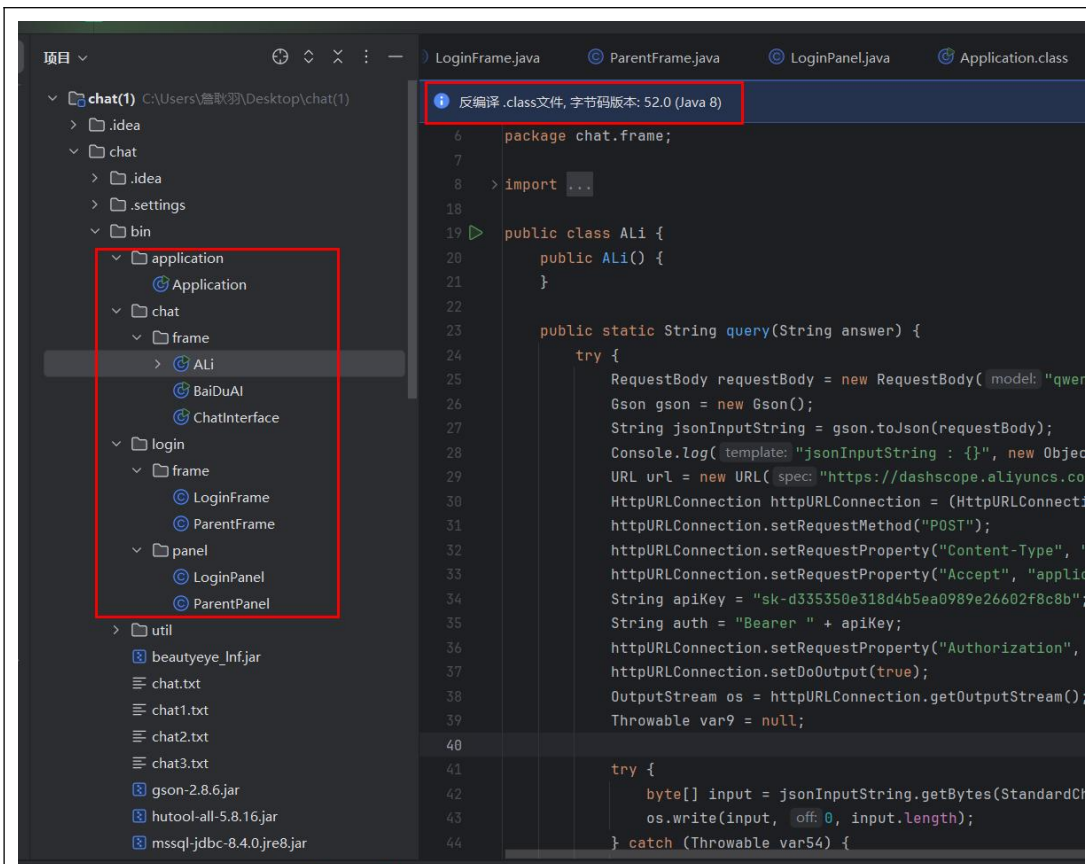
BeautyEye 是一个基于 Java Swing 的第三方 L&F 库, 可以让原生 Swing 界面更加现代化, 具体包括:

美化窗口标题栏、按钮、输入框、列表等组件。

提供多种边框样式和控件外观。

简单易用, 能够快速替换默认 Swing 外观。

3.Class 类的分析



1) Application 类

功能:

一个简单的主应用程序类, 用于启动程序。

调用了 LoginFrame 的单例模式实例, 并设置其为可见。

分析:

代码结构简单明了, 仅负责程序入口。

代码:

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package application;
7.
8.  import login.frame.LoginFrame;
9.
10. public class Application {
11.     public Application() {
12.     }
13.
14.     public static void main(String[] args) {
15.         LoginFrame.instance.setVisible(true);
16.     }
```

17. }

2) ALi 类

功能:

用于与阿里云 Qwen 的聊天接口交互。

构建 HTTPPOST 请求, 发送用户输入的问题, 并解析返回的 JSON 响应, 提取聊天机器人的回复。

分析:

使用了 Gson 和 Hutool 工具库, 提高了代码的简洁性和可读性。

具备完整的请求与响应解析流程, 处理了网络通信的关键细节。

代码:

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package chat.frame;
7.
8.  import cn.hutool.core.lang.Console;
9.  import cn.hutool.json.JSONArray;
10. import cn.hutool.json.JSONObject;
11. import com.google.gson.Gson;
12. import java.io.BufferedReader;
13. import java.io.InputStreamReader;
14. import java.io.OutputStream;
15. import java.net.HttpURLConnection;
16. import java.net.URL;
17. import java.nio.charset.StandardCharsets;
18.
19. public class ALi {
20.     public ALi() {
21.     }
22.
23.     public static String query(String answer) {
24.         try {
25.             RequestBody requestBody = new RequestBody("qwen-plus", new Message[]{new Message("user", answer)});
26.             Gson gson = new Gson();
27.             String jsonString = gson.toJson(requestBody);
28.             Console.log("jsonInputString : {}", new Object[]{jsonInputString});
29.             URL url = new URL("https://dashscope.aliyuncs.com/compatible-mode/v1/chat/completions");
30.             HttpURLConnection httpURLConnection = (HttpURLConnection
```

```
ction)url.openConnection();
31.         httpURLConnection.setRequestMethod("POST");
32.         httpURLConnection.setRequestProperty("Content-Type"
, "application/json; utf-8");
33.         httpURLConnection.setRequestProperty("Accept", "app
lication/json");
34.         String apiKey = "sk-d335350e318d4b5ea0989e26602f8c8
b";
35.         String auth = "Bearer " + apiKey;
36.         httpURLConnection.setRequestProperty("Authorization
", auth);
37.         httpURLConnection.setDoOutput(true);
38.         OutputStream os = httpURLConnection.getOutputStream
();
39.         Throwable var9 = null;
40.
41.         try {
42.             byte[] input = jsonString.getBytes(Standar
dCharsets.UTF_8);
43.             os.write(input, 0, input.length);
44.         } catch (Throwable var54) {
45.             var9 = var54;
46.             throw var54;
47.         } finally {
48.             if (os != null) {
49.                 if (var9 != null) {
50.                     try {
51.                         os.close();
52.                     } catch (Throwable var52) {
53.                         var9.addSuppressed(var52);
54.                     }
55.                 } else {
56.                     os.close();
57.                 }
58.             }
59.
60.         }
61.
62.         int responseCode = httpURLConnection.getResponseCod
e();
63.         System.out.println("Response Code: " + responseCode
);
64.         BufferedReader br = new BufferedReader(new InputStr
eamReader(httpURLConnection.getInputStream(), StandardCharsets.U
```

```

        TF_8));
65.         Throwable var63 = null;
66.
67.         try {
68.             StringBuilder response = new StringBuilder();
69.
70.             String responseLine;
71.             while((responseLine = br.readLine()) != null) {
72.                 response.append(responseLine.trim());
73.             }
74.
75.             System.out.println("Response Body: " + response
        );
76.             JSONObject jo = new JSONObject(response);
77.             JSONArray ja = jo.getJSONArray("choices");
78.             JSONObject oo = ja.getJSONObject(0);
79.             JSONObject mm = oo.getJSONObject("message");
80.             String cc = mm.getStr("content");
81.             Console.log("cc : {}", new Object[]{cc});
82.             String var18 = cc;
83.             return var18;
84.         } catch (Throwable var56) {
85.             var63 = var56;
86.             throw var56;
87.         } finally {
88.             if (br != null) {
89.                 if (var63 != null) {
90.                     try {
91.                         br.close();
92.                     } catch (Throwable var53) {
93.                         var63.addSuppressed(var53);
94.                     }
95.                 } else {
96.                     br.close();
97.                 }
98.             }
99.
100.        }
101.    } catch (Exception var58) {
102.        Exception e = var58;
103.        e.printStackTrace();
104.        return "";
105.    } finally {
106.        ;

```

```

107.     }
108. }
109.
110. public static void main(String[] args) {
111.     String answer = query("北京市天气如何");
112.     Console.log("answer : {}", new Object[]{answer});
113. }
114.
115. static class RequestBody {
116.     String model;
117.     Message[] messages;
118.
119.     public RequestBody(String model, Message[] messages) {
120.         this.model = model;
121.         this.messages = messages;
122.     }
123. }
124.
125. static class Message {
126.     String role;
127.     String content;
128.
129.     public Message(String role, String content) {
130.         this.role = role;
131.         this.content = content;
132.     }
133. }
134. }

```

3) BaiDuAI 类

功能:

调用百度 AI 接口, 通过 POST 请求发送用户问题, 接收并解析响应。

如果接口响应成功, 则返回 AI 的答案, 否则返回提示信息。

分析:

利用 Hutool 工具库简化了 HTTP 请求和 JSON 解析, 代码精简。

提供了一个备用的 AI 接口实现, 与 ALi 类功能互补。

代码:

```

1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package chat.frame;
7.

```



```

8.  import cn.hutool.core.lang.Console;
9.  import cn.hutool.core.util.ObjectUtil;
10. import cn.hutool.http.HttpUtil;
11. import cn.hutool.json.JSONObject;
12.
13. public class BaiDuAI {
14.     public static final String url = "https://luckycola.com.cn/
    ai/openwxxy";
15.     public static final String uid = "M0cNXW1718087588744FPC4KD
    AnWx";
16.     public static final String appKey = "6667f0686e15113935ab02
    03";
17.     public static final int isLongChat = 1;
18.
19.     public BaiDuAI() {
20.     }
21.
22.     public static String query(String ques) {
23.         JSONObject jsonObject = new JSONObject();
24.         jsonObject.put("ques", ques);
25.         jsonObject.put("appKey", "6667f0686e15113935ab0203");
26.         jsonObject.put("uid", "M0cNXW1718087588744FPC4KDAnWx");
27.         jsonObject.put("isLongChat", 1);
28.         String body = HttpUtil.post("https://luckycola.com.cn/a
    i/openwxxy", jsonObject.toString());
29.         Console.log("body : {}", new Object[]{body});
30.         JSONObject jo = new JSONObject(body);
31.         Integer i = jo.getInt("code");
32.         return ObjectUtil.equals(i, 0) ? jo.getJSONObject("data
    ").getStr("result") : "暂时无法解答，你可以问些其他问题！";
33.     }
34.
35.     public static void main(String[] args) {
36.         query("你是谁呀");
37.     }
38. }

```

4) ChatInterface 类

功能：

提供了一个 Swing 图形用户界面，用户可以通过界面向聊天机器人提问并接收回答。
支持选择不同的聊天机器人（百度、阿里）。

保存聊天记录到文件。

分析：

功能全面，涵盖了 UI 创建、机器人切换、聊天记录保存等。

线程分离，采用单独的线程处理响应，避免主线程阻塞。

使用了 Hutool 的 FileUtil 和 StrUtil，提高了文件操作的便捷性。

代码

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package chat.frame;
7.
8.  import cn.hutool.core.io.FileUtil;
9.  import cn.hutool.core.util.StrUtil;
10. import java.awt.BorderLayout;
11. import java.awt.Component;
12. import java.awt.event.ActionEvent;
13. import java.awt.event.ActionListener;
14. import java.nio.charset.Charset;
15. import javax.swing.JButton;
16. import javax.swing.JComboBox;
17. import javax.swing.JFrame;
18. import javax.swing.JMenu;
19. import javax.swing.JMenuBar;
20. import javax.swing.JMenuItem;
21. import javax.swing.JPanel;
22. import javax.swing.JScrollPane;
23. import javax.swing.JTextArea;
24. import javax.swing.JTextField;
25. import javax.swing.SwingUtilities;
26. import login.frame.LoginFrame;
27. import login.panel.LoginPanel;
28.
29. public class ChatInterface extends JFrame {
30.     public static String path = "D:\\graPro2024\\xiao\\AswingLi
        aoTian\\wareHouseManagerSystem\\wareHouseManagerSystem\\src\\";
31.     public static String path2 = "D:\\graPro2024\\xiao\\AswingL
        iaoTian\\wareHouseManagerSystem\\wareHouseManagerSystem\\src\\ch
        at2.txt";
32.     private JTextArea chatArea;
33.     private JTextField inputField;
34.     private JButton sendButton;
35.     private JButton zhongZhi;
36.     private JComboBox<String> comboBox;
37.     private static volatile boolean zz = false;
38.
```

```
39.     public ChatInterface() {
40.         this.createUI();
41.     }
42.
43.     private void createUI() {
44.         this.setTitle("知识问答机器人");
45.         this.setDefaultCloseOperation(3);
46.         this.setSize(600, 700);
47.         this.setLocationRelativeTo((Component)null);
48.         JPanel panel = new JPanel(new BorderLayout());
49.         this.chatArea = new JTextArea(12, 30);
50.         this.chatArea.setLineWrap(true);
51.         this.chatArea.setWrapStyleWord(true);
52.         this.chatArea.setEditable(false);
53.         String[] items = new String[]{"百度文心一言", "阿里通义千
    问"};
54.         this.comboBox = new JComboBox(items);
55.         panel.add(this.comboBox, "North");
56.         JScrollPane scrollPane = new JScrollPane(this.chatArea)
    ;
57.         panel.add(scrollPane, "Center");
58.         this.inputField = new JTextField(30);
59.         JPanel p = new JPanel();
60.         p.setLayout(new BorderLayout());
61.         p.add(this.inputField, "Center");
62.         panel.add(p, "South");
63.         JPanel p2 = new JPanel();
64.         this.zhongZhi = new JButton("中止");
65.         this.zhongZhi.addActionListener(new ActionListener() {
66.             public void actionPerformed(ActionEvent e) {
67.                 ChatInterface.zz = true;
68.             }
69.         });
70.         this.sendButton = new JButton("发送");
71.         this.sendButton.addActionListener(new ActionListener()
    {
72.             public void actionPerformed(ActionEvent e) {
73.                 ChatInterface.this.sendMessage();
74.             }
75.         });
76.         p2.add(this.sendButton);
77.         p2.add(this.zhongZhi);
78.         p.add(p2, "East");
79.         this.add(panel);
```

```
80.         JMenuBar mb = new JMenuBar();
81.         JMenu heSuan = new JMenu("菜单");
82.         JMenuItem jMenuItem = new JMenuItem("退出");
83.         heSuan.add(jMenuItem);
84.         mb.add(heSuan);
85.         this.setJMenuBar(mb);
86.         final ChatInterface c = this;
87.         jMenuItem.addActionListener(new ActionListener() {
88.             public void actionPerformed(ActionEvent e) {
89.                 c.setVisible(false);
90.                 LoginFrame.instance.setVisible(true);
91.             }
92.         });
93.         this.setDefaultCloseOperation(3);
94.     }
95.
96.     private void sendMessage() {
97.         String p = path;
98.         String fileName = "chat{}.txt";
99.         fileName = StrUtil.format(fileName, new Object[]{LoginPanel.userId});
100.        p = p + fileName;
101.        final String pp = p;
102.        final String s = (String)this.comboBox.getSelectedItem();
103.        final String message = this.inputField.getText();
104.        String wen = "";
105.        if (!message.trim().isEmpty()) {
106.            wen = "Me: " + message + "\n";
107.            this.chatArea.append(wen);
108.            this.inputField.setText("");
109.        }
110.
111.        FileUtil.appendString(wen, pp, Charset.defaultCharset());
112.        (new Thread(new Runnable() {
113.            public void run() {
114.                String huiDa = "";
115.                if (StrUtil.equals(s, "阿里通义千问")) {
116.                    huiDa = ALi.query(message);
117.                } else {
118.                    huiDa = BaiDuAI.query(message);
119.                }
120.
```

```

121.         String da = "";
122.         if (!huiDa.trim().isEmpty()) {
123.             da = "机器人: " + huiDa + "\n";
124.             ChatInterface.this.inputField.setText("");
125.
126.             for(int i = 0; i < da.length(); ++i) {
127.                 try {
128.                     Thread.sleep(20L);
129.                 } catch (Exception var5) {
130.                     Exception e = var5;
131.                     e.printStackTrace();
132.                 }
133.
134.                 if (ChatInterface.zz) {
135.                     ChatInterface.zz = false;
136.                     return;
137.                 }
138.
139.                 ChatInterface.this.chatArea.append(da.c
                    harAt(i) + "");
140.                 ChatInterface.this.inputField.repaint()
                    ;
141.             }
142.         }
143.
144.         FileUtil.appendString(da, pp, Charset.defaultCh
            arset());
145.     }
146.     })).start();
147. }
148.
149. public static void main(String[] args) {
150.     SwingUtilities.invokeLater(new Runnable() {
151.         public void run() {
152.             new ChatInterface();
153.         }
154.     });
155. }
156. }

```

5) LoginFrame 类

功能与特点

继承自 ParentFrame，表示登录窗口。

是一个单例模式的实现，通过 public static LoginFrame instance 提供全局唯一实例。

在构造函数中设置标题为“问答机器人”，并将 LoginPanel 添加到窗口中。

提供窗口宽度和高度的具体实现 (acWidth 和 acHeight)。

包含 xianShi 方法，用于显示窗口并清空用户名和密码输入框。

方法说明

LoginFrame()

构造函数，调用父类构造函数设置窗口标题。

将 LoginPanel 的单例对象添加到窗口中。

acWidth() & acHeight()

定义窗口的宽度（400）和高度（300）。

xianShi()

重置用户名和密码输入框为空。

调用父类的 xianShi 方法显示窗口。

优点

单例模式确保了登录窗口只有一个实例，方便全局管理。

通过继承 ParentFrame，避免了重复定义窗口初始化逻辑。

完整代码

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package login.frame;
7.
8.  import login.panel.LoginPanel;
9.
10. public class LoginFrame extends ParentFrame {
11.     private static final long serialVersionUID = 1L;
12.     public static LoginFrame instance = new LoginFrame();
13.
14.     public LoginFrame() {
15.         super("问答机器人");
16.         LoginPanel.instance.setOpaque(false);
17.         this.add(LoginPanel.instance);
18.     }
19.
20.     protected int acWidth() {
21.         return 400;
22.     }
23.
24.     protected int acHeight() {
25.         return 300;
26.     }
27.
28.     public void xianShi() {
```

```

29.         LoginPanel.instance.userNameInput.setText("");
30.         LoginPanel.instance.passwordInput.setText("");
31.         super.xianShi();
32.     }
33. }

```

6) ParentFrame 类

功能与特点

继承自 `JFrame`，作为通用的父窗口类，封装窗口的基本初始化逻辑。

提供了子类可以重写的窗口宽度和高度方法 (`acWidth`, `acHeight`)。

包含用于显示和隐藏窗口的方法。

方法说明

`ParentFrame()` & `ParentFrame(String name)`

无参和带标题的构造函数。

调用 `init()` 方法进行窗口初始化。

`init()`

初始化窗口的大小和位置，使窗口居中。

设置默认关闭操作为 `EXIT_ON_CLOSE`。

`xianShi()` & `yinCang()`

显示和隐藏窗口的封装方法。

优点

提供了通用的窗口初始化功能，便于子类复用。

屏蔽了复杂的窗口配置细节，提高代码可读性。

完整代码

```

1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package login.frame;
7.
8.  import java.awt.Dimension;
9.  import java.awt.Toolkit;
10. import javax.swing.JFrame;
11.
12. public class ParentFrame extends JFrame {
13.     private static final long serialVersionUID = 1L;
14.
15.     public ParentFrame() {
16.         this.init();
17.     }
18.
19.     public ParentFrame(String name) {
20.         super(name);

```

```

21.         this.init();
22.     }
23.
24.     protected void init() {
25.         Toolkit kit = Toolkit.getDefaultToolkit();
26.         Dimension screenSize = kit.getScreenSize();
27.         int screenWidth = screenSize.width;
28.         int screenHeight = screenSize.height;
29.         this.setSize(this.acWidth(), this.acHeight());
30.         this.setLocation(screenWidth / 2 - this.acWidth() / 2,
            screenHeight / 2 - this.acHeight() / 2);
31.         this.setDefaultCloseOperation(3);
32.     }
33.
34.     protected int acWidth() {
35.         return 400;
36.     }
37.
38.     protected int acHeight() {
39.         return 300;
40.     }
41.
42.     public void xianShi() {
43.         this.setVisible(true);
44.     }
45.
46.     public void yinCang() {
47.         this.setVisible(false);
48.     }
49. }

```

7) LoginPanel 类

功能与特点

继承自 ParentPanel，表示登录窗口中的主面板。

包含用户名、密码输入框及登录和重置按钮。

提供登录验证功能，成功后跳转到聊天界面，否则提示错误。

属性说明

userName & userNameInput 用户名标签和对应的文本输入框。

password & passwordInput 密码标签和对应的密码输入框。

loginButton & reset 登录和重置按钮。

方法说明

LoginPanel()构造函数，初始化组件并设置布局。

添加两个按钮的点击事件：登录按钮：校验用户名和密码是否匹配预设值（1~5）。

成功登录后跳转到聊天界面。

登录失败弹出错误提示。

重置按钮：清空输入框内容。

优点

清晰的布局设置，易于扩展。

登录验证逻辑和界面初始化分离，逻辑清晰。

完整代码

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package login.panel;
7.
8.  import chat.frame.ChatInterface;
9.  import cn.hutool.core.util.StrUtil;
10. import java.awt.LayoutManager;
11. import java.awt.event.ActionEvent;
12. import java.awt.event.ActionListener;
13. import javax.swing.JButton;
14. import javax.swing.JLabel;
15. import javax.swing.JOptionPane;
16. import javax.swing.JPasswordField;
17. import javax.swing.JTextField;
18. import login.frame.LoginFrame;
19.
20. public class LoginPanel extends ParentPanel {
21.     private static final long serialVersionUID = 1L;
22.     public static String userId = "";
23.     public JLabel userName = new JLabel("用户名");
24.     public JTextField userNameInput = new JTextField();
25.     public JLabel password = new JLabel("密码");
26.     public JPasswordField passwordInput = new JPasswordField();
27.     public JButton loginButton = new JButton("登录");
28.     public JButton reset = new JButton("重置");
29.     public static LoginPanel instance = new LoginPanel();
30.
31.     public LoginPanel() {
32.         this.setLayout((LayoutManager)null);
33.         this.add(this.userName);
34.         this.add(this.userNameInput);
35.         this.add(this.password);
36.         this.add(this.passwordInput);
37.         this.add(this.loginButton);
38.         this.add(this.reset);
```

```

39.         this.userName.setBounds(50, 40, 150, 30);
40.         this.userNameInput.setBounds(100, 40, 180, 30);
41.         this.password.setBounds(50, 80, 150, 30);
42.         this.passwordInput.setBounds(100, 80, 180, 30);
43.         this.loginButton.setBounds(80, 120, 70, 30);
44.         this.reset.setBounds(180, 120, 70, 30);
45.         this.loginButton.addActionListener(new ActionListener()
        {
46.             public void actionPerformed(ActionEvent e) {
47.                 String userName = LoginPanel.this.userNameInput
.getText();
48.                 String password = LoginPanel.this.passwordInput
.getText();
49.                 if ((!StringUtil.equals(userName, "1") || !StringUtil
.equals(password, "1")) && (!StringUtil.equals(userName, "2") || !S
trUtil.equals(password, "2")) && (!StringUtil.equals(userName, "3")
|| !StringUtil.equals(password, "3")) && (!StringUtil.equals(userName
, "4") || !StringUtil.equals(password, "4")) && (!StringUtil.equals(us
erName, "5") || !StringUtil.equals(password, "5")))) {
50.                     JOptionPane.showMessageDialog(LoginFrame.in
stance, "用户名或密码错误!");
51.                 } else {
52.                     LoginPanel.userId = userName;
53.                     LoginFrame.instance.setVisible(false);
54.                     (new ChatInterface()).setVisible(true);
55.                 }
56.
57.             }
58.         });
59.         this.reset.addActionListener(new ActionListener() {
60.             public void actionPerformed(ActionEvent e) {
61.                 LoginPanel.this.userNameInput.setText("");
62.                 LoginPanel.this.passwordInput.setText("");
63.             }
64.         });
65.     }
66. }

```

8) ParentPanel 类

功能与特点

继承自 JPanel, 是 LoginPanel 的父类。

目前没有具体逻辑, 仅作为扩展的基础类。

优点

提供继承的基础, 为未来扩展其他面板功能提供统一的入口。

完整代码

```
1.  //
2.  // Source code recreated from a .class file by IntelliJ IDEA
3.  // (powered by FernFlower decompiler)
4.  //
5.
6.  package login.panel;
7.
8.  import javax.swing.JPanel;
9.
10. public class ParentPanel extends JPanel {
11.     private static final long serialVersionUID = 1L;
12.
13.     public ParentPanel() {
14.     }
15. }
```

类之间的关系

- ParentFrame 和 LoginFrame

LoginFrame 继承了 ParentFrame，重用了窗口初始化的逻辑。

- ParentPanel 和 LoginPanel

LoginPanel 继承了 ParentPanel，提供了通用的面板扩展接口。

- LoginFrame 和 LoginPanel

LoginFrame 包含 LoginPanel，实现了窗口和内容的分离。

- LoginPanel 和事件逻辑

通过按钮的事件监听实现交互功能。

五、运行结果

1. 多线程的体现

ChatInterface 类中的多线程部分

```
1.  new Thread(new Runnable() {
2.      @Override
3.      public void run() {
4.          String huiDa = "";
5.          if (StrUtil.equals(s, "阿里通义千问")) {
6.              huiDa = ALi.query(message); // 调用阿里大模型的查询方法
7.          } else {
8.              huiDa = BaiDuAI.query(message); // 调用百度大模型的查询方法
9.          }
10.
11.          String da = "Bot: " + huiDa + "\n";
```

```

12.         chatArea.append(da); // 更新聊天界面
13.         FileUtil.appendString(da, pp, Charset.defaultCharset())
; // 记录到日志文件
14.     }
15. }).start();

```

每次用户点击“发送”按钮时，会启动一个新的线程处理查询操作。

通过 `new Thread` 创建新线程，这样的设计保证了查询操作（如调用阿里或百度大模型的接口）不会阻塞主线程，确保界面交互流畅。

模型的响应结果会异步更新到界面，同时记录到本地日志文件。

2. 大模型的调用和题目的测试（分为百度文心一言和阿里的通义千问）

为了确保测试结果的准确性和可靠性，我们采用了两个不同的 API 进行测试，并对每个测试步骤进行了详细的解答和记录。这样可以有效减少单一数据源可能带来的偏差，提高最终结果的可信度。

• 编程题测试

首先已经准备好了问题（问题在实验开头已经给出），现在询问机器人，以下是他的回复：



Me: 编写一个 Java 程序，计算并输出数组中的最大值；编写一个程序，检查输入的字符串是否是回文。
机器人: 当然可以！下面我将分别给出这两个 Java 程序的示例。

1. 计算并输出数组中的最大值

这个程序定义了一个方法 'findMax'，它接受一个整数数组作为参数，并返回数组中的最大值。然后在主函数中调用此方法，并打印结果。

```
...java
public class MaxValueInArray {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};
        int max = findMax(numbers);
        System.out.println("The maximum value in the array is: " + max);
    }

    public static int findMax(int[] array) {
        if (array == null || array.length == 0) {
            throw new IllegalArgumentException("Array must not be null or empty");
        }
        int max = array[0];
        for (int i = 1; i < array.length; i++) {
            if (array[i] > max) {
                max = array[i];
            }
        }
        return max;
    }
}
...
```

2. 检查输入的字符串是否是回文

这个程序定义了一个方法 'isPalindrome'，它接受一个字符串作为参数，检查该字符串是否为回文（即正读和反读都一样的字符串），并返回布尔值。主函数中通过用户输入获取字符串，并调用 'isPalindrome' 方法来判断并打印结果。

```
...java
import java.util.Scanner;

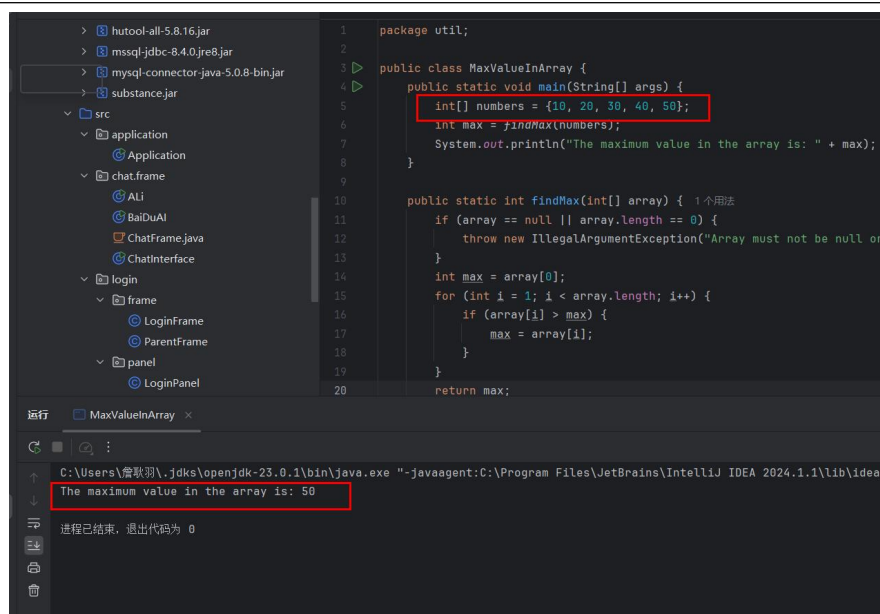
public class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string to check if it is a palindrome: ");
        String input = scanner.nextLine();
        boolean result = isPalindrome(input);
        if (result) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
        scanner.close();
    }

    public static boolean isPalindrome(String str) {
        if (str == null) {
            throw new IllegalArgumentException("String must not be null");
        }
        int left = 0;
        int right = str.length() - 1;
        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
...
```

对于“计算并输出数组中的最大值”，ai 回答：

```
3 public class MaxValueInArray {
4     public static void main(String[] args) {
5         int[] numbers = {10, 20, 30, 40, 50};
6         int max = findMax(numbers);
7         System.out.println("The maximum value in the array is: " + max);
8     }
9
10    public static int findMax(int[] array) { 1个用法
11        if (array == null || array.length == 0) {
12            throw new IllegalArgumentException("Array must not be null or empty");
13        }
14        int max = array[0];
15        for (int i = 1; i < array.length; i++) {
16            if (array[i] > max) {
17                max = array[i];
18            }
19        }
20        return max;
21    }
22 }
```

进行测试：



The screenshot shows the IntelliJ IDEA IDE. On the left, a project tree displays a package structure with files like Application, ChatInterface, LoginFrame, and LoginPanel. The main editor window shows the code for `MaxValueInArray.java`. The code defines a `package util;` and a `public class MaxValueInArray {`. Inside, there's a `main` method that takes a `String[] args` and an `findMax` method that takes an `int[] array`. The `main` method initializes an array `int[] numbers = {10, 20, 30, 40, 50};` and calls `findMax(numbers)`. The `findMax` method checks for null or empty arrays and then iterates through the array to find the maximum value. The execution output at the bottom shows the command used to run the program and the result: "The maximum value in the array is: 50".

```
1 package util;
2
3 public class MaxValueInArray {
4     public static void main(String[] args) {
5         int[] numbers = {10, 20, 30, 40, 50};
6         int max = findMax(numbers);
7         System.out.println("The maximum value in the array is: " + max);
8     }
9
10    public static int findMax(int[] array) { 1个用法
11        if (array == null || array.length == 0) {
12            throw new IllegalArgumentException("Array must not be null or
13        }
14        int max = array[0];
15        for (int i = 1; i < array.length; i++) {
16            if (array[i] > max) {
17                max = array[i];
18            }
19        }
20        return max;
21    }
```

运行 MaxValueInArray x

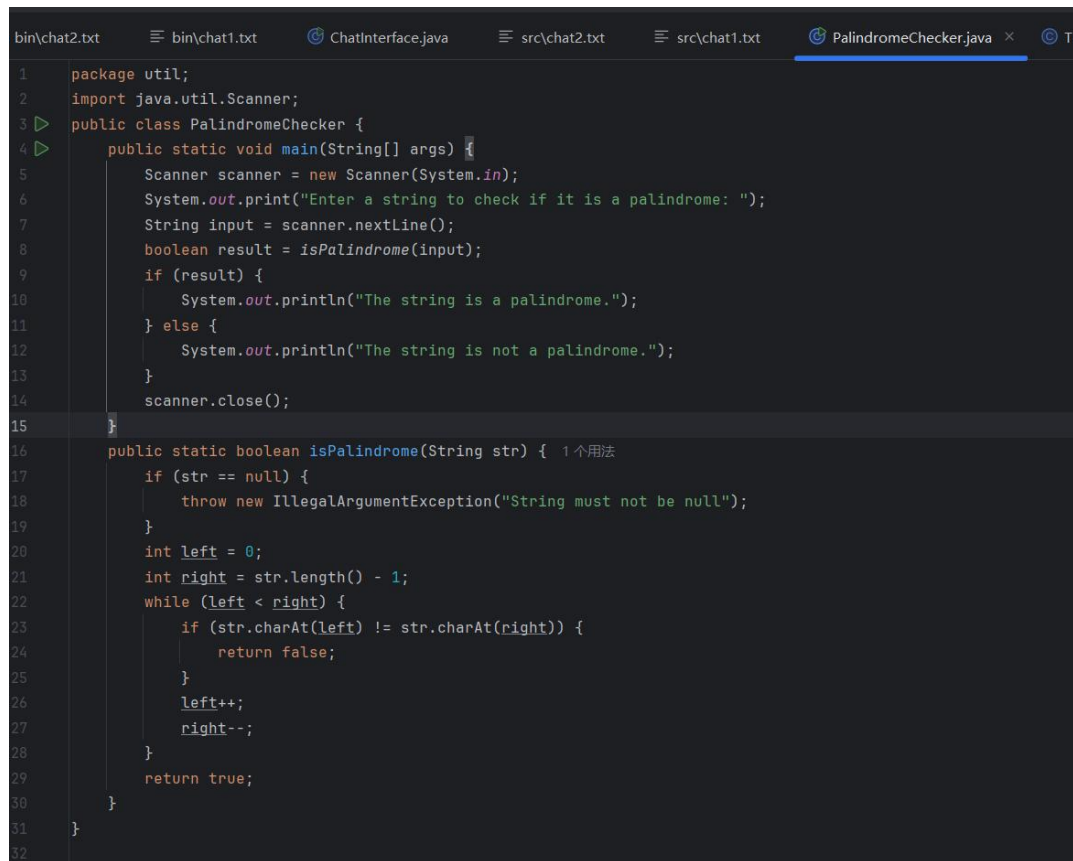
C:\Users\常歌羽\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1.1\lib\idea_...

The maximum value in the array is: 50

进程已结束，退出代码为 0

从中可见，答案正确。

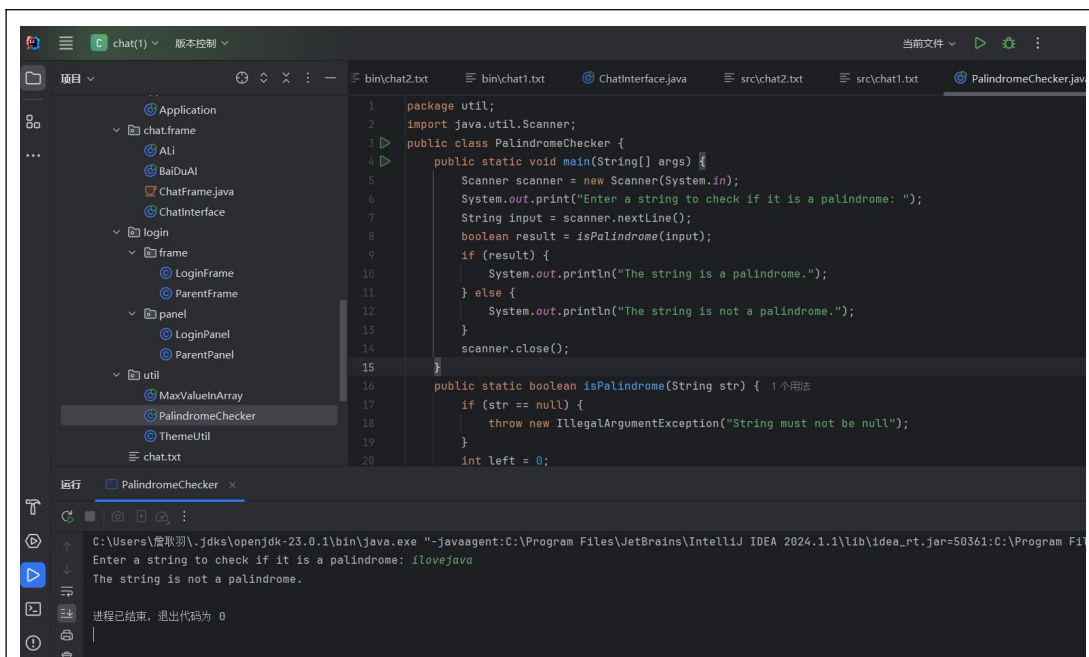
对于“检查输入的字符串是否是回文”，ai 回答：



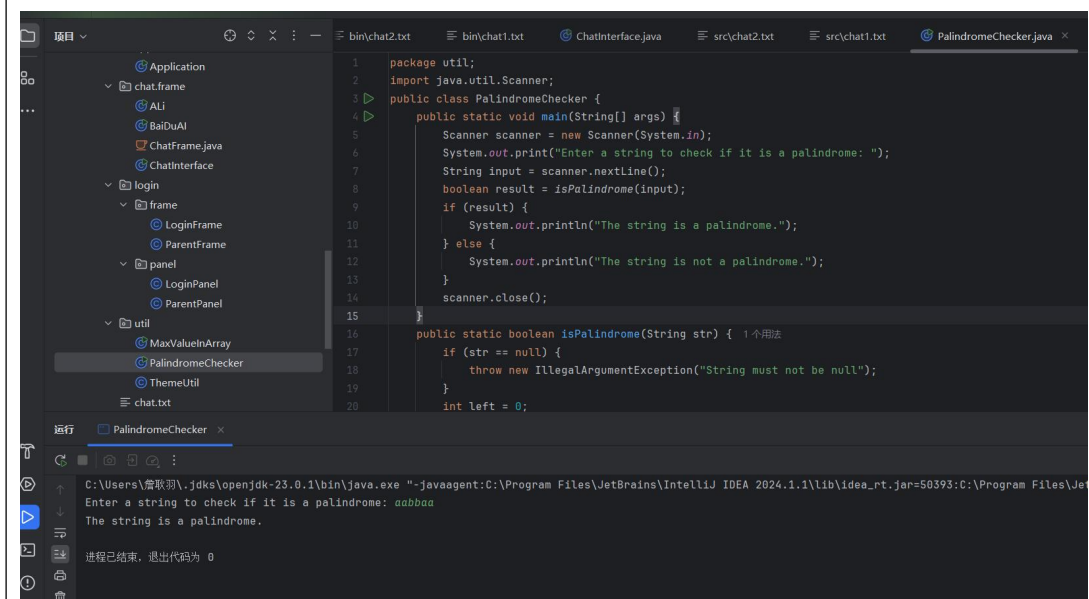
The screenshot shows the IntelliJ IDEA IDE with the `PalindromeChecker.java` file open. The code defines a `package util;` and a `public class PalindromeChecker {`. Inside, there's a `main` method that takes a `String[] args` and an `isPalindrome` method that takes a `String str`. The `main` method uses a `Scanner` to read input from the user and calls `isPalindrome(input)`. The `isPalindrome` method checks for null or empty strings and then uses a while loop to compare characters from both ends of the string towards the center. If the characters match, it continues; if not, it returns false. If the loop completes, it returns true.

```
1 package util;
2 import java.util.Scanner;
3 public class PalindromeChecker {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter a string to check if it is a palindrome: ");
7         String input = scanner.nextLine();
8         boolean result = isPalindrome(input);
9         if (result) {
10            System.out.println("The string is a palindrome.");
11        } else {
12            System.out.println("The string is not a palindrome.");
13        }
14        scanner.close();
15    }
16    public static boolean isPalindrome(String str) { 1个用法
17        if (str == null) {
18            throw new IllegalArgumentException("String must not be null");
19        }
20        int left = 0;
21        int right = str.length() - 1;
22        while (left < right) {
23            if (str.charAt(left) != str.charAt(right)) {
24                return false;
25            }
26            left++;
27            right--;
28        }
29        return true;
30    }
31 }
32 }
```

进行测试，输入 `ilovejava`，显示非回文：



输入 aabbaa，显示回文：



可见 ai 的代码正确。

- 选择题测试

将事先准备向 ai 提问：

知识问答机器人

菜单

百度文心一言

Me: 选择题 1.以下关于 Java 的描述中，哪一项是正确的？ A. Java 是一种面向过程的编程语言 B. Java 代码需要编译成机器码直接运行 C. Java 支持多继承 D. Java 使用字节码运行在虚拟机上； 2.以下关于 String 的描述中，哪一个错误的？ A. String 是不可变的类 B. String 可以使用 + 进行连接 C. String 是 Java 的基本数据类型 D. String 对象存储在字符串池中； 做一下上面两道Java知识选择题

机器人: 1. 以下关于 Java 的描述中，哪一项是正确的？

答案：D. Java 使用字节码运行在虚拟机上。

解释：

A. Java 是一种面向对象的编程语言，而不是面向过程的。

B. Java 代码需要编译成字节码，然后由 Java 虚拟机（JVM）解释或编译成机器码运行，而不是直接运行机器码。

C. Java 不支持多继承，但支持接口的多实现。

D. 正确，Java 的字节码是在 JVM 上运行的。

2. 以下关于 String 的描述中，哪一个错误的？

答案：C. String 是 Java 的基本数据类型。

解释：

A. String 是不可变的类，这意味着一旦一个字符串对象被创建并初始化，它的内容就不能被改变。

B. String 确实可以使用 + 进行连接来创建新的字符串。

C. 错误，String 不是 Java 的基本数据类型。Java 的基本数据类型包括 int、float、char 等。String 是一个类，用于表示和操作字符串。

D. String 对象通常存储在字符串池中，这样可以有效地重复使用相同的字符串对象。

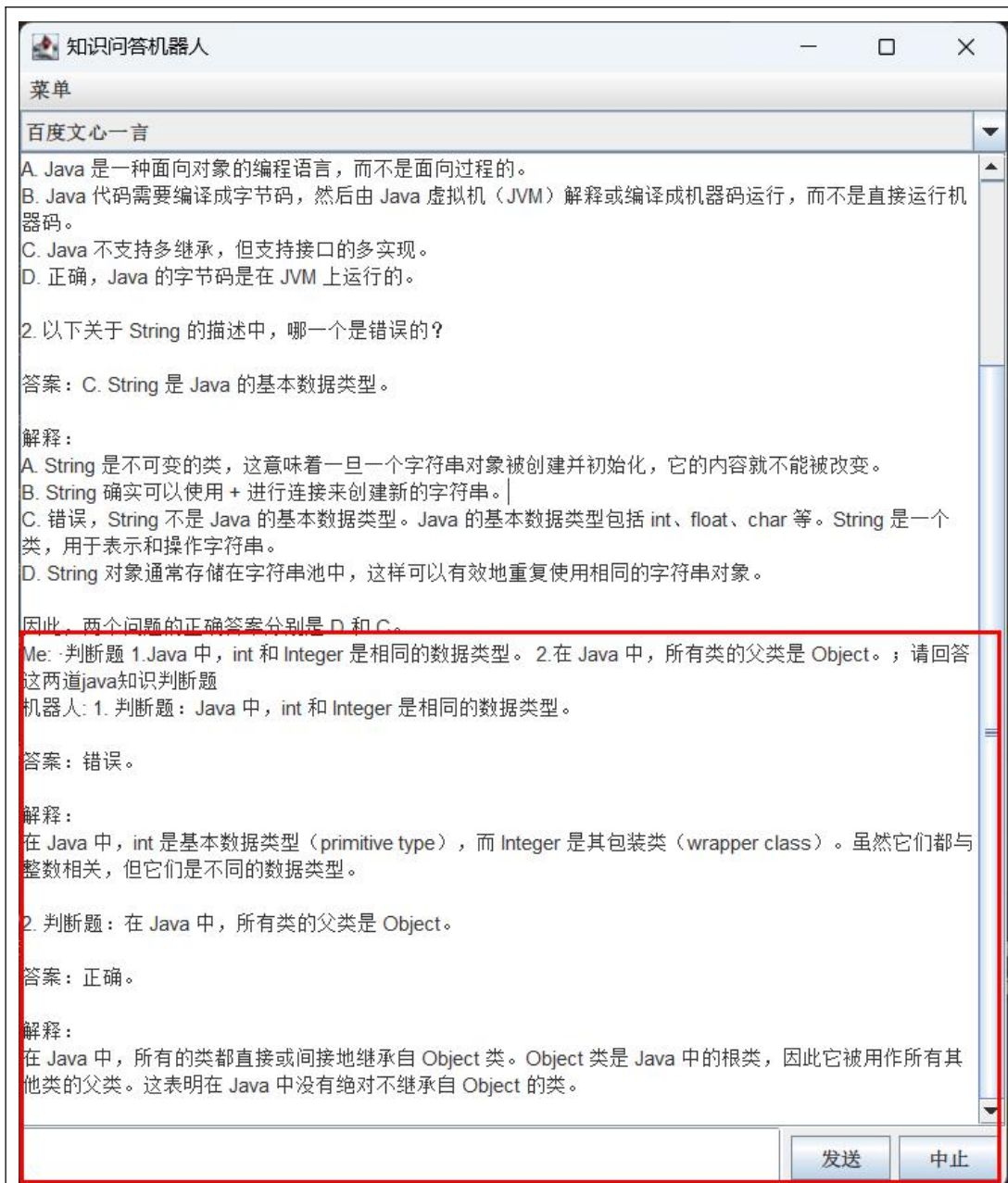
因此，两个问题的正确答案分别是 D 和 C。

发送 中止

从中可见，ai 的答案正确，解释也是合理的。

- 判断题测试

将事先准备向 ai 提问：



从中可见，ai 的答案正确，解释也是合理的。

• 编程填空题

将事先准备好的题目向机器人提问。

对于“计算两个数的最大公约数”，机器人回答如下：

知识问答机器人

菜单

百度文心一言 使用百度文心一言进行测试

Me: 1.计算两个数的最大公约数 请完成以下Java代码，计算两个整数的最大公约数（GCD）。 public class GCD { public static void main(String[] args) { int num1 = 56; int num2 = 98; System.out.println("GCD of " + num1 + " and " + num2 + " is: " + findGCD(num1, num2)); } public static int findGCD(int a, int b) { while (b != 0) { // 填空部分 } return a; }}

机器人: 要计算两个整数的最大公约数（GCD），你可以使用欧几里得算法（Euclidean algorithm）。在欧几里得算法中，我们会反复地将较大的数除以较小的数，并重复此过程直到两数相等或其中一个数为0。当其中一个数为0时，另一个数就是两数的最大公约数。

在Java中，你可以通过以下方式完成这个代码：

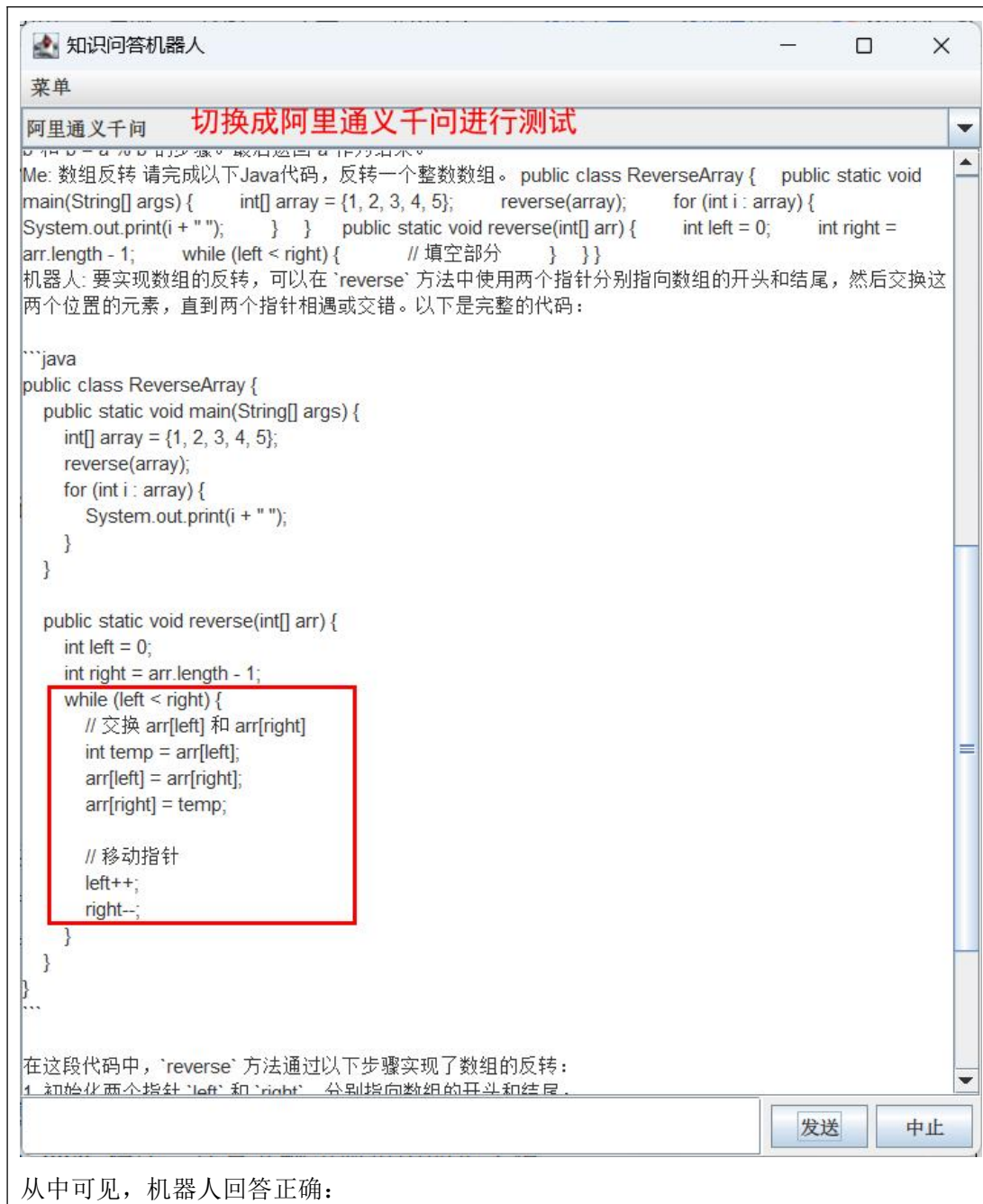
```
'''java
public class GCD {
    public static void main(String[] args) {
        int num1 = 56;
        int num2 = 98;
        System.out.println("GCD of " + num1 + " and " + num2 + " is: " + findGCD(num1, num2));
    }

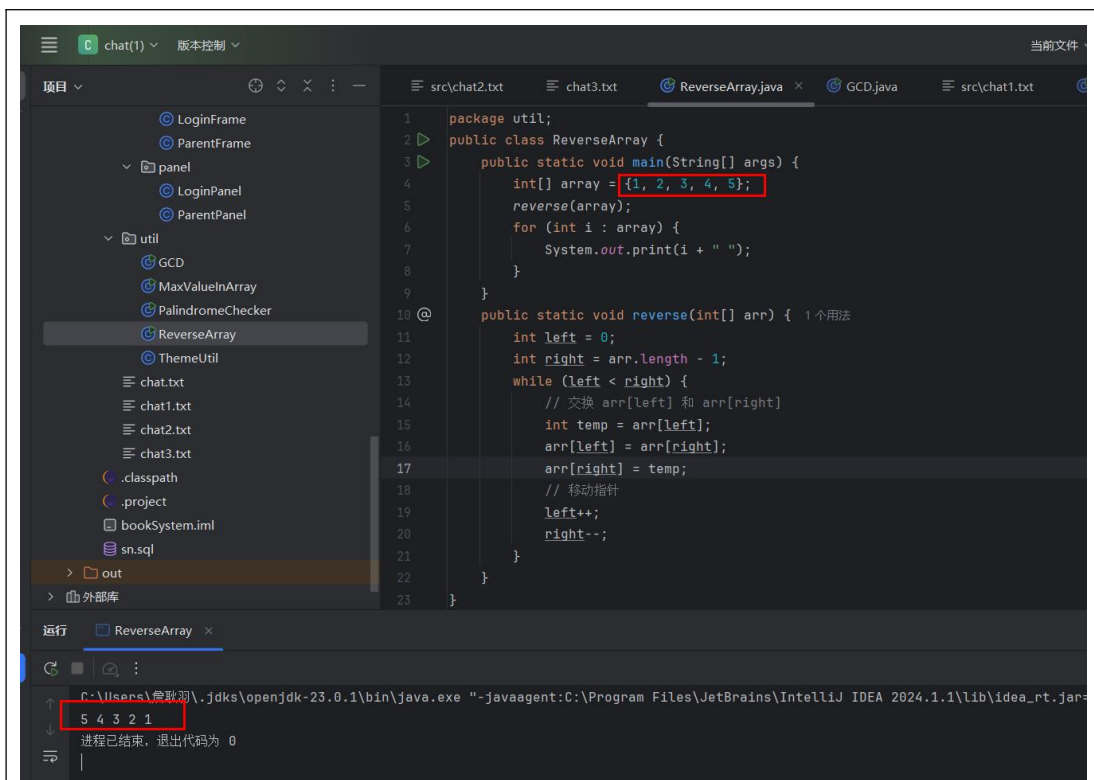
    public static int findGCD(int a, int b) {
        while (b != 0) { // 当b不为0时，继续循环
            int temp = b; // 保存b的值，因为接下来b会被新的值覆盖
            b = a % b; // 计算a除以b的余数，并将结果赋给b
            a = temp; // a现在应该是上一个b的值
        }
        return a; // 最后返回的就是GCD（因为当b变为0时，a即为最大公约数）
    }
}
'''
```

这段代码中，`findGCD` 方法实现了欧几里得算法来计算两个整数的最大公约数。在循环中，我们不断地用较小的数去除较大的数并获取余数，将余数作为新的较大数。循环继续直到其中一个数为0时结束，这时剩下的非零数就是最大公约数。当余数为0时，我们通过将`a`的值赋给`temp`来保存其值，然后继续执行`a = b`和`b = a % b`的步骤。最后返回`a`作为结果。

发送 中止

从中可得，机器人回答正确：





3. chat.txt 等文件的写入

1) 本地文本文件记录对话过程

日志记录通过 hutool 工具包的 `FileUtil.appendString` 方法，将用户输入和机器人回复逐条追加到本地文本文件中。具体实现如下：

在 `sendMessage` 方法中：

```
String wen = "Me: " + message + "\n";
```

```
chatArea.append(wen);
```

```
inputField.setText(""); // 清空输入框
```

```
FileUtil.appendString(wen, pp, Charset.defaultCharset()); // 将用户消息记录到日志文件
```

这里将用户的输入内容格式化为 "Me: {message}"，然后追加到文件路径 `pp` 指定的 .txt 文件中。

在 `Thread` 的 `run` 方法中：

```
String da = "机器人: " + huiDa + "\n";
```

```
FileUtil.appendString(da, pp, Charset.defaultCharset()); // 将机器人的回复记录到日志文件
```

机器人的回复同样以 "机器人: {huiDa}" 格式追加到同一个日志文件中。

2) 日志文件路径的动态生成

日志文件路径根据用户 ID 动态生成：

```
String p = path;
```

```
String fileName = "chat{}.txt";
```

```
fileName = StrUtil.format(fileName, LoginPanel.userId);
```

```
p += fileName;
```

日志文件的完整路径为：

C:\Users\詹耿羽\Desktop\chat(1)\chat\src\chat{userId}.txt

每个用户都有独立的日志文件，方便区分和管理。

3) 用户与机器人的对话记录格式

日志文件的格式为简单的文本形式，每一行记录一段对话，分为两种格式：

用户消息：Me: {用户消息}

机器人回复：机器人: {机器人回复}

例如，日志文件中的记录可能如下：

Me: 你好

机器人: 你好，有什么可以帮您的吗？

Me: 深圳天气怎么样？

机器人: 今天深圳市晴天，气温约为 15-25°C。

4) 多线程处理和日志记录的结合

日志记录的部分嵌套在多线程的处理逻辑中，具体通过以下代码实现：

```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        String huiDa = ALi.query(message); // 或 BaiDuAI.query(message)  
        String da = "机器人: " + huiDa + "\n";  
  
        // 将日志写入文件  
        FileUtil.appendString(da, pp, Charset.defaultCharset());  
    }  
}).start();
```

多线程保证了用户发送消息和系统处理回复的异步执行，不会阻塞主线程的运行。

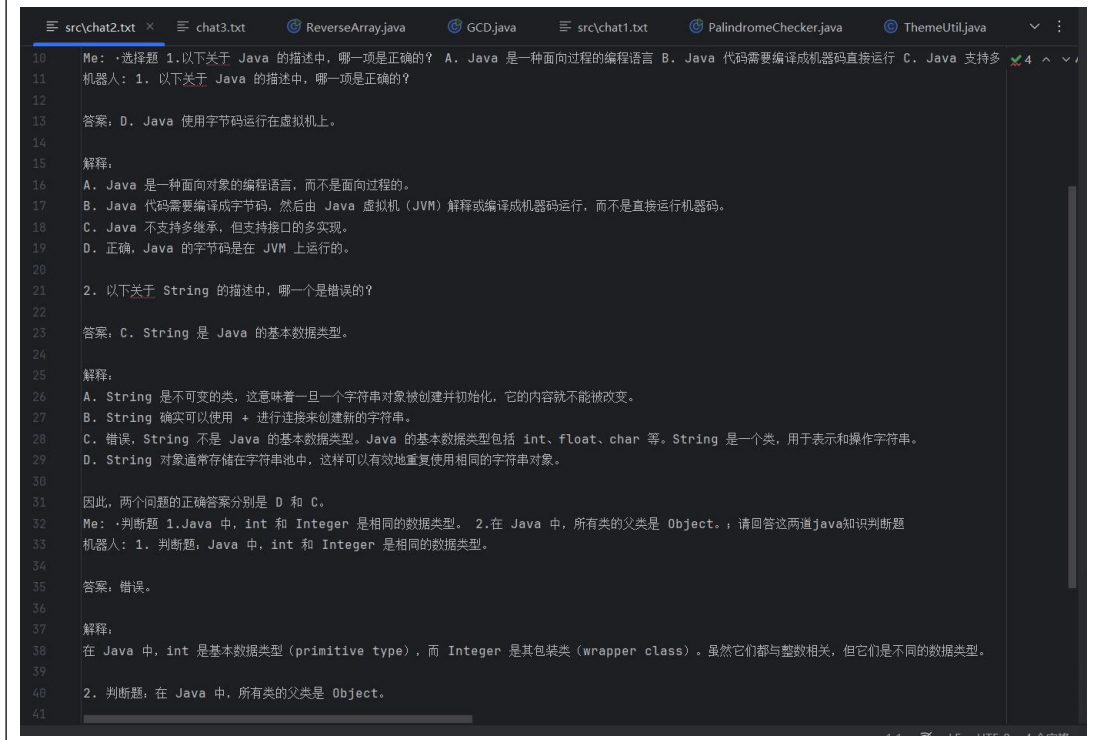
在多线程中，机器人回复通过 `FileUtil.appendString` 方法追加到日志文件中。

5) 总结

日志记录功能：

- 用户输入和机器人回复分别通过 `FileUtil.appendString` 方法写入本地 .txt 文件。
- 日志文件路径根据用户 ID 动态生成，保证了每个用户的对话记录独立保存。
- 日志内容格式清晰、简洁，便于后续查看和分析。
- 多线程的结合保证了对话日志记录的实时性，同时不影响界面响应速度。

6) 效果图



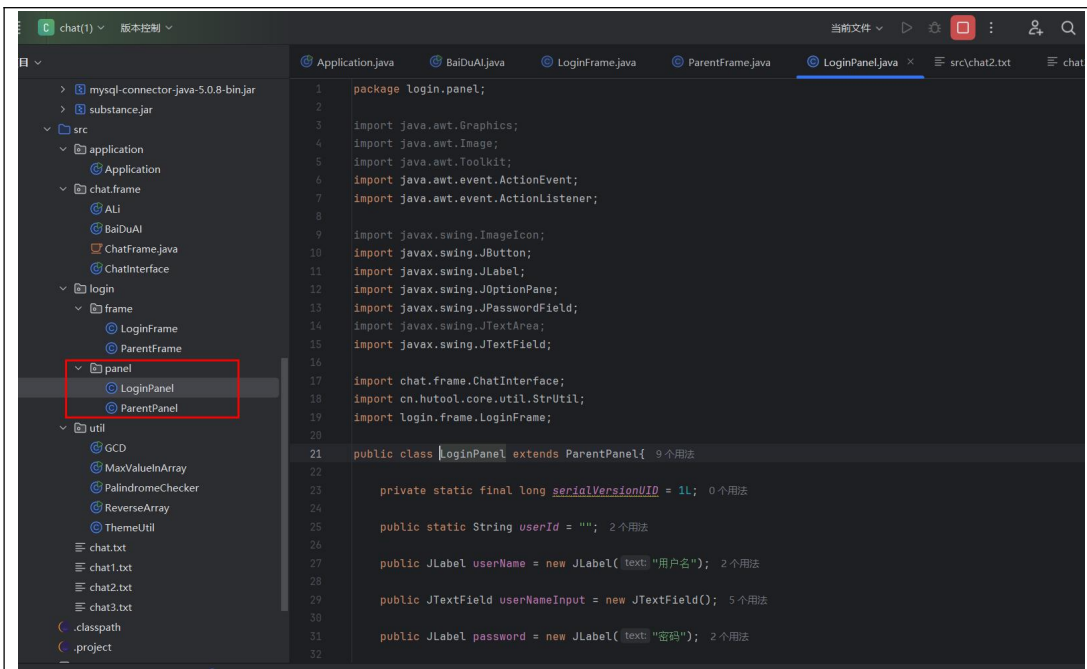


4. 特色功能的实现

1) 登录界面



相关代码为:



2) 用户系统

我设置了多个用户，如：

用户名：1；密码：1；

用户名：2；密码：2；

...

...

（相关代码分析已经在前面提及）



3) 分别存储功能

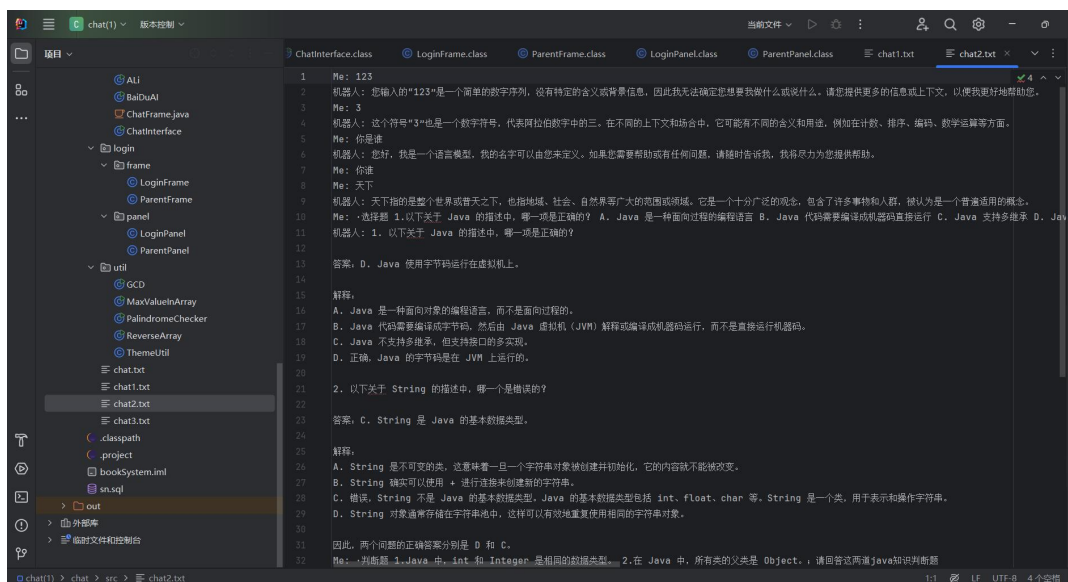
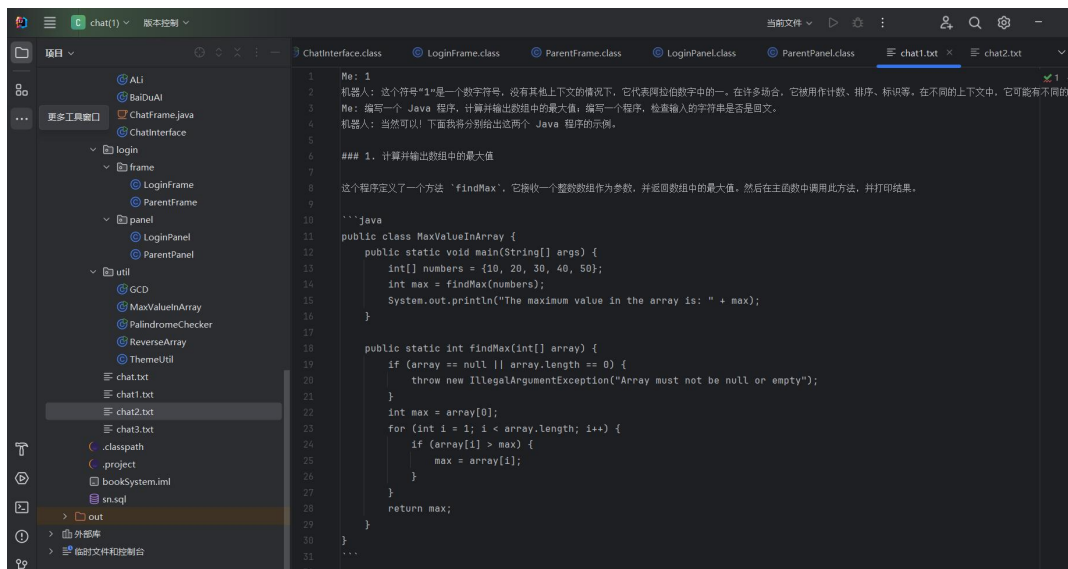
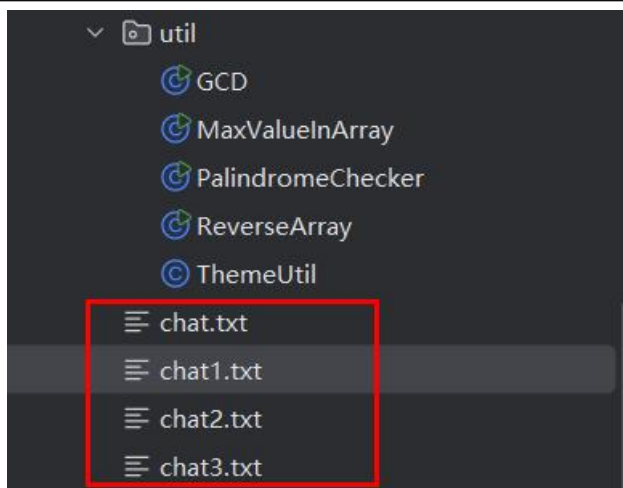
针对每个用户，都有每个用户自己专属的 chat.txt 用来存储对话信息。

如：

用户 1 的对话记录存储在 chat1.txt 中；

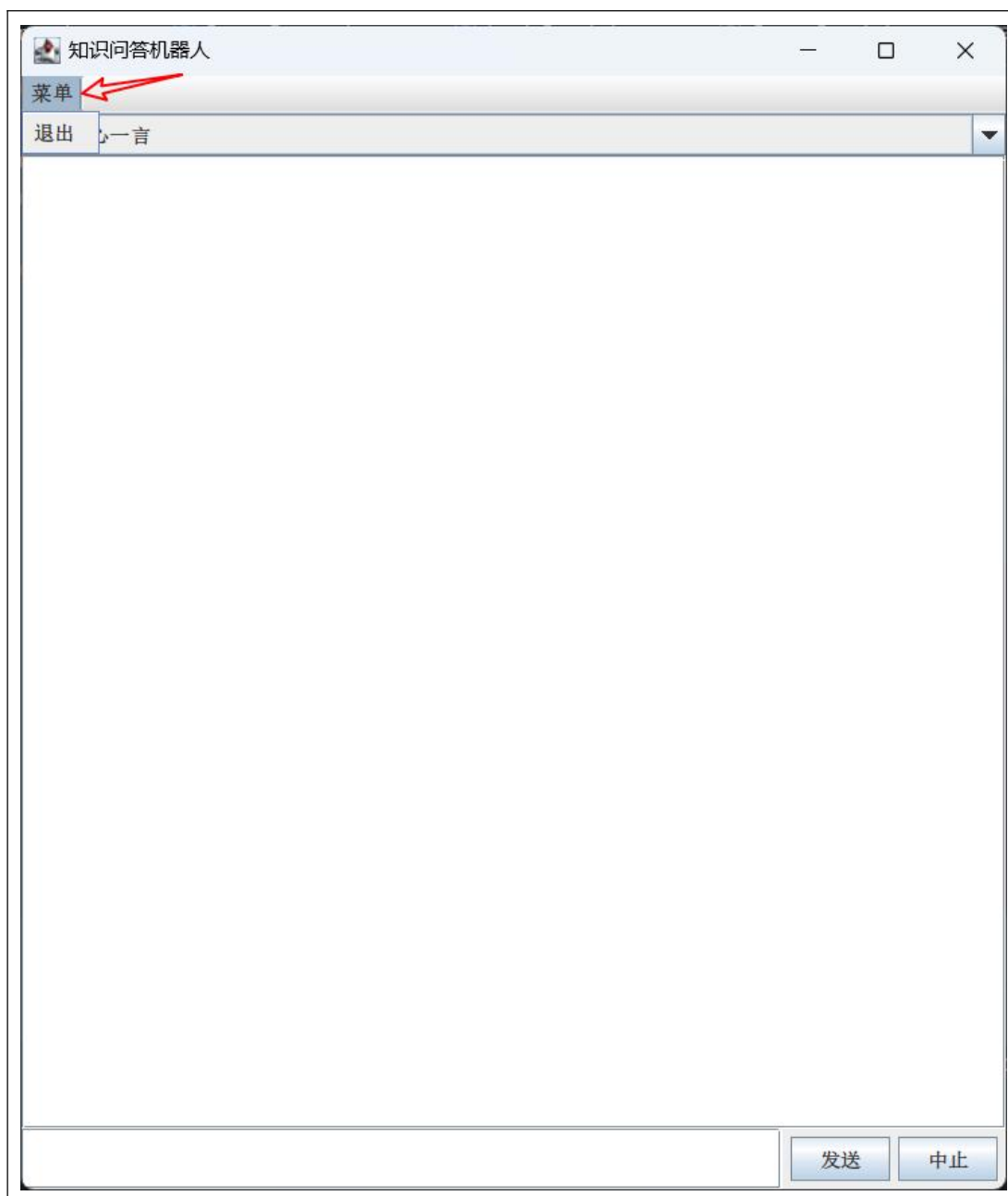
用户 2 的对话记录存储在 chat2.txt 中；

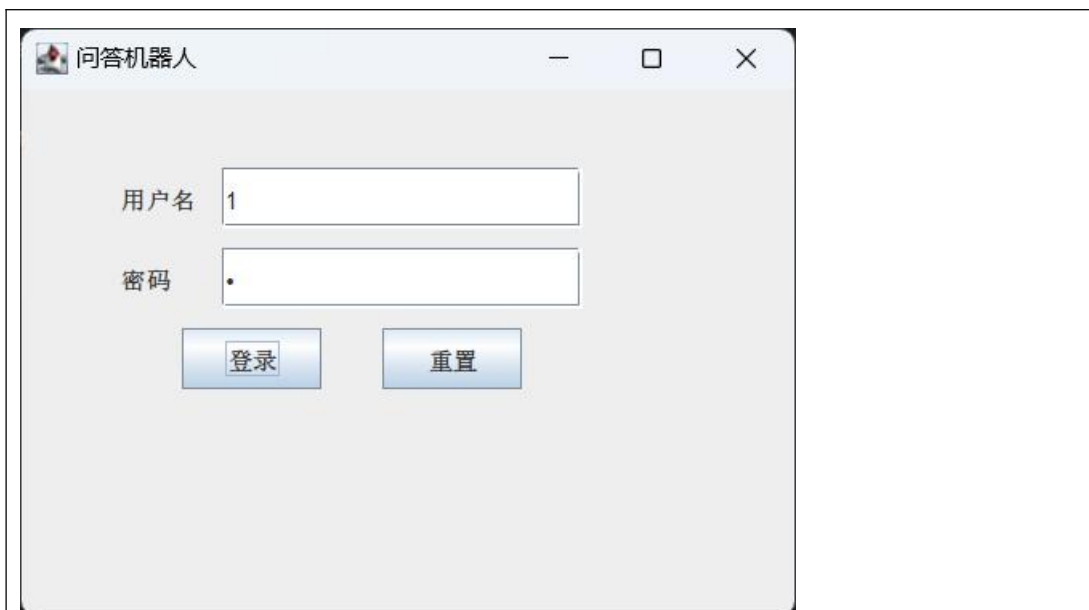
.....



4) 退出登录功能

点击菜单，跳出退出按钮。





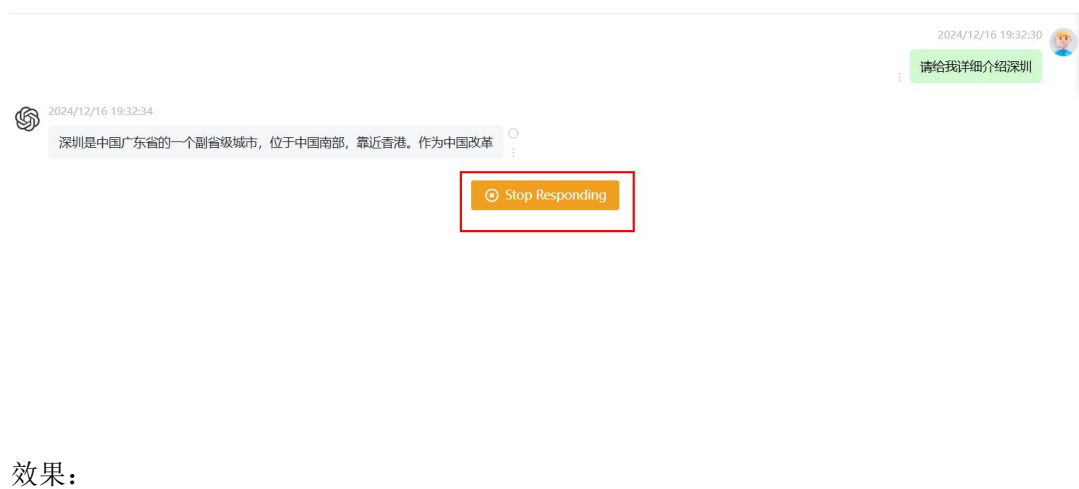
点击退出后，重新回到登录界面。

5) 中止功能

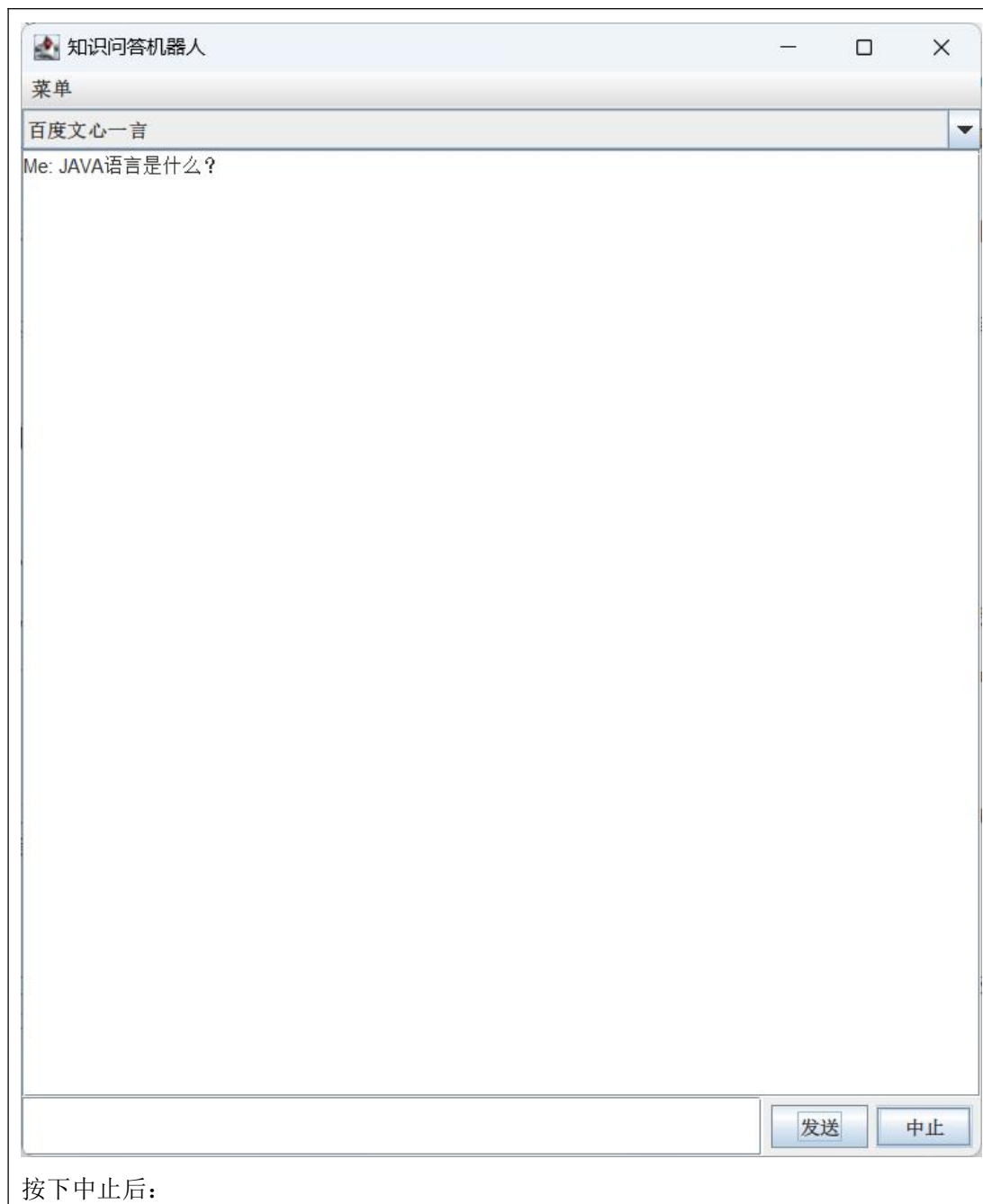
如果不想让机器人继续就该问题对话，可以点击“中止”按钮，使得机器人停止对话。但是接下来仍然可以继续问问题。

灵感来源：

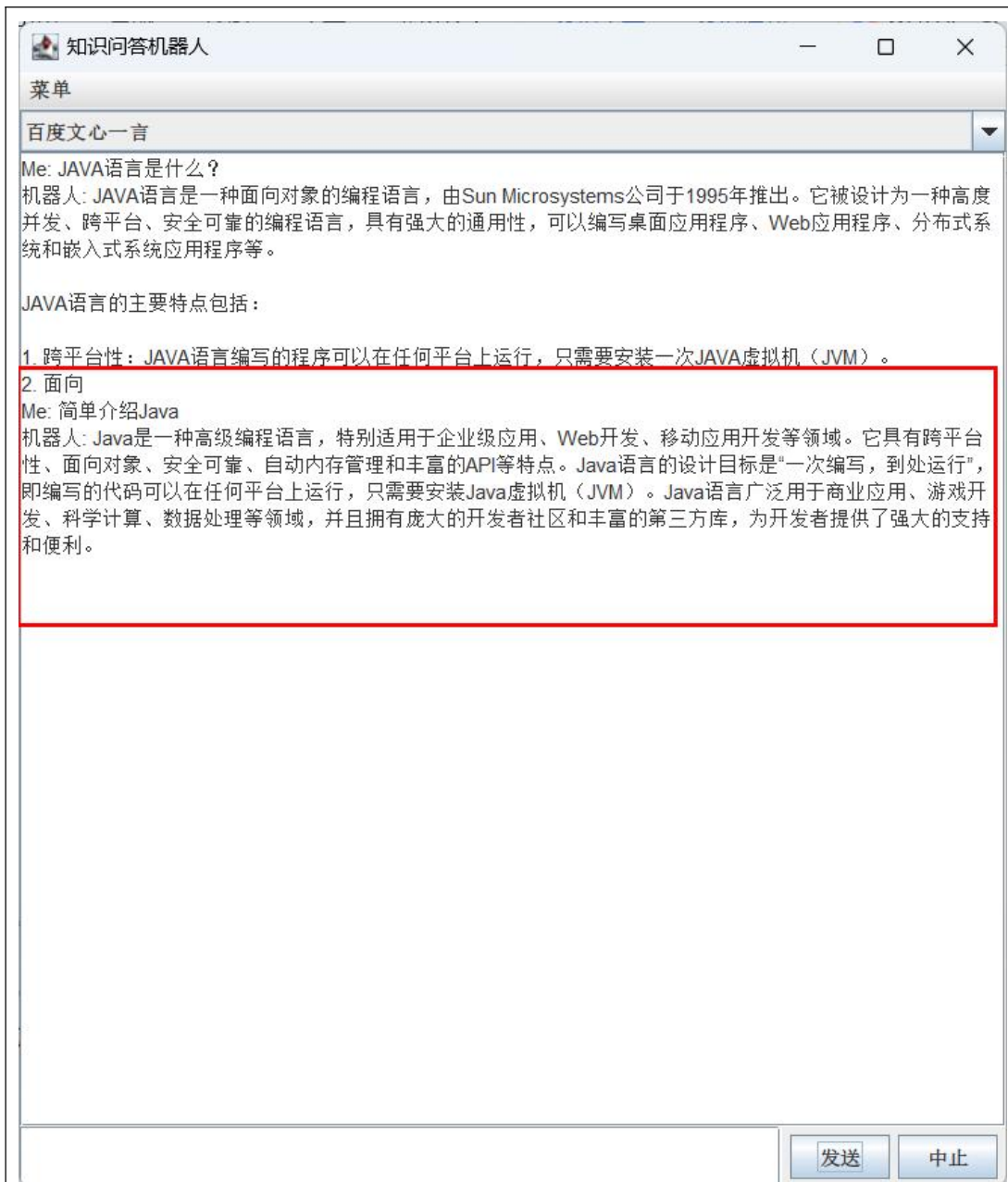
GPT 也有 STOP 功能，于是我自己也做了一个中止功能。



效果：







Tips: 源码已被压缩成 zip 提交，另外附上了 gif 动图，以更好地使用该机器人程序。

+++++

其他（例如感想、建议等等）。

本次实验难度较大，但是完成过后感觉收获颇丰。特色功能也是根据现实生活中的
一些灵感来进行改善的。

| |
|------------------|
| 指导教师批阅意见： |
| |
| 成绩评定： |
| |
| 指导教师签字： |
| 2024 年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生课程作业报告时间应在学生提交课程作业报告时间后 10 日内。