**深 圳 大 学 实 验 报 告**

**课程名称：Java程序设计**

**实验项目名称： 必实验5 GUI高级应用**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师：卢亚辉**

**报告人：林宪亮 学号：2022150130 班级：国际班**

**实验时间： 2023年12月7日（周四）~2022年12月13日（周三）**

**实验报告提交时间： 2023年12月10号**

**教务部制**

|  |
| --- |
| **实验目的与要求：**  **实验目的：** 初步掌握图形界面程序设计。  **实验要求：**  1.使用JLabel、JTextArea、JButton等控件实现句子的中译英demo，该demo包含四个文本框，在第一个文本框输入一句中文，在第二个和第三个文本框显示该句的英文翻译（要求使用百度翻译API、有道翻译API或其他API中的两种），在第四个文本框显示两个翻译的相同之处。在报告中附上程序截图、运行结果和详细的文字说明。（40分）  2. 编写Java应用程序，实现“Java机考”的功能（“单机版Java简易机考程序”），包含单选题、多选题和判断题三种题型。  在主线程中创建一个Frame类型的窗口，在该窗口中再创建一个线程giveQuestion。  线程giveQuestion每隔20秒钟输出一个选择题（含A,B,C,D共4个选项，要求支持单选题和多选题，单选题用radio button，多选题用check box）或一个判断题（用radio button），选择题和判断题混合着给出；用户输入答案并按提交按钮提交结果（达到20秒自动提交结果）；程序判断用户输入的答案是否正确（如果错选或漏选均得零分），并实时显示当前题目的正确答案、已经给出的题目的数量（分别给出单选题数量、多选题数量和判断题数量）、用户答对的数量（分别给出单选题数量、多选题数量和判断题数量）、用户的成绩和用户答题所花的总的时间。  如此循环15次，包括随机选择的5个单选题（每题1分）、随机选择的5个多选题（每题2分）和随机选择的5个判断题（每题1分），结束测试时给出最终成绩。  题库应至少包含10个单选题、10个多选题和10个判断题。要求使用图形用户界面。  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。（40分）  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）本次实验课作业满分为100分，占总成绩的比例（待定）。  （2）本次实验课作业截至时间2023年12月13日（周三）21:59。  （3）报告正文：请在指定位置填写，本次实验**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；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”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（含抄袭往届同学的作业）。**  （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luyahui@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.使用JLabel、JTextArea、JButton等控件实现句子的中译英demo，该demo包含四个文本框，在第一个文本框输入一句中文，在第二个和第三个文本框显示该句的英文翻译（要求使用百度翻译API、有道翻译API或其他API中的两种），在第四个文本框显示两个翻译的相同之处。在报告中附上程序截图、运行结果和详细的文字说明。（40分）   1. 程序截图： TranslationDemo类：   这部分代码声明了五个私有成员变量，其中四个是用于在图形用户界面（GUI）中显示文本的文本区域（JTextArea），一个是用于触发翻译的按钮（JButton）：  inputTextArea：文本区域，用于接收用户输入的中文句子。  baiduTranslationTextArea：文本区域，用于显示百度翻译的结果。  youdaoTranslationTextArea：文本区域，用于显示有道翻译的结果。  commonTranslationTextArea：文本区域，用于显示两个翻译结果中的相同部分。translateButton：按钮，点击此按钮将触发翻译操作。  这部分代码创建了用户界面的主窗口和相关组件，设置了窗口的基本属性和布局。  创建 JFrame：  ·创建一个新的 JFrame 对象，窗口标题为 "简易翻译器"。  ·设置窗口大小为 1000x700 像素。  ·设置窗口关闭操作为退出程序。  ·设置窗口的布局管理器为垂直的 BoxLayout。  创建组件：  ·创建一个文本区域用于用户输入中文句子。  ·创建一个文本区域用于显示百度翻译的结果。  ·创建一个文本区域用于显示有道翻译的结果。  ·创建一个文本区域用于显示两个翻译结果的共同部分。  创建按钮和面板：  ·创建一个按钮，上面显示 "翻译"。  ·设置按钮的助记符为 Enter 键。  ·创建一个面板，使用右对齐的流式布局。  ·将翻译按钮添加到面板中。  Translate按钮的ActionListener:  ·添加一个监听器，监听 Translate 按钮的点击事件。  ·在点击按钮时，将执行 translate() 方法。  创建面板和添加组件:  ·使用 createPanel 方法创建面板，并将不同的文本区域添加到面板中。  ·设置 JFrame 的可见性为 true，使窗口显示在屏幕上。  这是createPanel方法：  ·使用 BorderLayout 布局管理器创建一个 JPanel。  ·创建一个 JLabel，并将其文本设置为传入的 labelText。  ·将标签水平对齐到左边。  ·将标签添加到面板的北部。  ·创建一个 JScrollPane，并将传入的文本区域 textArea 包装在其中。这样，如果文本内容太大而无法完全显示，用户可以通过滚动条查看全部内容。  将包含滚动条的文本区域添加到面板的中央。  点击翻译后，translate方法的实现：  获取输入文本:  使用 inputTextArea.getText() 获取用户在输入文本区域中输入的文本。  使用 trim() 方法去除首尾空格。  Baidu翻译:  ·使用 translateWithBaidu 方法进行百度翻译，将翻译结果保存在 baiduTranslation 变量中。  使用 baiduTranslationTextArea.setText(baiduTranslation) 将百度翻译结果设置到相应的文本区域（baiduTranslationTextArea）中。  Youdao翻译:  ·使用 translateWithYoudao 方法进行有道翻译，将翻译结果保存在 youdaoTranslation 变量中。  使用 youdaoTranslationTextArea.setText(youdaoTranslation) 将有道翻译结果设置到相应的文本区域（youdaoTranslationTextArea）中。  查找相同翻译:  ·使用 findCommonTranslation 方法查找百度和有道翻译结果的共同部分，将结果保存在 commonTranslation 变量中。  使用 commonTranslationTextArea.setText(commonTranslation) 将相同翻译结果设置到相应的文本区域（commonTranslationTextArea）中。  这三种翻译的实现：  有道翻译的实现：（从有道官网的下载的demo进行一点点修改）  APP\_KEY 和 APP\_SECRET 存储了在有道开放平台申请的应用ID和应用密钥，用于身份鉴权。  createRequestParams(text) 方法根据传入的文本 text 构造了请求参数 params。  AuthV3Util.addAuthParams(APP\_KEY, APP\_SECRET, params) 通过 AuthV3Util 工具类向请求参数中添加了身份鉴权相关的信息，以确保请求是合法的。  HttpUtil.doPost("https://openapi.youdao.com/api", null, params, "application/json") 发起了一个 POST 请求，向有道翻译 API 提交了请求参数。请求的目标地址是 "https://openapi.youdao.com/api"。  通过 result 获取了 API 的响应结果。在这里，使用 result2 将响应结果转换为 UTF-8 编码的字符串。  使用字符串处理方法截取了翻译结果的部分，从 "translation" 开始到 "mTerminalDict" 结束，并将截取的部分赋值给变量 en。  最终将截取得到的翻译结果 en 作为方法的返回值。  这是使用有道api需要的一些别的类，从有道官网直接下载下来即可，不需要更改。  AuthV3Util：  这个工具类的主要作用是生成有道翻译API请求的签名，以确保请求的合法性和安全性。签名计算的过程主要包括将请求相关的参数按照一定格式拼接，并进行 SHA-256 加密。  该方法用于添加鉴权相关参数到请求参数表中，包括应用ID (appKey)、随机值 (salt)、当前时间戳 (curtime)、签名类型 (signType) 和请求签名 (sign)。  获取请求参数中 "q" 或 "img" 对应的值，并将它们拼接成一个字符串。  生成随机的 salt，获取当前时间戳 curtime。  调用 calculateSign 方法计算签名。  将计算得到的签名和其他鉴权参数放入请求参数表中。  该方法用于计算签名，它采用 SHA-256 算法。  输入参数包括应用ID (appKey)、应用密钥 (appSecret)、请求内容 (q)、随机值 (salt) 和当前时间戳 (curtime)。  将这些参数按照特定格式拼接成一个字符串，并调用 encrypt 方法对这个字符串进行 SHA-256 加密。  该方法用于对输入字符串进行 SHA-256 加密。  使用 Java 的 MessageDigest 类来进行 SHA-256 加密，将加密后的结果转换为十六进制字符串。  该方法用于处理请求内容，确保请求内容长度不超过20个字符。  如果请求内容长度小于或等于20个字符，直接返回原始内容。  如果请求内容长度超过20个字符，取前10个字符、后10个字符，并在中间加上长度信息，返回处理后的字符串。  AuthV4Util，FileUtil这两个方法类好像并没有用上。  HttpUtil：用于发送 HTTP 请求。  WebSocketUtil：要用于实现 WebSocket 连接，发送和接收消息，并对接收到的消息进行简单的处理。  使用百度api进行翻译：同样，在百度官网下载demo改改即可  ·APP\_ID 和 SECURITY\_KEY，这是通过注册百度翻译开放平台并创建应用所获得的应用ID和密钥。（其实也就只是需要改这部分而已）  ·使用 TransApi 类的实例化对象 api，该类是通过在代码中提供 APP\_ID 和 SECURITY\_KEY 实现的。  ·调用 api.getTransResult(text, "auto", "en") 方法，传递待翻译的文本 (text)、源语言 (auto 表示自动检测语言) 和目标语言 (en 表示英语)。  ·获取翻译结果字符串 result。  ·通过字符串处理，找到翻译结果的开始位置，即 "dst" 字段。从 "dst" 字段的位置开始，提取翻译结果的内容，并返回。  还有别的一些下载的方法类：（无需修改）  TransApi：  TRANS\_API\_HOST 常量定义了百度翻译 API 的请求地址。  TransApi 类有一个带有两个参数的构造方法，用于初始化 appid 和 securityKey。  getTransResult 方法：  该方法用于发送翻译请求，接收参数包括待翻译的文本 query、源语言 from 和目标语言 to。  调用了 buildParams 方法构建请求参数。  调用 HttpGet.get 方法发起 GET 请求，传递翻译 API 的地址和构建好的请求参数。  返回翻译结果的字符串。  buildParams 方法：  该方法用于构建请求参数的 Map。  将待翻译的文本 query、源语言 from、目标语言 to、appid 加入参数中。  生成一个随机数作为 salt。  计算签名 sign，签名的原文是 appid + query + salt + securityKey，其中使用了 MD5.md5 方法对原文进行 MD5 加密。  HttpGet：提供了发送 HTTP GET 请求的方法，包含了一些基本的 SSL 支持和 URL 处理功能，方便进行简单的网络请求。  MD5：主要用来计算字符串的MD5值。  查看两种翻译的相同部分：  对两个输入字符串 translation1 和 translation2 中双引号括起来的内容进行匹配，找到两者之间相同的部分，并将这些相同部分组成一个新的字符串返回。  ·创建了一个 StringBuilder 对象，用于存储找到的相同部分。  ·使用正则表达式 \"([^\"]\*)\" 匹配双引号内的内容。  ·创建了两个 Matcher 对象 matcher1 和 matcher2，分别用于匹配 translation1 和 translation2 中的双引号括起来的内容。  ·双重循环比较：  第一个 while 循环遍历 translation1 中的所有匹配项。  第二个 while 循环在每个 translation1 的匹配项上，遍历 translation2 中的所有匹配项。  如果找到了相同的内容（match1.equals(match2)），则将该内容添加到 result 中。  ·Matcher的重置：  在第一个 while 循环中，每次找到 translation1 中的一个匹配项后，都需要将 matcher2 重置，以便从头开始匹配 translation2。  ·结果的返回：  如果找到了相同的部分，将结果以字符串形式返回，并去除末尾可能存在的空格。  如果没有找到相同的部分，则返回 "没有相同部分"。  运行结果：  界面展示:  翻译效果：  对于输入的中文，可以正确的翻译成英文。  可以输出有道翻译和百度翻译的相同翻译。  2.编写Java应用程序，实现“Java机考”的功能（“单机版Java简易机考程序”）（40分）  Question类：  Question 类具有以下属性：  text: 问题的文本描述。  correctAnswer: 问题的正确答案。  构造函数：  public Question(String text, String correctAnswer): 通过传入问题文本和正确答案来创建问题实例。  方法：  public String getText(): 获取问题的文本描述。  public String getCorrectAnswer(): 获取问题的正确答案。  public boolean isCounted(): 检查问题是否已被计数。  public void setCounted(boolean counted): 设置问题是否已被计数。  这个类的目的是提供一个通用的问题模型，可以通过继承它来创建不同类型的问题，例如单选题、多选题等。  这是一个继承自Question类的SingleChoiceQuestion类，用于表示单选题。SingleChoiceQuestion 类扩展了 Question 类，具有相同的属性和方法。  新增属性：  options: 一个包含单选题选项的列表。  构造函数：  public SingleChoiceQuestion(String text, List<String> options, String correctAnswer): 通过传入问题文本、选项列表和正确答案来创建单选题实例。  新增方法：  public List<String> getOptions(): 获取单选题的选项列表。  这是一个继承自Question类的TFQuestion类，用于表示判断题。  TFQuestion 类扩展了 Question 类，具有相同的属性和方法。  新增属性：  options: 一个包含判断题选项的列表。  构造函数：  public TFQuestion(String text, List<String> options, String correctAnswer): 通过传入问题文本、选项列表和正确答案来创建判断题实例。  新增方法：  public List<String> getOptions(): 获取判断题的选项列表。  这是一个继承自Question类的MultipleChoiceQuestion类，用于表示多选题。MultipleChoiceQuestion 类扩展了 Question 类，具有相同的属性和方法。  新增属性：  options: 一个包含多选题选项的列表。  构造函数：  public MultipleChoiceQuestion(String text, List<String> options, String correctAnswer): 通过传入问题文本、选项列表和正确答案来创建多选题实例。  新增方法：  public List<String> getOptions(): 获取多选题的选项列表。  public boolean checkAnswer(List<String> selectedOptions): 检查给定的选项是否是正确答案的子集。  private List<String> getCorrectOptions(): 获取正确答案的选项列表（通过逗号分隔）。  public List<String> getCorrectOptions2(): 获取修剪过的正确答案选项列表（去除空格）。  公共类JavaExamGUI：  一些常量和成员变量的定义  TOTAL\_QUESTIONS 定义了总问题数，限制了用户需要回答的总问题数量为15。  SINGLE\_CHOICE\_SCORE 定义了单选题的分数，每个正确回答的单选题得1分。  MULTIPLE\_CHOICE\_SCORE 定义了多选题的分数，每个正确回答的多选题得2分。  TRUE\_FALSE\_SCORE定义了判断题的分数，每个正确回答的判断题得1分。  questionBank一个存储问题的列表，包含了各种不同类型的问题。  questionIndex当前问题的索引，用于跟踪当前要显示的问题。  correctAnswers用户已经回答正确的总问题数，这是一个计数器。  totalQuestionsGiven用户已经回答的总问题数，包括正确和错误的回答。  singleChoiceQuestionsGiven 用户已经回答的单选题数。  multipleChoiceQuestionsGiven用户已经回答的多选题数。  private int trueFalseQuestionsGiven 用户已经回答的判断题数，初始值为-1，表示还未开始回答判断题。  correctSingleChoiceAnswers用户已经回答正确的单选题数。  correctMultipleChoiceAnswers用户已经回答正确的多选题数。  correctTrueFalseAnswers用户已经回答正确的判断题数。  totalTimeSpent用户总共花费的时间，以秒为单位。  这是Java程序的main方法，它使用了Swing的SwingUtilities.invokeLater方法来确保在事件分发线程（Event Dispatch Thread）中执行GUI相关的操作。  这是JavaExamGUI类的构造函数。  调用generateQuestionBank方法生成题库，并将其赋值给questionBank成员变量。这个题库包含了不同类型的问题。  设置GUI窗口的标题为"Java Exam"。  设置GUI窗口的大小为宽1000像素、高700像素。  设置当用户关闭窗口时默认的操作是退出应用程序。  设置窗口相对于屏幕的位置为居中。  调用initializeUI方法，用于初始化用户界面。  创建一个新线程，该线程运行giveQuestion方法。这个方法负责定时提问问题。  这是我自己写的十五个题目。  ·这是initializeUI方法，用于初始化用户界面。  ·设置GUI的布局管理器为BoxLayout，并将其方向设置为纵向（Y\_AXIS）。这意味着后续添加的组件将按垂直方向排列。  ·创建一个标签用于显示问题，设置标签的文本为空字符串，水平对齐方式为居中。  ·将问题标签添加到GUI中。  ·创建一个面板用于放置问题选项，使用GridLayout，每行一个元素，列数不限制。这将确保选项以垂直方向的单列布局。  ·将选项面板添加到GUI中。  ·创建一个提交按钮，并设置其文本为"Submit"。  ·为提交按钮添加一个动作监听器，指定在按钮被点击时调用submitAnswer方法。  ·创建一个新的面板，使用FlowLayout，并将其水平对齐方式设置为左对齐。  ·将提交按钮添加到左对齐的面板中。  ·将包含提交按钮的面板添加到GUI中。  ·创建一个标签用于显示结果信息，设置标签的文本为空字符串，水平对齐方式为居中。  ·将结果标签添加到GUI中。  这是giveQuestion方法。  ·创建一个新的Timer对象，用于执行定时任务。  ·使用schedule方法安排定时任务。这里的定时任务是一个匿名内部类，实现了TimerTask接口。这个任务在每次执行时会执行run方法。  ·使用SwingUtilities.invokeLater确保任务在事件调度线程上执行，以避免与Swing组件的更新冲突。  ·检查是否还有未提问的问题。  ·在每次定时任务执行时自动提交答案。  ·显示下一个问题。  ·更新问题索引，以便下次显示下一个问题。  ·增加已提问问题的计数。  ·如果已经提问了所有问题，则执行以下操作：  ·显示最终结果。  ·取消定时器，停止定时任务的执行。  这是submitAnswer方法，它用于提交用户的答案并处理评估和显示结果。  获取当前问题，由于questionIndex是从0开始的，使用(questionIndex - 1 + questionBank.size()) % questionBank.size()确保在第一次调用时获取正确的问题。  根据问题的类型分别调用不同的答案评估方法。  如果问题是单选题，则调用单选题答案评估方法。  如果问题是多选题，则调用多选题答案评估方法。  如果问题是判断题，则调用判断题答案评估方法。  这是displayQuestion方法，它用于在界面上显示问题的文本和选项。设置问题文本到界面上的标签。  清空选项面板，以便填充新的选项。  根据问题的类型调用不同的创建选项方法。  如果问题是单选题，则调用创建单选题选项的方法。  如果问题是多选题，则调用创建多选题选项的方法。  如果问题是判断题，则调用创建判断题选项的方法。  重新验证UI，确保新的组件布局生效。  重新绘制UI，以便立即显示更新后的界面。  这段代码是createSingleChoiceOptions方法的实现，该方法用于创建单选题的选项。 创建一个ButtonGroup对象，用于将单选按钮（JRadioButton）分组，确保用户只能选择其中一个选项。  遍历单选题的选项列表。  对于每个选项，创建一个单选按钮，并使用选项的文本进行初始化。  将每个单选按钮添加到ButtonGroup中，以便进行分组。  将单选按钮添加到选项面板，以便它们显示在用户界面上。  这段代码是createMultipleChoiceOptions方法的实现，该方法用于创建多选题的选项。  遍历选项列表： 使用增强的for循环，遍历多选题的选项列表。  创建复选框： 对于每个选项，创建一个复选框（JCheckBox），并使用选项的文本进行初始化。  添加到选项面板： 将每个复选框添加到选项面板，以便它们在用户界面上显示出来。  逻辑和createSingleChoiceOptions方法一致。  evaluateSingleChoiceAnswer方法：  通过getSelectedRadioButton方法获取用户选择的单选按钮。  如果用户选择了单选按钮且其文本的第一个字符与问题的正确答案相符，则将SINGLE\_CHOICE\_SCORE（单选题的分数）添加到correctAnswers中。  evaluateTFAnswer方法：  通过getSelectedRadioButton方法获取用户选择的单选按钮。  如果用户选择了单选按钮且其文本的第一个字符与问题的正确答案相符，则将SINGLE\_CHOICE\_SCORE（单选题的分数）添加到correctAnswers中。  evaluateMultipleChoiceAnswer方法：  通过getSelectedCheckBoxes方法获取用户选择的复选框的文本列表。  检查用户选择的复选框是否与问题的正确答案列表相匹配，如果匹配，则将MULTIPLE\_CHOICE\_SCORE（多选题的分数）添加到correctAnswers中。  getSelectedRadioButton方法：  遍历optionsPanel中的所有组件。  对于每个组件，检查是否是JRadioButton。  如果找到选中的单选按钮，返回该按钮；否则，返回null。  getSelectedCheckBoxes方法：  遍历optionsPanel中的所有组件。  对于每个组件，检查是否是JCheckBox。  如果找到选中的复选框，将其文本添加到selectedOptions列表中。  displayResult 方法负责在用户回答问题后更新界面，显示正确答案、用户答案、得分以及其他统计信息。  创建一个 StringBuilder 对象，用于构建结果文本。  显示正确答案： 将问题的正确答案附加到结果文本中。  根据问题类型更新结果文本：  ·单选题（SingleChoiceQuestion）：  追加用户的单选题答案。  增加单选题数。  检查并更新正确单选题答案数。  ·多选题（MultipleChoiceQuestion）：  追加用户的多选题答案。  增加多选题数。  检查并更新正确多选题答案数。  ·判断-错题（TFQuestion）：  追加用户的判断-错题答案。  增加判断-错题数。  检查并更新正确判断-错题答案数。  追加其他统计信息：  当前得分。  已回答问题数。  单选题、多选题、判断-错题的数量。  正确的单选题、多选题、判断-错题的数量。  总共花费的时间。  将结果文本设置到 resultLabel： 将构建好的结果文本应用到界面的 resultLabel 上。  刷新 UI： 使用 revalidate() 和 repaint() 方法刷新用户界面，确保更新后的结果能够正确显示。  输出最后的结果。  运行结果：  可以实时显示当前题目的正确答案、已经给出的题目的数量（分别给出单选题数量、多选题数量和判断题数量）、用户答对的数量（分别给出单选题数量、多选题数量和判断题数量）、用户的成绩和用户答题所花的总的时间。  并在答题结束后给出最终成绩。  ++++++++++++++++++++++++++++++++++++++++++++++++++++++  **其他（例如感想、建议等等）。**  本次实验是非常复杂的一次实验，我用了非常多的时间去调试修改我的代码。  第一个实验的难点是如何接入有道翻译和百度翻译的api，这是我琢磨了许久才完成的。需要分别到百度和有道官网注册账号，开通应用服务，获取个人的密钥和id（所以也请不要过度使用这个翻译功能，用多了是要给钱的），然后去查看官方的开发者文档，幸好官方提供了java开发的demo，不然就凭我自己恐怕是实现不了这个api接入的功能，所以我只需要对官方提供的api进行一些修改就可以使用了，百度api是更好改的一个，有道的api就麻烦一点。  第二道题目就更加麻烦了，不仅要用到GUI，还要用到线程以及许多没有学过的方法，然后看似是一个简单的考试app，但是所需要的代码，要考虑的bug却是非常非常多，比如多选题选对但是不加分的问题就困扰了我非常久，还有实时刷新题目的功能也是让我无从下手，只能上网查找各种资料，方法来解决。然后就是代码量挺大的，函数很多，所以组织也是很复杂(我组织的还是比较乱）。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  2021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、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。