

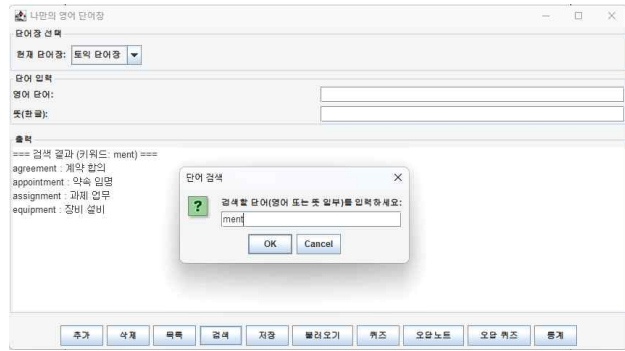
### 3. 활동일지(주간)

기본정보	보고주차	4주차 ( 12월 10일 ~ 12월 16일 )
	작성자	이강민, 장정원, 신상현, 이민제
	제출일	2025.12.13.
활동내용	주요 진행 사항	<p>GitHub( <a href="https://github.com/hjus36/java-vocab-project">https://github.com/hjus36/java-vocab-project</a> )</p> <p>3주차에 프로젝트가 마무리되는 것으로 판단했으나 실제 일정은 4주차까지였음을 확인하였다. 이에 따라 총 5개의 기능을 추가하여 프로그램의 완성도를 더욱 높였다.</p> <ol style="list-style-type: none"> <li>단어 검색 기능 추가 <ul style="list-style-type: none"> <li>■ WordBook에 검색 메서드(search) 구현</li> <li>■ 단어와 뜻에 대해 부분 일치 검색이 가능하도록 처리</li> <li>■ MainFrame에 검색 버튼을 추가하고, 검색 결과를 출력 영역에 표시되도록 연동</li> </ul> </li> <li>퀴즈 종료 요약 기능 추가 <ul style="list-style-type: none"> <li>■ 퀴즈 종료 시 이번 세션 동안 푼 문제 수, 정답 수, 정답률을 팝업으로 표시</li> <li>■ QuizFrame에서 퀴즈 시작 시점의 통계를 저장하고, 종료 시 비교하여 세션 요약을 계산하도록 구현</li> <li>■ 닫기 버튼 및 창 닫기(X) 모두 요약이 출력되도록 WindowListener를 통해 처리</li> </ul> </li> <li>오답노트 기록 기능 구현 <ul style="list-style-type: none"> <li>■ 퀴즈에서 오답인 경우 해당 단어를 오답 리스트에 자동으로 기록하도록 QuizManager 로직 추가</li> <li>■ 같은 단어가 여러 번 틀려도 중복해서 쌓이지 않도록 오답 리스트 중복 체크 처리</li> <li>■ MainFrame에서 오답노트 보기 기능을 통해 현재까지의 오답 단어 목록을 출력할 수 있도록 연동</li> </ul> </li> <li>오답노트 파일 저장 기능 추가 <ul style="list-style-type: none"> <li>■ BookType별로 오답노트 저장용 파일 경로를 별도로 정의</li> <li>■ FileManager에 오답 리스트를 저장하는 saveWrongNotes 메서드 구현</li> <li>■ 퀴즈 종료 시 세션 요약을 보여준 뒤, 현재 오답 리스트를 자동으로 파일에 저장되도록 QuizFrame과 연동</li> </ul> </li> <li>오답 퀴즈 기능 구현 <ul style="list-style-type: none"> <li>■ 기존 일반 퀴즈와 별도로 오답 리스트에서 문제를 출제하는 오답 퀴즈 기능 추가</li> <li>■ QuizFrame 생성 시 일반 모드/오답 모드를 구분하는 boolean 필드를 정의하여, 오답 모드에서는 오답 리스트에서만 랜덤 출제되도록 처리</li> <li>■ 오답 리스트가 비어 있을 경우 오답 퀴즈가 실행되지 않도록 사전 검사 후, 안내 메시지를 출력하도록 예외 처리</li> </ul> </li> </ol>

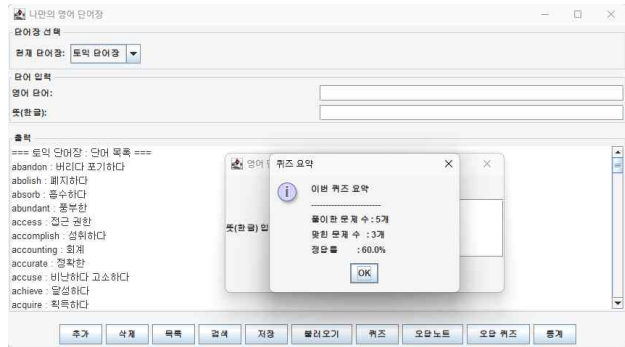
활동내용

주요 진행 사항

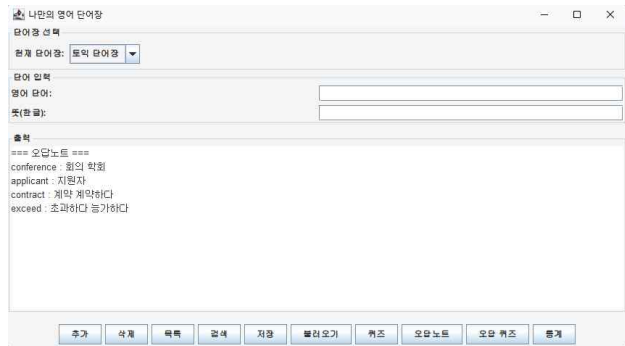
## 1. 단어 검색 기능



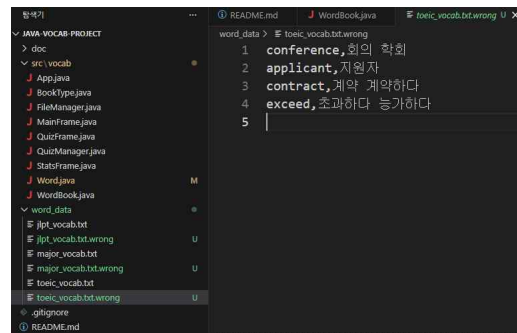
## 2. 퀴즈 종료 요약 기능



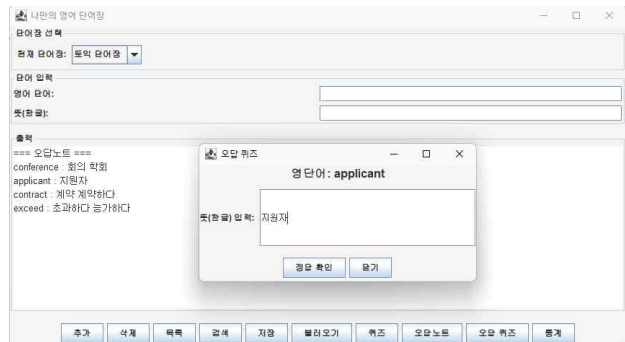
## 3. 오답노트 기록 기능



## 4. 오답노트 파일 저장 기능



## 5. 오답 퀴즈 기능



다음 주 계획

최종 보고서 및 발표 자료 준비

활동내용	문제점 및 해결 방안	1. 오답 퀴즈 실행 시 오답 리스트가 비어 있을 경우 프로그램 흐름이 비정상적으로 종료되는 문제				
		<table><tr><td>원인 분석</td><td>오답 퀴즈는 오답 리스트에서 단어를 랜덤으로 추출하는 방식이지만, 리스트가 비어 있는 상태에서 실행될 경우 random.nextInt(0)이 호출되어 예외가 발생하였다. 또한 MainFrame에서 사전 확인이 없어 빈 리스트 상태에서도 퀴즈 창이 열리는 문제가 있었다.</td></tr><tr><td>해결 방안</td><td><pre>// 오답퀴즈 열기 private void openWrongQuiz() {     if (!quizManager.hasWrongNotes()) {         JOptionPane.showMessageDialog(             this,             message: "오답노트가 비어 있습니다.\n먼저 퀴즈를 풀어서 오답을 만들어 주세요.",             title: "오답 퀴즈 시작 불가",             JOptionPane.INFORMATION_MESSAGE         );         return;     }     new QuizFrame(book, quizManager, fileManager, currentType, wrongOnly: true); }</pre><div><pre>// 오답 리스트에서 랜덤 단어 하나 반환 public Word getRandomWrongWord() {     if (wrongList.isEmpty()) return null;     int idx = random.nextInt(wrongList.size());     return wrongList.get(idx); }</pre><pre>public boolean hasWrongNotes() {     return !wrongList.isEmpty(); }</pre></div><p>-&gt; MainFrame에서 오답 퀴즈 실행 전 quizManager.hasWrongNotes()로 리스트 비어 있음 여부를 먼저 검사하였다.</p><p>-&gt; QuizManager 내부에서도 wrongList가 비어 있을 경우 문제를 반환하지 않도록 하였다.</p></td></tr></table>	원인 분석	오답 퀴즈는 오답 리스트에서 단어를 랜덤으로 추출하는 방식이지만, 리스트가 비어 있는 상태에서 실행될 경우 random.nextInt(0)이 호출되어 예외가 발생하였다. 또한 MainFrame에서 사전 확인이 없어 빈 리스트 상태에서도 퀴즈 창이 열리는 문제가 있었다.	해결 방안	<pre>// 오답퀴즈 열기 private void openWrongQuiz() {     if (!quizManager.hasWrongNotes()) {         JOptionPane.showMessageDialog(             this,             message: "오답노트가 비어 있습니다.\n먼저 퀴즈를 풀어서 오답을 만들어 주세요.",             title: "오답 퀴즈 시작 불가",             JOptionPane.INFORMATION_MESSAGE         );         return;     }     new QuizFrame(book, quizManager, fileManager, currentType, wrongOnly: true); }</pre> <div><pre>// 오답 리스트에서 랜덤 단어 하나 반환 public Word getRandomWrongWord() {     if (wrongList.isEmpty()) return null;     int idx = random.nextInt(wrongList.size());     return wrongList.get(idx); }</pre><pre>public boolean hasWrongNotes() {     return !wrongList.isEmpty(); }</pre></div> <p>-&gt; MainFrame에서 오답 퀴즈 실행 전 quizManager.hasWrongNotes()로 리스트 비어 있음 여부를 먼저 검사하였다.</p> <p>-&gt; QuizManager 내부에서도 wrongList가 비어 있을 경우 문제를 반환하지 않도록 하였다.</p>
원인 분석	오답 퀴즈는 오답 리스트에서 단어를 랜덤으로 추출하는 방식이지만, 리스트가 비어 있는 상태에서 실행될 경우 random.nextInt(0)이 호출되어 예외가 발생하였다. 또한 MainFrame에서 사전 확인이 없어 빈 리스트 상태에서도 퀴즈 창이 열리는 문제가 있었다.					
해결 방안	<pre>// 오답퀴즈 열기 private void openWrongQuiz() {     if (!quizManager.hasWrongNotes()) {         JOptionPane.showMessageDialog(             this,             message: "오답노트가 비어 있습니다.\n먼저 퀴즈를 풀어서 오답을 만들어 주세요.",             title: "오답 퀴즈 시작 불가",             JOptionPane.INFORMATION_MESSAGE         );         return;     }     new QuizFrame(book, quizManager, fileManager, currentType, wrongOnly: true); }</pre> <div><pre>// 오답 리스트에서 랜덤 단어 하나 반환 public Word getRandomWrongWord() {     if (wrongList.isEmpty()) return null;     int idx = random.nextInt(wrongList.size());     return wrongList.get(idx); }</pre><pre>public boolean hasWrongNotes() {     return !wrongList.isEmpty(); }</pre></div> <p>-&gt; MainFrame에서 오답 퀴즈 실행 전 quizManager.hasWrongNotes()로 리스트 비어 있음 여부를 먼저 검사하였다.</p> <p>-&gt; QuizManager 내부에서도 wrongList가 비어 있을 경우 문제를 반환하지 않도록 하였다.</p>					
2. 퀴즈 종료 요약 기능에서 첫 문제 이후 종료 시 통계 값이 잘못 계산되는 문제						
		<table><tr><td>원인 분석</td><td>퀴즈 종료 요약은 세션 시작 시 통계를 기록하고 종료 시점과 비교해 계산하는 방식이다. 그러나 QuizFrame을 여러 번 열고 닫을 때 시작 시점 값이 올바르게 갱신되지 않아 이전 세션의 누적값이 포함되는 문제가 발생하였다.</td></tr><tr><td>해결 방안</td><td><pre>// 생성자 public QuizFrame(WordBook book, QuizManager quizManager, FileManager fileManager,     BookType currentType, boolean wrongOnly) {     this.book = book;     this.quizManager = quizManager;     this.fileManager = fileManager;     this.currentType = currentType;     this.wrongOnly = wrongOnly;      // 세션 시작 시점 통계 기록     this.startTotalQuestions = quizManager.getTotalQuestions();     this.startCorrectAnswers = quizManager.getCorrectAnswers(); }</pre><pre>// 퀴즈 세션 요약 보여주기 private void showQuizSummary() {     int endTotal = quizManager.getTotalQuestions();     int endCorrect = quizManager.getCorrectAnswers();      int sessionTotal = endTotal - startTotalQuestions;     int sessionCorrect = endCorrect - startCorrectAnswers; }</pre><p>-&gt; QuizFrame 생성 시점에 startTotalQuestions, startCorrectAnswers 값을 항상 새로 저장하도록 구조를 변경하였다.</p><p>-&gt; 종료 시에는 현재 누적값과 비교하여 이번 세션 값만 계산하도록 구조를 변경하였다.</p></td></tr></table>	원인 분석	퀴즈 종료 요약은 세션 시작 시 통계를 기록하고 종료 시점과 비교해 계산하는 방식이다. 그러나 QuizFrame을 여러 번 열고 닫을 때 시작 시점 값이 올바르게 갱신되지 않아 이전 세션의 누적값이 포함되는 문제가 발생하였다.	해결 방안	<pre>// 생성자 public QuizFrame(WordBook book, QuizManager quizManager, FileManager fileManager,     BookType currentType, boolean wrongOnly) {     this.book = book;     this.quizManager = quizManager;     this.fileManager = fileManager;     this.currentType = currentType;     this.wrongOnly = wrongOnly;      // 세션 시작 시점 통계 기록     this.startTotalQuestions = quizManager.getTotalQuestions();     this.startCorrectAnswers = quizManager.getCorrectAnswers(); }</pre> <pre>// 퀴즈 세션 요약 보여주기 private void showQuizSummary() {     int endTotal = quizManager.getTotalQuestions();     int endCorrect = quizManager.getCorrectAnswers();      int sessionTotal = endTotal - startTotalQuestions;     int sessionCorrect = endCorrect - startCorrectAnswers; }</pre> <p>-&gt; QuizFrame 생성 시점에 startTotalQuestions, startCorrectAnswers 값을 항상 새로 저장하도록 구조를 변경하였다.</p> <p>-&gt; 종료 시에는 현재 누적값과 비교하여 이번 세션 값만 계산하도록 구조를 변경하였다.</p>
원인 분석	퀴즈 종료 요약은 세션 시작 시 통계를 기록하고 종료 시점과 비교해 계산하는 방식이다. 그러나 QuizFrame을 여러 번 열고 닫을 때 시작 시점 값이 올바르게 갱신되지 않아 이전 세션의 누적값이 포함되는 문제가 발생하였다.					
해결 방안	<pre>// 생성자 public QuizFrame(WordBook book, QuizManager quizManager, FileManager fileManager,     BookType currentType, boolean wrongOnly) {     this.book = book;     this.quizManager = quizManager;     this.fileManager = fileManager;     this.currentType = currentType;     this.wrongOnly = wrongOnly;      // 세션 시작 시점 통계 기록     this.startTotalQuestions = quizManager.getTotalQuestions();     this.startCorrectAnswers = quizManager.getCorrectAnswers(); }</pre> <pre>// 퀴즈 세션 요약 보여주기 private void showQuizSummary() {     int endTotal = quizManager.getTotalQuestions();     int endCorrect = quizManager.getCorrectAnswers();      int sessionTotal = endTotal - startTotalQuestions;     int sessionCorrect = endCorrect - startCorrectAnswers; }</pre> <p>-&gt; QuizFrame 생성 시점에 startTotalQuestions, startCorrectAnswers 값을 항상 새로 저장하도록 구조를 변경하였다.</p> <p>-&gt; 종료 시에는 현재 누적값과 비교하여 이번 세션 값만 계산하도록 구조를 변경하였다.</p>					

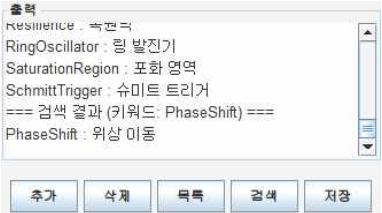
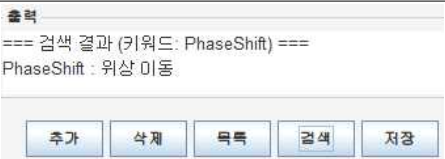
## 활동내용

## 문제점 및 해결 방안

### 3. 오답노트 중복 기록 문제

원인 분석	같은 단어를 여러 번 틀릴 경우, 오답 리스트에 동일한 단어가 중복으로 계속 추가되었다. 리스트 형태로 저장되기 때문에 별도의 중복 체크가 없어 발생한 문제였다.
해결 방안	<pre>// 정답 체크 및 통계 업데이트 public boolean checkAnswer(Word questionWord, String userAnswer) {     if (questionWord == null) return false;      String correct = questionWord.getMeaning();     String answer = userAnswer == null ? "" : userAnswer.trim();     boolean isCorrect = correct.trim().equals(answer);      totalQuestions++;     if (isCorrect) {         correctAnswers++;     } else {         // 오답이면 오답리스트에 등록 (중복 방지)         if (!wrongList.contains(questionWord)) {             wrongList.add(questionWord);         }     }      return isCorrect; }</pre> <p>-&gt; QuizManager에서 오답 추가 시 wrongList.contains(word)로 중복 여부를 먼저 검사한 후 추가하도록 수정하였다.</p>

### 4. 검색 후 출력이 이전 내용과 섞여 보이는 문제

원인 분석	<p>검색 기능 실행 시 outputArea.append()로 결과를 출력하면서, 기존에 출력되어 있던 단어 목록 및 메시지가 그대로 남아 있었다. 이로 인해 검색 결과가 이전 내용과 섞여 보이는 문제가 발생하였다.</p>  <pre>for (Word w : results) {     sb.append(w.toString()).append(str: "\n"); }  outputArea.append(sb.toString());</pre>
해결 방안	 <pre>for (Word w : results) {     sb.append(w.toString()).append(str: "\n"); }  outputArea.setText(sb.toString());</pre> <p>-&gt; 출력 방식을 append()에서 setText()로 변경하여 출력 영역을 초기화한 뒤 새로운 검색 결과만 표시하도록 하였다.</p>

팀원 기여도 평가	이름	계획 역할	수행 내용	달성도%
	이강민	<p>〈프로젝트 총괄 / GUI 개발〉</p> <ul style="list-style-type: none"> <li>- 전체 프로그램 구조 설계 및 GUI 통합</li> <li>- MainFrame 화면 구성 및 이벤트 처리</li> <li>- QuizFrame · StatsFrame 연동 제어</li> <li>- 사용자 입력 검증 및 예외 처리 담당</li> </ul>	<ul style="list-style-type: none"> <li>- 검색 버튼 · 오답노트 · 오답퀴즈 버튼 추가 및 이벤트 처리</li> <li>- 검색 기능 연동 및 출력 영역 표시 방식 개선</li> <li>- 오답퀴즈 기능을 GUI에 연결하고 동작 흐름 점검</li> <li>- 퀴즈 종료 요약 팝업 동작 확인 및 화면 구성 정리</li> <li>- 전체 GUI 흐름 안정화 및 오류 없는 동작 검증</li> </ul>	100%
	장정원	<p>〈데이터 로직 개발 / 기능 통합〉</p> <ul style="list-style-type: none"> <li>- WordBook 클래스 구현 (단어 등록, 검색, 통계 기능)</li> <li>- Word 클래스 정의 및 데이터 구조 설계</li> <li>- 중복 처리 및 통계 알고리즘 작성</li> <li>- 기능별 단위 테스트 수행</li> </ul>	<ul style="list-style-type: none"> <li>- WordBook에 부분 검색 기능(search) 구현</li> <li>- 검색 결과 전처리(공백 · 대소문자 등) 로직 보완</li> <li>- 오답 리스트 구조 점검 및 단어 데이터 처리 검증</li> <li>- 검색 기능과 단어장 데이터 흐름 전체 점검</li> <li>- 기능별 동작 테스트 수행 및 결과 정리</li> </ul>	100%
	신상현	<p>〈퀴즈 기능 / 문서화 및 발표 자료〉</p> <ul style="list-style-type: none"> <li>- QuizFrame 클래스 구현 (랜덤 문제 출제 및 정답 판별)</li> <li>- 퀴즈 결과 저장 및 통계 기능 보조 개발</li> <li>- 전체 프로젝트 테스트 케이스 작성</li> <li>- 프로젝트 보고서 작성</li> <li>- 발표 자료 제작 및 시연 영상 준비</li> </ul>	<ul style="list-style-type: none"> <li>- QuizFrame에 퀴즈 종료 요약 기능 구현</li> <li>- QuizManager에 오답노트 기록 및 오답퀴즈 모드 기능 추가</li> <li>- 오답 전용 퀴즈 흐름 정비 및 화면 동작 개선</li> <li>- 퀴즈 관련 기능 테스트 및 점검</li> <li>- 보고서 내용 정리 및 발표 자료 준비 진행</li> </ul>	100%
	이민제	<p>〈파일 입출력 및 예외 처리 담당〉</p> <ul style="list-style-type: none"> <li>- FileManager 파일 저장/불러오기 개발</li> <li>- 예외 처리 로직 작성</li> <li>- 인코딩 문제 해결</li> </ul>	<ul style="list-style-type: none"> <li>- BookType에 오답노트 파일 경로 추가 (getWrongPath) 작성</li> <li>- FileManager에 오답노트 저장 기능 (saveWrongNotes) 구현</li> <li>- 퀴즈 종료 시 오답 파일 자동 저장 기능 연동</li> <li>- word_data 디렉토리 및 파일 생성 · 저장 동작 점검</li> <li>- 단어장 및 오답 파일 입출력 테스트 수행</li> </ul>	100%