

3. 활동일지(주간)

기본정보	보고주차	2주차 (11월 26일 ~ 12월 02일)
	작성자	이강민, 장정원, 신상현, 이민제
	제출일	2025.12.02.
활동내용	GitHub(https://github.com/hjus36/java-vocab-project)	
	<p>1. 단어 기능 구현 (Word, WordBook)</p> <ul style="list-style-type: none"> ■ Word 클래스 완성 (필드, 생성자, getter, toString) ■ WordBook: add, delete, find, getAll, size 메서드 모두 구현 ■ 검색 및 삭제 시 대소문자 무시 처리 <p>2. 파일 입출력 기능 개발</p> <ul style="list-style-type: none"> ■ FileManager.save() / load() 기능 완성 ■ UTF-8 인코딩 적용 ■ 파일 미존재(FileNotFound) 예외 처리 ■ 저장 파일 구조(english,korean) 통일 ■ load()시 기존 데이터 초기화 후 새로 저장하도록 개선 <p>3. GUI 화면 기본 구성</p> <ul style="list-style-type: none"> ■ MainFrame 레이아웃 구성(BorderLayout + GridLayout) ■ 단어 입력창(JTextField), 출력창(JTextArea), 버튼 패널 구현 ■ 단어 추가/삭제/목록 버튼 이벤트 처리 완료 ■ 저장/불러오기 버튼 이벤트 처리 완료 ■ 퀴즈 버튼을 통한 QuizFrame 연동 구현 ■ 통계 버튼을 통한 StatsFrame 연동 구현(StatsFrame 미구현 상태) <p>4. QuizFrame 구현</p> <ul style="list-style-type: none"> ■ 랜덤 단어 출제 ■ 정답 입력 및 판별 기능 구현 ■ 오답 시 정답 표시 ■ GUI 분리창(JDialog/Frame) 구현 	

문제점 및 해결 방안	<p>1. 파일 저장 시 한글이 깨져 보이는 현상 발생</p> <table border="1"> <tr> <td>원인 분석</td><td>텍스트 파일 저장 시 인코딩을 지정하지 않으면 운영체제 기본 인코딩과 불일치하여 한글 손상이 발생할 수 있음.</td></tr> <tr> <td>해결 방안</td><td> <pre>OutputStreamWriter osw = new OutputStreamWriter(new FileOutputStream(FILE_PATH), charsetName: "UTF-8"); -> OutputStreamWriter을 통해 UTF-8을 지정하도록 코드 수정</pre> </td></tr> </table> <p>2. FileNotFoundException, IOException 예외 발생</p> <table border="1"> <tr> <td>원인 분석</td><td>프로그램 최초 실행시 Word.txt 파일이 존재하지 않는 상태에서 FileInputStream이 호출되며 FileNotFoundException이 발생함. 또한 읽기/쓰기 도중 스트림이 정상적으로 닫히지 않거나 파일 접근 중 예상치 못한 출력력 문제(IOError)가 발생할 경우 IOException이 발생.</td></tr> <tr> <td>해결 방안</td><td> <pre>} catch (FileNotFoundException e) { // 파일이 없을 경우 System.out.println(x: "저장된 파일이 없습니다."); } catch (IOException e) { // 읽기 중 오류 System.out.println(x: "불러오기 오류"); } -> try-catch 문을 이용해 FileNotFoundException과 IOException을 처리하여 안내 메시지만 출력하고 프로그램은 정상 실행되도록 구성함.</pre> </td></tr> </table> <p>3. NullPointerException 예외 발생</p> <table border="1"> <tr> <td>원인 분석</td><td>단어 목록이 비어 있는 상태에서 퀴즈 기능을 실행하거나 getRandomWord() 수행 시 WordBook.size()가 0임에도 랜덤 인덱스에 접근하여 null 값을 참조하는 상황이 발생함.</td></tr> <tr> <td>해결 방안</td><td> <pre>if (book.size() == 0) { JOptionPane.showMessageDialog(parentComponent: null, message: "단어가 없습니다."); dispose(); // 단어 없으면 창 닫음 return; } // 랜덤 단어 반환 (없으면 null) public Word getRandomWord() { int n = book.size(); if (n == 0) return null; int idx = rand.nextInt(n); return book.getAll().get(idx); } -> 퀴즈 실행 시 book.size()가 0인지 사전에 검사하여 단어가 없을 경우 퀴즈 창을 띄우지 않고 사용자에게 안내 메시지를 출력함. 또한 getRandomWord()에서 null 반환 처리를 하고 GUI(Frame) 측에서도 null 체크를 거쳐 안정적으로 제어하도록 개선함.</pre> </td></tr> </table>		원인 분석	텍스트 파일 저장 시 인코딩을 지정하지 않으면 운영체제 기본 인코딩과 불일치하여 한글 손상이 발생할 수 있음.	해결 방안	<pre>OutputStreamWriter osw = new OutputStreamWriter(new FileOutputStream(FILE_PATH), charsetName: "UTF-8"); -> OutputStreamWriter을 통해 UTF-8을 지정하도록 코드 수정</pre>	원인 분석	프로그램 최초 실행시 Word.txt 파일이 존재하지 않는 상태에서 FileInputStream이 호출되며 FileNotFoundException이 발생함. 또한 읽기/쓰기 도중 스트림이 정상적으로 닫히지 않거나 파일 접근 중 예상치 못한 출력력 문제(IOError)가 발생할 경우 IOException이 발생.	해결 방안	<pre>} catch (FileNotFoundException e) { // 파일이 없을 경우 System.out.println(x: "저장된 파일이 없습니다."); } catch (IOException e) { // 읽기 중 오류 System.out.println(x: "불러오기 오류"); } -> try-catch 문을 이용해 FileNotFoundException과 IOException을 처리하여 안내 메시지만 출력하고 프로그램은 정상 실행되도록 구성함.</pre>	원인 분석	단어 목록이 비어 있는 상태에서 퀴즈 기능을 실행하거나 getRandomWord() 수행 시 WordBook.size()가 0임에도 랜덤 인덱스에 접근하여 null 값을 참조하는 상황이 발생함.	해결 방안	<pre>if (book.size() == 0) { JOptionPane.showMessageDialog(parentComponent: null, message: "단어가 없습니다."); dispose(); // 단어 없으면 창 닫음 return; } // 랜덤 단어 반환 (없으면 null) public Word getRandomWord() { int n = book.size(); if (n == 0) return null; int idx = rand.nextInt(n); return book.getAll().get(idx); } -> 퀴즈 실행 시 book.size()가 0인지 사전에 검사하여 단어가 없을 경우 퀴즈 창을 띄우지 않고 사용자에게 안내 메시지를 출력함. 또한 getRandomWord()에서 null 반환 처리를 하고 GUI(Frame) 측에서도 null 체크를 거쳐 안정적으로 제어하도록 개선함.</pre>
원인 분석	텍스트 파일 저장 시 인코딩을 지정하지 않으면 운영체제 기본 인코딩과 불일치하여 한글 손상이 발생할 수 있음.													
해결 방안	<pre>OutputStreamWriter osw = new OutputStreamWriter(new FileOutputStream(FILE_PATH), charsetName: "UTF-8"); -> OutputStreamWriter을 통해 UTF-8을 지정하도록 코드 수정</pre>													
원인 분석	프로그램 최초 실행시 Word.txt 파일이 존재하지 않는 상태에서 FileInputStream이 호출되며 FileNotFoundException이 발생함. 또한 읽기/쓰기 도중 스트림이 정상적으로 닫히지 않거나 파일 접근 중 예상치 못한 출력력 문제(IOError)가 발생할 경우 IOException이 발생.													
해결 방안	<pre>} catch (FileNotFoundException e) { // 파일이 없을 경우 System.out.println(x: "저장된 파일이 없습니다."); } catch (IOException e) { // 읽기 중 오류 System.out.println(x: "불러오기 오류"); } -> try-catch 문을 이용해 FileNotFoundException과 IOException을 처리하여 안내 메시지만 출력하고 프로그램은 정상 실행되도록 구성함.</pre>													
원인 분석	단어 목록이 비어 있는 상태에서 퀴즈 기능을 실행하거나 getRandomWord() 수행 시 WordBook.size()가 0임에도 랜덤 인덱스에 접근하여 null 값을 참조하는 상황이 발생함.													
해결 방안	<pre>if (book.size() == 0) { JOptionPane.showMessageDialog(parentComponent: null, message: "단어가 없습니다."); dispose(); // 단어 없으면 창 닫음 return; } // 랜덤 단어 반환 (없으면 null) public Word getRandomWord() { int n = book.size(); if (n == 0) return null; int idx = rand.nextInt(n); return book.getAll().get(idx); } -> 퀴즈 실행 시 book.size()가 0인지 사전에 검사하여 단어가 없을 경우 퀴즈 창을 띄우지 않고 사용자에게 안내 메시지를 출력함. 또한 getRandomWord()에서 null 반환 처리를 하고 GUI(Frame) 측에서도 null 체크를 거쳐 안정적으로 제어하도록 개선함.</pre>													
<p>4. StatsFrame 기능 구현 (단어 수, 퀴즈 정답률 등)</p> <p>5. GUI 디자인 및 버튼 레이아웃 개선</p> <p>6. 예외 처리 보강</p> <p>7. QuizFrame 정답 기록 및 통계 기능과의 연동</p> <p>8. 단어 중복 처리 로직 구현 및 전체 단어 데이터 정합성 확보</p> <p>9. 전체 통합 테스트 및 버그 수정</p>														
<p>다음 주 계획</p>														

	이름	계획 역할	수행 내용	달성도%
팀원 기여도 평가	이강민	<p><프로젝트 총괄 / GUI 개발></p> <ul style="list-style-type: none"> - 전체 프로그램 구조 설계 및 GUI 통합 - MainFrame 화면 구성 및 이벤트 처리 - QuizFrame 연동 및 UI 흐름 제어 - GUI 사용자 입력 검증 및 예외 상황 처리 	<ul style="list-style-type: none"> - MainFrame의 입력창 · 출력창 · 버튼 패널 GUI 구성 완료 - 단어 추가 · 삭제 · 목록 · 저장 · 불러오기 등 버튼 이벤트 처리 구현 - QuizFrame 호출 기능 연동 및 퀴즈 실행 흐름 통합 - 통계창(StatsFrame) 기본 형태 추가 및 화면 전환 구성 	60%
	장정원	<p><데이터 로직 개발 / 기능 통합></p> <ul style="list-style-type: none"> - WordBook 클래스 구현(단어 등록, 검색, 통계 기능) - Word 클래스 정의 및 데이터 구조 설계 <ul style="list-style-type: none"> - 중복 처리 및 통계 알고리즘 작성 - 기능별 단위 테스트 수행 	<ul style="list-style-type: none"> - Word, WordBook 클래스 코드 구조 확인 및 기본 기능 작성 - Word 클래스 구현(필드 선언, 생성자, getter, toString 정리) - WordBook 클래스 초기 기능 구현 (List 선언, add · getAll 메소드 작성) - 영어 단어 기반 조회(findByEnglish) 및 삭제(removeByEnglish) 기능 추가 - 검색(search) 기능 구현(영어/뜻에 키워드 포함 시 결과 반환) 	65%
	신상현	<p><퀴즈 기능 / 문서화 및 발표 자료></p> <ul style="list-style-type: none"> - QuizFrame 클래스 구현(랜덤 문제 출제 및 정답 판별) - 퀴즈 결과 저장 및 통계 기능 보조 개발 <ul style="list-style-type: none"> - 전체 프로젝트 테스트 케이스 작성 - 프로젝트 보고서 작성 - 발표 자료 제작 및 시연 영상 준비 	<ul style="list-style-type: none"> - QuizFrame에서 랜덤 문제 출제 및 정답 확인 기능 구현 - WordBook 비어 있을 때 NPE 방지를 위한 예외 상황 처리로직 적용 - 퀴즈 정답 결과 안내 메시지 처리 및 UI 구성 - QuizManager를 통한 랜덤 문제 제공 로직 점검 - 활동일지 및 README 문서 구조 점검, 보고서 요소 정리 	60%
	이민제	<p><파일 입출력 및 예외 처리 담당></p> <ul style="list-style-type: none"> - FileManager 파일 저장/불러오기 개발 - 예외 처리 로직 작성 - 인코딩 문제 해결 	<ul style="list-style-type: none"> - FileManager.save/load 기능 완성 (UTF-8 인코딩 적용) - 한글 깨짐 문제 분석 및 OutputStreamWriter를 통한 인코딩 지정 해결 - FileNotFoundException 처리 및 사용자 안내 메시지 적용 - 파일 미존재 시 프로그램 종료 없이 정상 동작 유지하도록 개선 - split 파싱 시 잘못된 데이터 검증 로직 적용(p.length 검사) - 전체 GUI와 파일 입출력 연동 테스트 수행 	70%