객체지향프로그래밍 HW#2

최지헌, 추민솔 TA
oop-ta_grp@ajou.ac.kr

2025-06-11

목차

1 과제

1.1 목표

- Java Swing과 MVC 패턴을 활용하여 GUI 애플리케이션 구조화 방법을 학습한다.
- 사용자 입력 처리, 동적 UI 구성, 상태 기반 필터링 등 이벤트 중심 프로그래밍의 작동 원리를 실습을 통해 익힌다.
- View와 Controller의 분리, Model 관리 구조를 통해 유지보수성과 확장성이 높은 코드 구조를 경험한다.

1.2 프로그램 개요

- 사용자는 텍스트 입력창에 내용을 입력한 뒤 추가 버튼을 눌러 TodoList에 항목을 등록할 수 있다.
- 각 작업 항목은 완료 여부(체크박스)와 우선순위(High / Low)를 가지며, 우선순위는 버튼을 눌러 전환(Toggle)할 수 있다.
- 각 항목 옆에는 삭제 버튼이 있어 클릭 시 해당 항목이 제거된다.
- 하단의 콤보박스를 통해 상태(Done / Not Done / None) 및 우선순위(High / Low / None)를 동시에 필터링할 수 있다.
- 전체 항목 수 및 완료 항목 수는 하단 우측 상태 라벨에 실시간 표시된다.
- 동일한 텍스트를 가진 항목은 중복해서 추가할 수 없다.

1.3 프로그램 구성

- 본 과제는 MVC(Model-View-Controller) 구조를 기반으로 작성되었으며, 각 구성 요소는 다음과 같은 역할을 담당한다:
  - `Task.java` 작업(Task)의 상태 및 속성을 관리하는 데이터 모델
  - `TaskItemView.java` 단일 작업 항목을 UI로 표현하는 View 구성 요소
  - `TodoPanel.java` 전체 화면 구성 및 사용자 인터랙션을 관리하는 메인 GUI 패널
  - `TodoController.java` 작업 추가, 삭제, 필터링 등 주요 비즈니스 로직을 처리

1.4 구현 요구사항

1.4.1 Model: Task 클래스

필드 변수

- `private String text` 작업 내용을 저장하는 문자열
- `private boolean done` 완료 여부를 나타내는 플래그
- `private String priority` 작업의 우선순위를 나타내는 문자열 (LOW 또는 HIGH)

주요 메서드

- `public Task(String text)`
  텍스트를 받아 새로운 작업을 생성한다. 기본 완료 상태는 `false`, 우선순위는 `"LOW"`로 설정된다.
- `public String getText()`
  작업 텍스트를 조회한다.
- `public boolean isDone()`, `public void setDone(boolean done)`
  작업의 완료 여부를 확인하거나 설정한다.
- `public String getPriority()`, `public void togglePriority()`
  현재 우선순위를 반환하거나, `"LOW"`와 `"HIGH"` 사이로 전환한다.

1.4.2 View: TaskItemView 클래스

필드 변수

- `private JPanel panel` — 전체 UI 요소를 포함하는 패널
- `private Task task` — 현재 UI 요소와 연결된 Task 데이터 객체

주요 메서드

- `public TaskItemView(Task task, Runnable onUpdate, Runnable onDelete)`
  주어진 `Task` 객체를 기반으로 UI를 구성하며, 완료 체크박스, 우선순위 전환 버튼, 삭제 버튼 등의 이벤트를 설정한다. 각 이벤트 발생 시 `onUpdate` 또는 `onDelete` 콜백을 실행한다.
- `public JPanel getPanel()`
  구성된 UI 패널을 반환한다.

1.4.3 View: TodoPanel 클래스

필드 변수

- `private JTextField inputField` — 사용자 입력을 받는 텍스트 필드
- `private JButton addButton` — 작업을 추가하는 버튼
- `private JPanel listPanel` — 작업 항목들을 나열하는 목록 패널
- `private JScrollPane scrollPane` — 스크롤 가능한 목록 영역
- `private JLabel statusLabel` — 전체 작업 수와 완료 수를 표시하는 라벨
- `private JComboBox<String> statusFilter` — 완료 상태 필터를 선택하는 콤보박스
- `private JComboBox<String> priorityFilter` — 우선순위 필터를 선택하는 콤보박스
- `private TodoController controller` — 작업 데이터를 관리하는 컨트롤러

주요 메서드

- `public TodoPanel()`
  전체 컴포넌트를 초기화하고, 입력 패널, 작업 목록, 필터, 상태 표시를 포함한 화면을 구성한다. 각 이벤트에 대한 리스너를 등록하고 초기 리스트를 렌더링한다.
- `private void refreshList()`
  현재 선택된 필터 조건에 따라 작업 목록을 갱신한다. 각 작업 항목을 `TaskItemView`로 감싸어 `listPanel`에 추가하며, 하단 상태 라벨의 텍스트를 최신 상태로 갱신한다.
- `public static void main(String[] args)`
  `TodoPanel`을 포함한 창을 생성하여 애플리케이션을 실행한다.

1.4.4 Controller: TodoController 클래스

필드 변수

- `private List<Task> taskList` 전체 작업(Task) 객체들을 저장하는 리스트

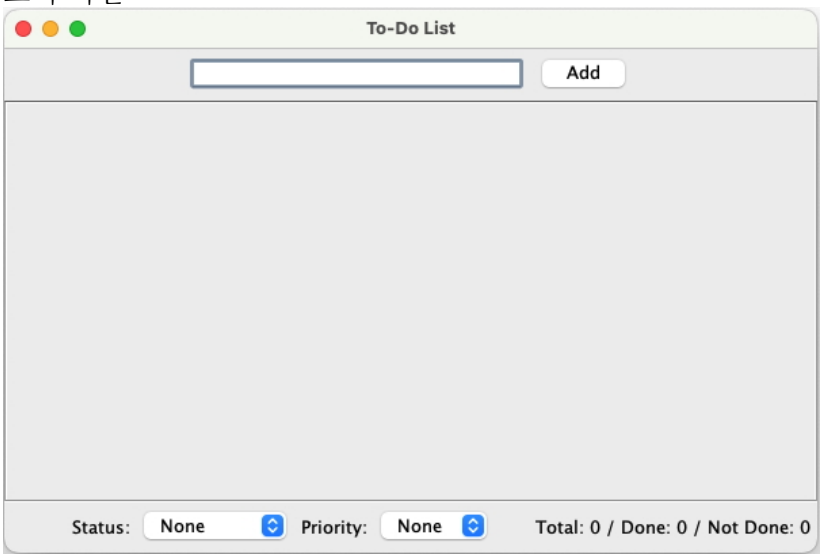
주요 메서드

- `public boolean addTask(String text)`
  전달받은 텍스트를 기반으로 새로운 작업을 생성하여 리스트에 추가한다. 동일한 텍스트의 작업이 이미 존재하는 경우에는 추가하지 않고 `false`를 반환한다.
- `public void removeTask(Task task)`
  리스트에서 해당 작업 객체를 제거한다.
- `public List<Task> getFilteredTasks(String status, String priority)`
  주어진 상태('None', 'Done', 'Not Done') 및 우선순위('None', 'High', 'Low') 조건을 기준으로 작업을 필터링한 후, 그 결과 리스트를 반환한다.
- `public int getTotalCount()`
  전체 작업의 개수를 반환한다.
- `public int getDoneCount()`
  완료된 작업의 수를 반환한다.

1.5 예시 실행 시나리오

- 예시 실행 결과는 과제 프로그램의 실행 예시로, 버튼의 위치 등 상세 UI는 예시와 상이해도 가능
- 단, 모든 기능(완료 상태 변경, 추가, 삭제, 필터링)은 예시와 동일하게 동작해야 함

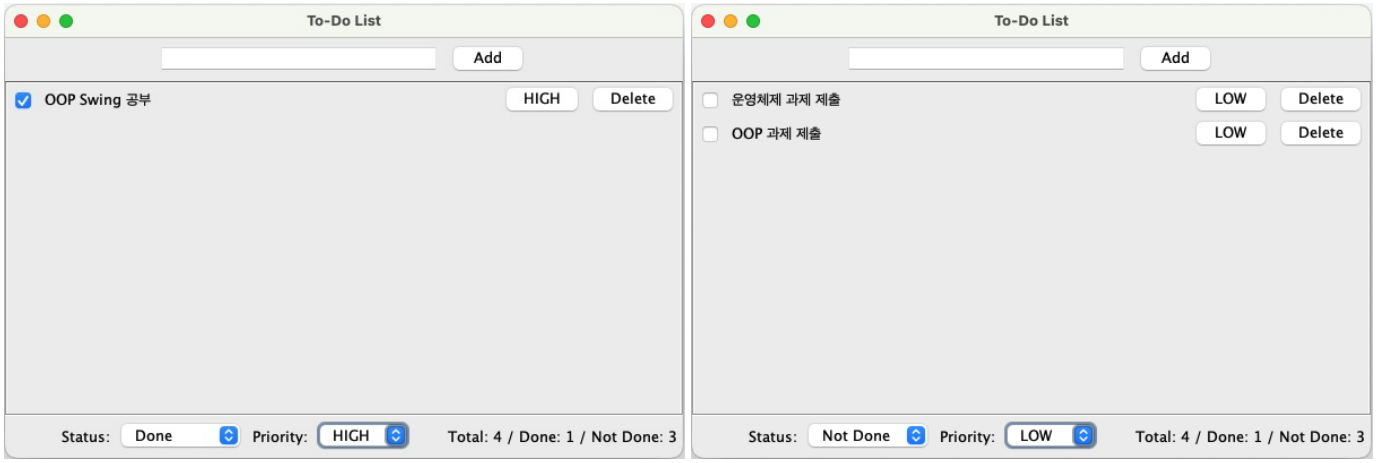
- 초기 화면



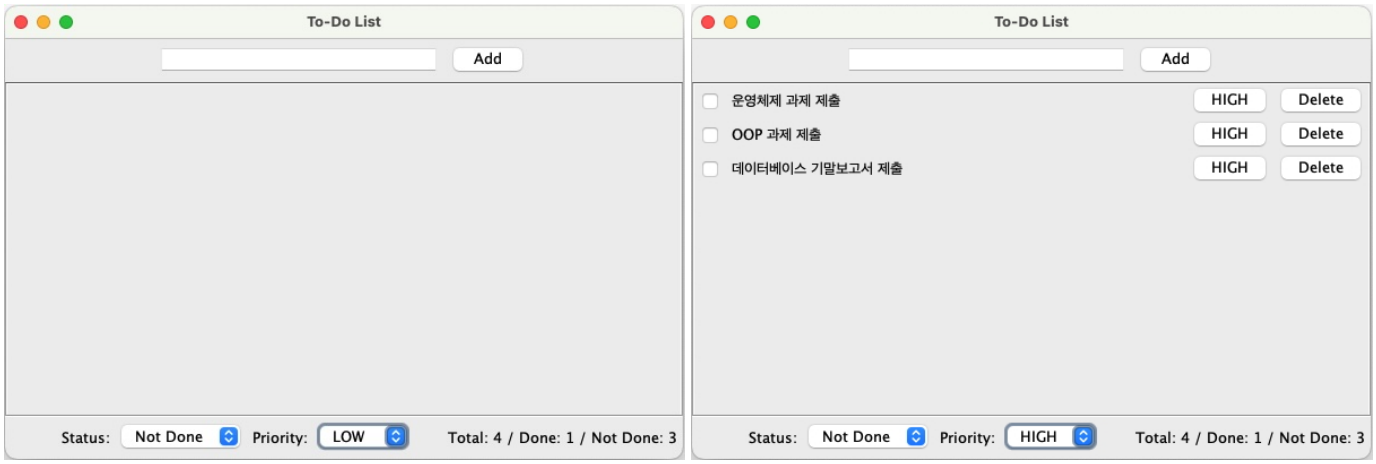
- 텍스트 입력 창을 통해 아래 항목을 순서대로 추가 한 경우
  - OOP Swing 공부
  - 운영체제 과제 제출
  - OOP 과제 제출
  - 데이터베이스 기말보고서 제출

- 항목 별로 완료 및 우선 순위를 변경한 경우

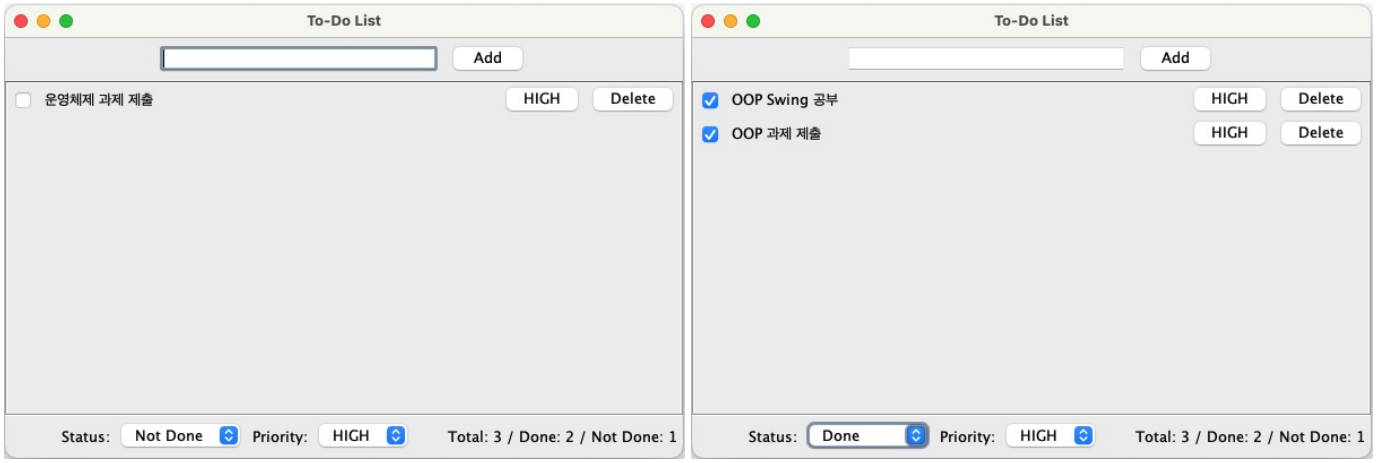
- 필터링을 변경한 경우



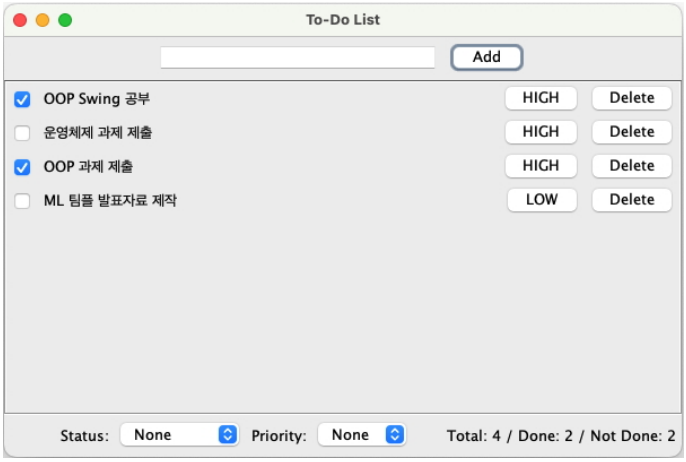
- "운영체제 과제 제출", "OOP 과제 제출"의 우선순위를 HIGH로 변경한 경우



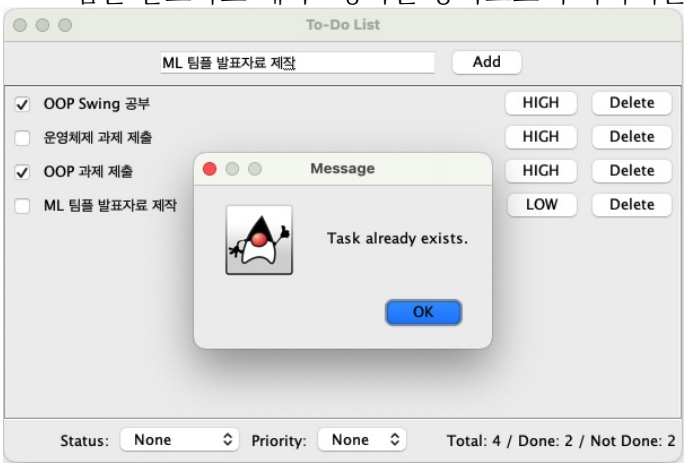
- "OOP 과제 제출", "데이터베이스 기말보고서 제출"을 완료 처리한 경우



- 필터링 변경 후, "ML 팀플 발표자료 제작"을 추가한 경우



- "ML 팀플 발표자료 제작" 항목을 중복으로 추가하려는 경우



1.6 유의 사항

- 본 과제는 학생들의 GUI 프로그래밍 능력 함양을 위한 목적으로, ChatGPT, Claude 등 생성형 AI의 사용을 **지양**합니다.
- 하지만, 부득이하게 생성형 AI의 도움을 받은 경우, 반드시 보고서에 다음 내용을 **구체적으로 기술**하여야 합니다.
  - AI의 도움을 받은 기능 또는 코드의 구체적인 내용
  - AI가 생성한 코드 또는 설명을 기반으로 본인이 추가적으로 수행한 분석, 수정 또는 이해 내용
  - 해당 기능을 스스로 구현하기 위해 시도한 과정이 있다면 그 과정도 함께 기술
- 생성형 AI를 사용하고도 보고하지 않은 것이 확인될 경우, 감점 또는 기타 불이익이 있을 수 있습니다.

1.7 제출 요구사항

- Java 소스 파일(.java) 및 보고서를 이름_학번.zip 형식으로 압축하여 제출

  예시
  홍길동_202401234.zip/
  ├── code/
  │ ├── Task.java
  │ ├── TaskItemView.java
  │ ├── TodoController.java
  │ └── TodoPanel.java
  └── 홍길동_202500000_보고서.pdf

- 보고서에는 구현한 클래스 및 메서드 설명, 설계 방식, 실행 시나리오를 모두 포함시킬 것
- 과제 제출은 6/22 (일요일) 23:59까지 이며, 지각 제출 D+1 시, -40%, D+2 시, -70%, D+3 시, -100% 감점 처리