2025-05-24 ~ 2025-06-10 정리

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.time.LocalDateTime;
```

- `javax.swing.*` : GUI 구성 요소 (버튼, 라벨 등)
- `java.awt.*` : 레이아웃 및 컴포넌트 스타일링
- `java.awt.event.*` : 마우스/키보드 이벤트 처리
- `java.io.*` : 파일 입출력 (기록 저장용)
- `java.time.LocalDateTime` : 현재 시간 기록

```
public class StudyGardenApp extends JFrame {
```

- `JFrame` 을 상속받은 메인 GUI 클래스입니다.
- 이 안에 모든 버튼, 텍스트 필드, 타이머 등의 GUI 구성 요소가 정의됩니다.

```
private JTextField nicknameField, taskTitleField;
private JButton startButton, stopButton, loadButton, deleteButton;
private JLabel plantStatusLabel, timerLabel;

private Timer timer, idleTimer;
private int elapsedSeconds = 0;
private boolean isIdle = false;
private final int IDLE_TIMEOUT = 30;
```

- **입력 필드**: 닉네임, 과제명
- **버튼들**: 시작, 종료, 기록 불러오기, 기록 삭제
- **라벨들**: 식물 상태, 타이머 시간
- **타이머들**:
  - `timer` : 실제 경과 시간 측정용
  - `idleTimer` : 유휴(아무 동작 없음) 상태 감지용
- **그 외**: 경과시간 카운트, 유휴 상태 플래그

```
public StudyGardenApp() {
    setTitle("StudyGarden 🌱");
    setSize(400, 350);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    initUI();
    initIdleDetection();
    setVisible(false); // 로그인 성공 전까지 숨김
}
```

- 창 제목/크기 설정
- 레이아웃 설정 (BorderLayout 사용)
- `initUI()` 로 UI 생성
- `initIdleDetection()` 으로 유휴 상태 감지 기능 등록
- 최초에는 창을 보이지 않음 → 로그인 성공 후 표시

```
private void initUI() {
    JPanel backgroundPanel = new JPanel(new BorderLayout());
    backgroundPanel.setBackground(new Color(210, 250, 210));
    add(backgroundPanel);

    JPanel topPanel = new JPanel(new GridLayout(4, 2));
    topPanel.setBackground(new Color(200, 240, 200));

    topPanel.add(new JLabel("닉네임:"));
    nicknameField = new JTextField();
    topPanel.add(nicknameField);

    topPanel.add(new JLabel("과제 제목:"));
    taskTitleField = new JTextField();
    topPanel.add(taskTitleField);

    startButton = new JButton("시작하기");
    startButton.setBackground(new Color(150, 200, 150));
    startButton.addActionListener(e -> startTimer());
    topPanel.add(startButton);

    stopButton = new JButton("종료하기");
    stopButton.setBackground(new Color(200, 100, 100));
    stopButton.setForeground(Color.WHITE);
    stopButton.addActionListener(e -> stopTimer());
    topPanel.add(stopButton);
```

```
        loadButton = new JButton("기록 불러오기");
        loadButton.setBackground(new Color(100, 150, 200));
        loadButton.setForeground(Color.WHITE);
        loadButton.addActionListener(e -> loadStudyRecords());
        topPanel.add(loadButton);

        deleteButton = new JButton("기록 초기화");
        deleteButton.setBackground(Color.RED);
        deleteButton.setForeground(Color.WHITE);
        deleteButton.addActionListener(e -> deleteStudyRecords());
        topPanel.add(deleteButton);

        backgroundPanel.add(topPanel, BorderLayout.NORTH);

        JPanel centerPanel = new JPanel(new GridLayout(2, 1));
        centerPanel.setBackground(new Color(180, 230, 250));

        timerLabel = new JLabel("⏱️ 경과 시간: 0초", SwingConstants.CENTER);
        timerLabel.setFont(new Font("SansSerif", Font.BOLD, 16));
        centerPanel.add(timerLabel);

        plantStatusLabel = new JLabel("🌱 현재 상태: 씨앗", SwingConstants.CENTER);
        plantStatusLabel.setFont(new Font("SansSerif", Font.BOLD, 18));
        plantStatusLabel.setForeground(new Color(0, 128, 0));
        centerPanel.add(plantStatusLabel);

        backgroundPanel.add(centerPanel, BorderLayout.CENTER);
    }
```

- 전체 `backgroundPanel` 설정 (연한 녹색 배경)
- `topPanel` : 닉네임, 과제명 입력 필드, 4개의 버튼 추가
- `centerPanel` : 시간 표시와 식물 상태 표시
- 각 버튼에 색상, 이벤트 핸들러 연결 (`addActionListener`)
- `BorderLayout.NORTH` / `CENTER` 를 사용해 화면 배치

```
    private void stopTimer() {
        if (timer != null && timer.isRunning()) {
            timer.stop();
            saveStudyRecord(); // ⬅️ 저장 기능
            JOptionPane.showMessageDialog(this, "■타이머가 종료되었습니다!");
        }
    }
```

- `elapsedSeconds` 초기화

- 기존 타이머 중지
- 새 `Timer` 생성: 1초마다 `elapsedSeconds` 증가
- 시간 경과에 따라 식물 상태 변화:
  - 10초: 씨앗 → 새싹
  - 20초: 새싹 → 나무
- 타이머 시작 후 알림 메시지 출력

```
private void stopTimer() {
    if (timer != null && timer.isRunning()) {
        timer.stop();
        saveStudyRecord(); // ⬅️ 저장 기능
        JOptionPane.showMessageDialog(this, "■ 타이머가 종료되었습니다!");
    }
}
```