

# 자바 프로젝트 계획 문서

32201525 문주안

**주제:** Swing을 이용한 간단한 스톱워치/타이머

## 1. 프로젝트 목표

- A. **핵심 목표:** Java Swing을 이용하여 기본적인 시작, 중지, 리셋 기능이 있는 스톱워치와 시간만큼 카운트 다운하는 타이머 기능을 가진 애플리케이션 제작을 목표로 한다.
- B. **학습 목표:** 클래스 다이어그램 작성을 통한 계획법, Java Swing GUI 학습, 이벤트 리스너 처리, 스레드 또는 타이머 클래스 활용, 시간 데이터 처리 방법의 학습을 목표로 한다.

## 2. 주요 기능

이 애플리케이션의 목표는 최소한의 기능을 구현하는 것이고 선택 기능의 추가를 고려한다.

### A. 스톱워치

- i. 시간 표시 (h:m:s.ms)
- ii. 시작 버튼 (시작 버튼을 눌러 스톱워치를 동작하고, 중지된 스톱워치를 다시 동작하는 버튼)
- iii. 중지 버튼 (스톱워치를 일시 중지하는 버튼)
- iv. 리셋 버튼 (스톱워치를 0:0:0으로 리셋하는 버튼)
- v. (선택) 스톱워치 동작 중 기록 버튼

### B. 타이머

- i. 시간 표시 (h:m:s.ms)
- ii. 시간 설정 (카운트 다운 시간 설정)
- iii. 중지 버튼 (타이머를 일시 중지하는 버튼)
- iv. 시작 버튼 (시작 버튼을 눌러 타이머를 동작하고, 중지된 타이머를 다시 동작하는 버튼)
- v. 리셋 버튼 (타이머를 설정한 시간으로 리셋하는 버튼)
- vi. 알림 (타이머가 0초가 되었을 때 알리는 기능)

### 3. 사용 기술 스택

- A. GUI 라이브러리: Swing
- B. 개발 도구(IDE): Visual Studio Code
- C. 버전 관리: Git, Github

### 4. 개발 단계

- A. 기본 UI 창 만들기: 메인 화면
- B. 스톱워치 UI 추가: 스톱워치 시간 표시, 버튼
- C. 스톱워치 기본 로직 구현: 시작, 중지, 리셋
- D. UI와 로직 연결 (스톱워치): 버튼 클릭 시 로직 작동
- E. 타이머 UI 추가: 타이머 시간 표시, 시간 설정 UI, 버튼
- F. 타이머 기본 로직 구현: 시간 설정, 시작, 중지, 리셋, 알람
- G. UI와 로직 연결 (스톱워치): 버튼 클릭 시 로직 작동, 시간 설정
- H. 테스트 및 디버깅: 기능이 제대로 동작하는지와 예외 처리 확인
- I. (선택) 기능 개선: 스톱워치 동작 중 기록, 디자인 개선, 코드 리팩토링

### 5. 클래스 다이어그램 (변경 가능성 있음)

