

2025 소프트웨어프로젝트 **프로젝트 6**

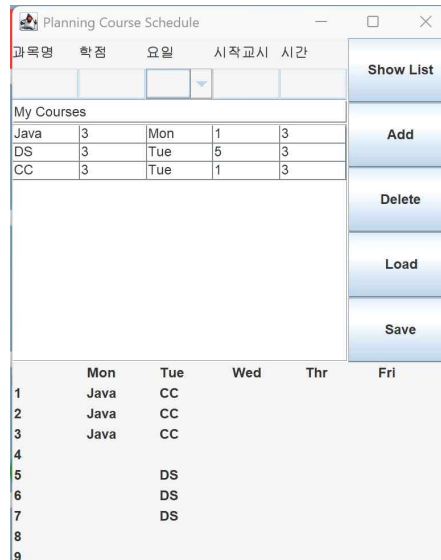
Course Schedule 프로그램 최종 설계 및 구현

1. 내용

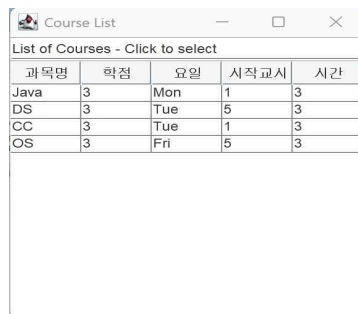
프로젝트 5의 GUI에 이벤트 핸들링을 추가해서 Course Schedule 프로그램을 완성한다.

주의. 프로젝트5는 GUI의 연습 목적. 코드를 구조화해서 반드시 재작성하고, 그 위에 동작을 추가.

- 프로그램을 수행하면, 정해진 파일 (예, myschedule-normal.data)에 있는 수강 과목 정보들을 읽어 와서 내부 자료구조에 저장하고, 화면에 표시한다. 파일에서 읽어 오는 프로그램 코드는 프로젝트 3의 것을 사용.



- 메인 화면의 오른쪽에 있는 “Show List”버튼을 누르면, 수강 신청이 가능한 과목들의 리스트를 포함한 별도의 화면이 아래와 같이 추가로 표시된다. 표시되는 과목들은, 현재 수강 신청하고 있는 과목들과는 다른, 별도의 파일에 저장되어 있다. (예, course-DB.data)



- 별도의 과목 리스트에서 수강하려는 과목을 클릭하여 선택하면,
- 선택한 과목의 내용이 수강 신청 화면 상단에 표시되며 (**중요 사항**), 추후 “Add” 버튼 클릭을 통해서 수강 신청이 이루어진다.
- 과목 리스트는 별도의 종료 클릭을 하지 않으면, 계속 화면에 표시되어 과목 선택의 변경 및 추가가 가능하다.
- 수강 신청 화면의 상단에는 추가될 과목의 정보가 표시된다. 과목 정보는, 사용자가 임의로 입력/수정할 수 없으며, 반드시 미리 확립 정의된 과목 리스트에서 전달되어야 한다.
- “Add” 버튼을 누르면, 상단의 과목에 대한 수강 신청이 이루어지면서, 중하단의 테이블과 시간표 내용이 갱신된다.
- “Delete”버튼을 누르면, 선택된 과목을 화면과 메모리에 있는 course schedule list에서 제거한다. (가운데 과목 리스트에서 특정 과목이 선택되지 않은 경우, 먼저 선택해야 한다는 dialogue를 띄운다.)

- “Load” 버튼을 누르면, 현재 상태, 즉, 메모리에 있는 수강 과목 정보에 상관없이, 파일에 있는 수강 과목 정보들로 메모리의 course schedule을 새로 구축한다. 당연히 화면도 파일에 있는 수강 과목 정보들로 새로 표시되어야 한다. data 파일의 지정은 GUI를 이용해서 선택되어야 한다.
- “Save” 버튼을 누르면, 현재 화면에 표시된 수강 과목들 정보를 정해진 파일에 저장한다. (별도로 “Save”를 하지 않으면, 수강 과목 정보의 갱신은 프로그램이 종료되면 무효화된다.) 파일의 지정은 GUI를 이용해서 선택되어야 한다.

2. 목적

- 객체지향 프로그램 설계 연습
 - * 필요 Class 정의, 클래스별로 기능 나누기
 - * 프로그램 크기의 증대에 따른 관리 문제 및 Java의 해결법 체득
- 한 학기 배운 모든 주요 Java 기능 체득
 - * 기본 GUI, File I/O, 스트링 처리 (간단한 DB 프로그래밍)
 - * Event-driven programming, object reference 처리
 - * incremental programming
- 개발도구 활용법, 디버깅
- 프로그램 개발 과정 체득

3. 추진 방법

- 1) 서로 독립적인 Class들 사이에서 자료 공유하는 방법 연구
 - * 이번 과제에서는 JFrame을 기반으로 하는 클래스들이 복수 개 생성됨.
 - * 그들 간의 정보 교환 필요
 - (예, 과목 리스트 창에서 특정 과목을 선택한 경우, 해당 과목 정보가 메인 화면으로 전달되어 상단의 과목 정보가 갱신되어야 함)
 - * 프로그램 구조의 틀을 유지하면서 정보 교환
- 2) 또한, 프로그램 내부에 있는 CourseSchedule 객체 정보와 화면에 보여지는 내용을 항상 일치시켜야 함.
- 3) 클래스 구조를 유지하면서 위 두 문제를 동시에 해결하는 방법을 찾아야 함.
- 4) Delete 기능 구현을 위해서는, 메인 화면에 나오는 여러 과목 중 어떤 과목이 선택되었는지 파악하는 기법도 필요. (JTable 사용을 추천하는 주 이유)
- 5) 설계 사항
 - 사용자에게 제공되는 GUI 설계 (프로젝트 5)
 - 어떤 class 들로 구성을 할 것인지 결정
 - 두 화면 모두에 필요한 과목들의 리스트를
 - * 어떤 GUI 컴포넌트를 사용할 것인지,
 - * 같은 class로 또는 다른 class로 구현할 것인지 결정
 - 다시 한번, **구조 > 기능 > 성능**
 - 어떤 기능을 어떻게 구현할 것인가는 설계자의 권한. 단, 결정 후 리포트에 명시.
 - * 요구되는 최소 기능만 구현하면 되며, 구현 방법은 설계자의 선택

6) 프로그램 완성 제출

4. 평가항목

- 프로그램의 동작 여부 (70%)
 - * 동작 여부를 확연히 알아볼 수 있는 결과
 - * 본인이 디버깅해가면서 프로그래밍을 했다면, 설명을 대충해서 오해를 만들겠는가?
 - * 본인이 작성한 프로그램의 진위 여부를 판단하는데 도움 되는 인증 자료 포함
- 각 단계에 대한 설계 노트 (30%)
 - * 어떤 이유로 어떤 결정을 내렸는지 명확하게 설명
 - * 설계 및 구현에 대한 자체 평가: 한계점 및 개선 방안 제시
- 자체평가표 참조
- 평가에 대해 불안해 할 필요 없음. 불안한 사항이 있으면, 리포트에 찾기 쉽게 기술
- 수강 과목 중 적어도 1 과목은 본인 관련 내용 포함 (본인의 과제 수행 여부를 증명)

5. 리포트 제출

6/8(일)까지. (마감 시간 이후에 제출 불가)

- 최종 테스트 과정 화면, 설계 노트, 소스 프로그램, 자체평가표
- 하나로 묶어서 eClass로 제출