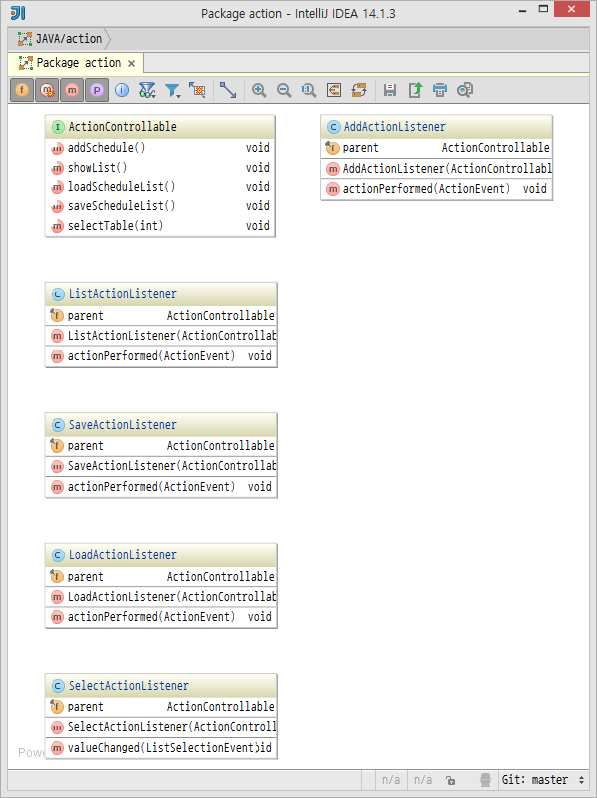
1. **설계 목표**

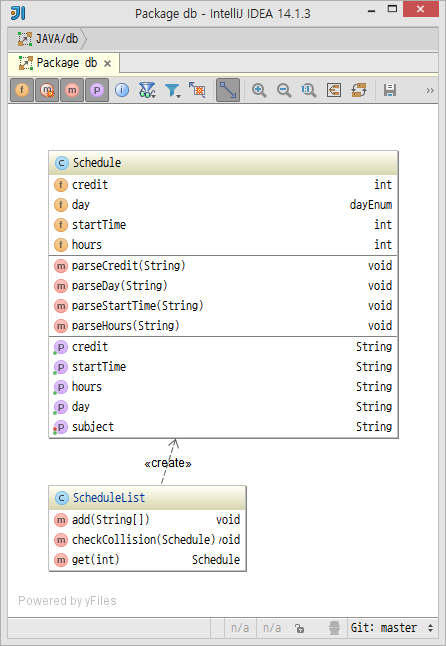
* 객체지향 프로그램 설계하기: 필요한 Class를 정의하고, 각각의 클래스를 최대한 작은 기능만을 담당하게 해서, 객체지향 프로그래밍의 원칙을 최대한 준수한다.
* 코드 재활용: 기존에 만들었던 CourseSchedule 코드를 최대한 그대로 써서 붙일 수 있게 한다.
* 프로그램 개발 과정 체득: 설계하고, 구현하고, 테스트하고, 디버그하는 프로그램 개발의 기본과정을 직접 터득한다.

1. **프로젝트 구성**
2. package main: 프로그램 전체와 테스트 모듈을 넣는 패키지
   * class CourseSchedule: 프로그램 전체를 관리, 정해진 Action이 발생하면, 그에 맞는 메소드를 실행한다.
3. package action: 프로그램에 필요한 Action을 넣어놓은 패키지
   * class AddActionListener: 새로운 Schedule을 추가하려고 하면, 지정된 table에 새로 추가한다.
   * class ListActionListener: 현재 파일에 저장된 Schedule 목록을 보여주는 창을 보여준다.
   * class LoadActionListener: 현재 파일에 저장된 Schedule 목록을 읽어서 지정된 table에 출력한다. (기존에 있던 내용은 지워짐)
   * class SaveActionListener: 지정된 table이 현재 가지고 있는 Schedule 목록을 파일로 만든다.
   * interface ActionControllable: action 패키지 안에 있는 ActionListener을 사용하려면 반드시 구현해야 되는 method를 정의해 둔 Interface
4. package db: 저장된 강의를 관리하는 패키지
   * class Schedule: 강의 하나에 대한 클래스를 정의함.
   * class ScheduleList: 강의 목록을 저장하는 클래스, table의 데이터를 저장하기 위해 사용한다.
5. package fileio: 강의 목록 파일의 입출력을 관리하는 패키지
   * class ScheduleFileReader: 강의 목록 파일을 한 줄씩 읽어서 ArrayList<String>로 반환한다.
   * class ScheduleFileParser: 강의 목록 파일을 String으로 Reader가 불러오면 그걸 Schedule 형태로 바꿔준다.
   * class ScheduleFileWriter: 강의 목록을 파일로 저장한다.
6. package gui: GUI와 관련된 클래스를 담고 있는 패키지
   1. package gui.component: GUI에서 사용할 Custom Component
      * class JAddButton: add 버튼, AddActionListener가 기본적으로 포함됨
      * class JListButton: list 버튼, ListActionListener가 기본적으로 포함됨
      * class JLoadButton: load 버튼, LoadActionListener가 기본적으로 포함됨
      * class JSaveButton: save 버튼, SaveActionListener가 기본적으로 포함됨
      * class JScheduleComboBox
      * class JScheduleTable: 강의 목록을 출력해주는 표
      * class ScheduleTableModel: JScheduleTable을 구현하기 위한 TableModel Class
      * class JScheduleTextField
   2. package gui.frame: GUI에서 띄워주는 창
      * class CourseList: 강의 목록을 출력하는 창
      * class MakeCourseSchedule: 강의를 편집하는 창
      * class JOperatorFrame: 프로그램에 명령을 내릴 수 있는 창을 정의한 abstract class
   3. package gui.panel: GUI의 창 안에 넣어둘 panel
      * class JButtonPanel: 버튼들을 가지고 있는 panel
      * class JInputPanel: 새 Schedule의 내용을 입력하는 panel
      * class JScheduleFormPanel: 입력 panel과 현재 편집 중인 강의 목록을 보여주는 panel
      * class JTablePanel: 사용자에게 보여줄 메시지와 강의 목록을 보여주는 panel
7. **프로젝트 구현**
   1. Action 구현



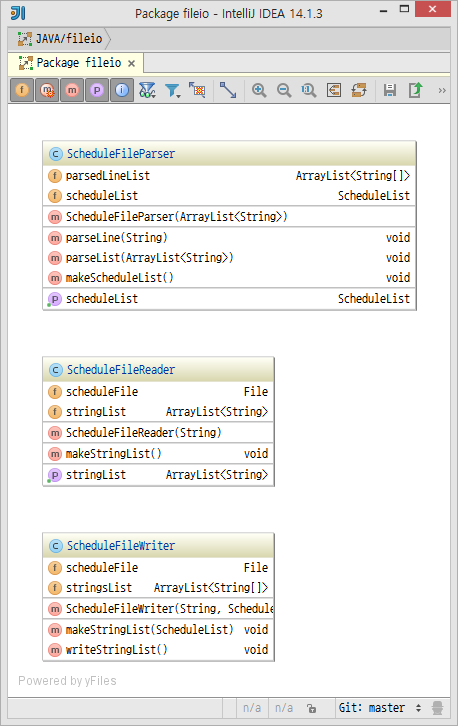
버튼을 누르거나 표의 항목을 입력했을 때, 반응하는 Action을 구현하기 위해 ActionListener를 implement한 후에 프로그램을 관리하는 클래스의 method를 호출할 수 있도록 구현했습니다. ActionControllable이라는 interface를 만들어서 프로그램을 관리하는 클래스가 Action에 반드시 반응하도록 하기 위해 속성을 정해주고, 프로그램을 관리하는 클래스를 직접 넘기는 대신 interface를 넘겨서 Action 클래스들이 프로그램에 Action에 관련된 메소드 이외에는 볼 수 없도록 제한했습니다.

* 1. DB



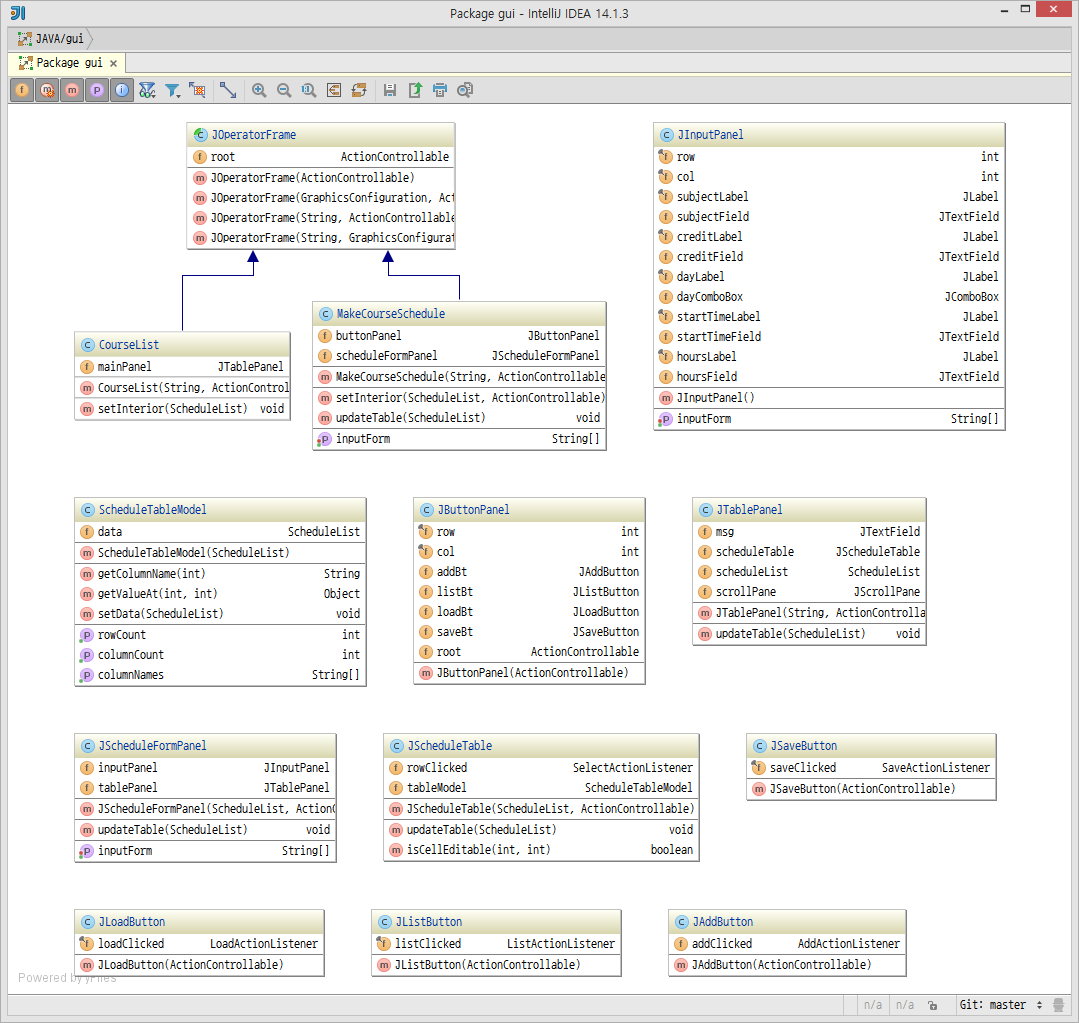
GUI에서 보여주고, 파일에 저장하기 위해 사용하는 강의 자료형과 강의 목록 자료형을 정의한 클래스들입니다. Schedule 클래스로 강의에 필요한 정보와 각 정보의 자료형을 제한하고, 이 강의들을 ArrayList를 상속받은 ScheduleList에 저장합니다. 기존의 add기능에 시간표 충돌을 확인하는 기능을 확장해야 했기 때문에 기존 ArrayList를 확장시킨 ScheduleList를 사용했습니다.

* 1. File I/O



강의 목록을 파일에 불러오고 저장하기 위한 클래스들입니다. ScheduleFileReader는 파일을 한 줄마다 String 단위로 끊어서 ArrayList를 만들어주는 클래스이고, ScheduleParser를 통해 강의 목록으로 바꿔줍니다. 파일에 저장하는 경우는 ScheduleFileWriter를 통해 편집한 강의 목록을 강의당 한 줄씩 String으로 만들어서 파일로 쓴다.

* 1. GUI



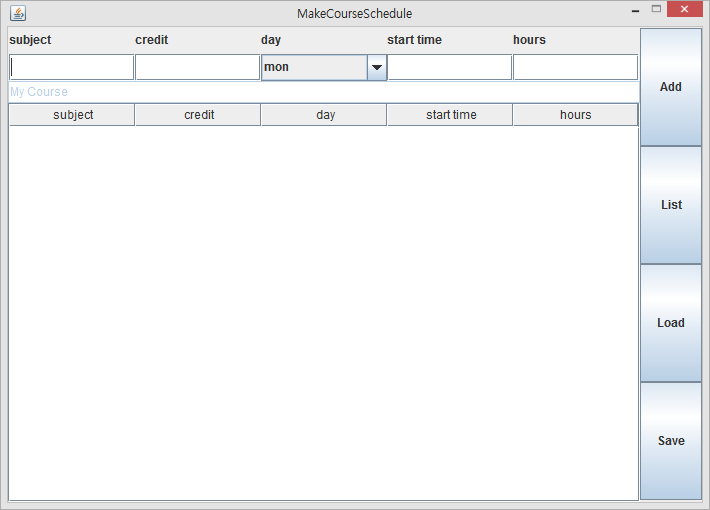
GUI를 구현할 때는 새로 만들 컴포넌트와 패널, 프레임을 패키지에서 구분해서 넣고, 컴포넌트 -> 패널 -> 프레임 순으로 구현을 했습니다.

새로 만들어서 사용할 컴포넌트로 미리 만든 action들을 가지고 생성되는 버튼들과 항목을 선택하면 값을 입력창에 넣어주는 action을 가지고 있는 JScheduleTable이 있습니다. table에서 ScheduleList로 구현된 강의 목록들을 바로 사용할 수 있도록 같은 파일에 ScheduleTableModel을 구현해서 JScheduleTable의 형태와 각 cell마사 사용할 데이터를 정의해준 후, SelectActionListener를 연결해서 사용자의 입력에 반응하도록 구현했습니다.

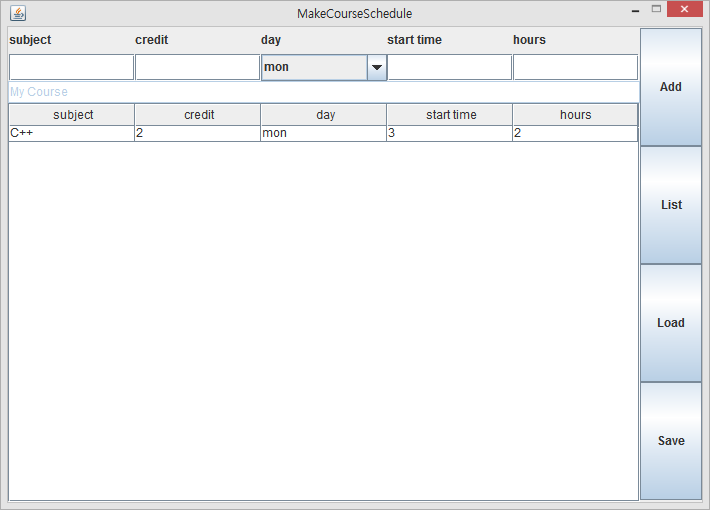
다음으로 이 컴포넌트와 기존의 javax.swing에 있는 컴포넌트로 패널을 구현했습니다. 이 패널들은 단순히 컴포넌트를 배치하는 용도가 아닌 컴포넌트와 프로그램이 action이 발생하면 서로 데이터를 주고 받을 수 있는 통로로도 쓰이기 때문에, 확장된 패널이므로 JPanel을 상속받아서 패널을 새로 정의했습니다.

마지막으로 이 패널을 담고 있는 프레임을 구현했습니다. 이 프레임들은 패널들을 담고 있고, 또한 새로 정의한 action을 사용하기 위해서 반드시 필요한 member field인 ActionControllable root(action으로 행동을 지시할 프로그램 관리 객체를 가리킴)을 넣어둔 JOperationFrame을 abstract class로 만들고 새로 만든 프레임들이 컴포넌트를 통해 프로그램 관리 객체에 명령을 내릴 수 있도록, 상속받게 구현했습니다.

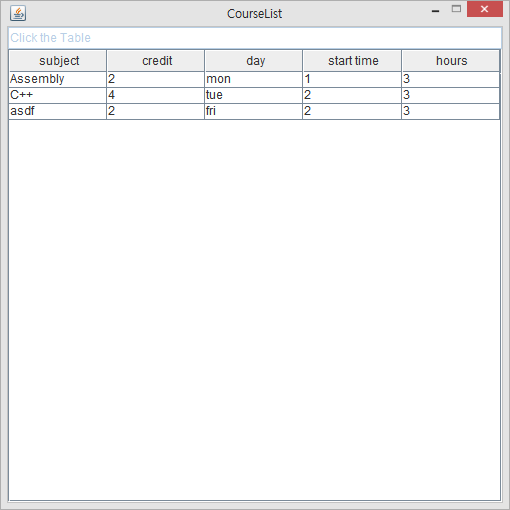
1. **테스트 화면**



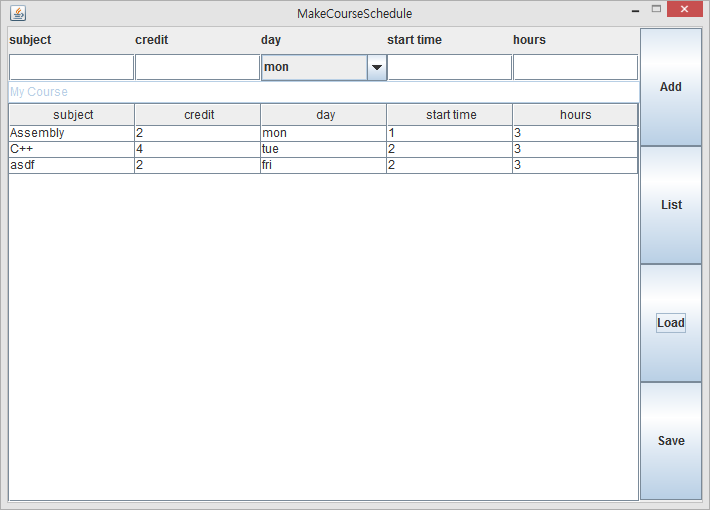
프로그램을 맨 처음 실행하면 나오는 화면입니다.



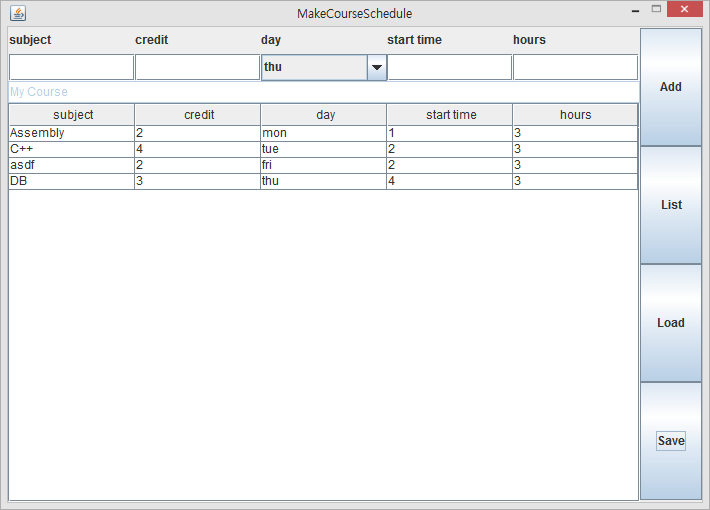
add 버튼을 눌러서 과목을 입력하면 값이 들어간다.



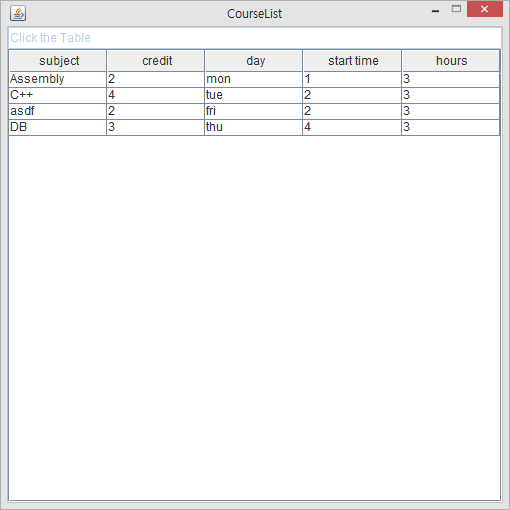
list 버튼을 누르면 파일에 저장된 강의 목록이 출력된다.



load 버튼을 누르면 CourseList 창에서 본 목록이 강의 편집 목록으로 그대로 불러와지는 것을 볼 수 있다.



새로운 과목을 넣고 save 버튼을 누르면,



CourseList에서 볼 수 있듯이 파일에 새 과목도 같이 저장되어서 불러와지는 것을 볼 수 있다.

1. **총평**

* 예외 처리와 File IO을 구현하면서, 익명클래스를 많이 사용했는데, 자세히 언급하면, 예외 처리에서는 예외 메시지를 보내기 위해 new Exception(msg)와 같은 형태로 보내는 방식으로 썼고, File IO는 save, load, list method에서만 사용한다고, 익명클래스로 한 번만 불러서 나온 결과값으로만 필요한 작업을 실행하도록 구현했다. 작동 여부에는 문제가 없지만, 이 프로그램이 더 커지면, 묶어놓지 않고 임시로 이렇게 불러서 쓴 익명클래스가 문제가 될 것이라는 생각을 했다.
* 패키지를 분리해서 작업하다 보니, 다른 패키지의 클래스를 불러서 새로운 클래스로 작업하려면 불러온 클래스가 public이여야 한다. 소스코드를 다시 읽어보니, 생각보다 public이 너무 많아서 필요하지도 않은 공개범위를 크게 한 것 같다는 의문이 남아있다.
* GUI의 경우 컴포넌트들만 일을 하게 하고, 패널들은 일을 전달하는 용도로만 구현하려고 했는데, JInputPanel의 경우, setInputForm()과 getInputForm()이라는 method가 패널이 컴포넌트에게 지시를 내리도록 구현이 되어 있어서, 설계 기준에 맞지 않게 구현되었는데 해결 방법을 찾지 못하고 남겨서 아쉬웠다.
* 예전 코드를 갖다 썼기 때문에, ArrayList를 사용했는데, HashMap과 같은 key, value로 데이터를 관리하는 Collection을 사용했다면, 더 효율적으로 table을 관리할 수 있었을 것이라는 생각이 들었다.

1. **참고 자료**

* <https://docs.oracle.com/javase/tutorial/>: 오라클에서 제공하는 JAVA tutorial
* <https://docs.oracle.com/javase/8/docs/api/index.html>: JAVA SE 8 API

평가표 (즉, 리포트 작성시 체크 사항)

|  |  |  |  |
| --- | --- | --- | --- |
| 평가 항목 | 학생 자체 평가  (리포트 해당 부분 표시 및 간단한 의견) | 평가  (빈칸) | 점수  (빈칸) |
| 완성도 (동작 여부)   * 직접 입력 후, ADD 동작 - List 클릭 후, 화면 추가? * 리스트에서 과목 선택하면? * “Save" 동작? * “Load"? * 과목이 있는 상태에서 Load 하면? * 기타 비정상 동작 실험 | 테스트 항목에 표시됨. 비정상 동작에 관해서는 이전 프로젝트에서 했던 시간표 충돌이나 학점, 시간에 숫자가 아닌 값을 넣는 동작들, 잘못된 파일에 대해서는 예외처리가 되어 있음. |  |  |
| 설계 노트   * 주요 결정사항 및 근거 * 한계/문제점 * 해결 방안 - 필수내용: * 프로그램 구성(Class 구조) * member visibility * 일정 저장 형식 | 총평과 프로젝트 구현에 결정사항, 한계점 등이 기록됨.  프로그램 구성은 기록되어 있고, member visibility는 총평의 결정사항 및 한계점으로 언급함. 일정 저장 형식은 File I/O에 대한 프로젝트 구현 설명에 기록됨 |  |  |
| 리포트   * 평가자 시각으로 리포트 검 토 * 위의 평가 요소들이 명확하 게 기술되었는가? |  |  |  |
| 총평/계 |  |  |  |

* 학생 자체 평가는 점수에 반영되지 않음.
* 학생 스스로 자신의 보고서를 평가하면서, 체계적으로 프로젝트를 마무리하도록 유도하는 것이 목 적임.