

프로젝트 6 설명

핵심 사항: Course 정보들의 일관성 유지

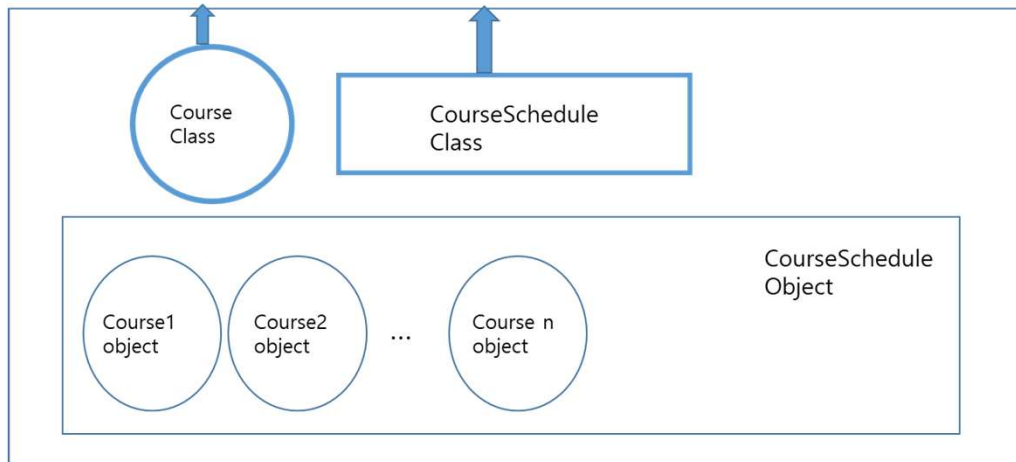
- 사용자의 action
 - ⇒ GUI의 변화
 - ⇒ 내부 Course 정보들의 갱신
- 중요한 것은 GUI의 표시되는 내용과 내부적으로 유지하는 정보가 일관성을 유지하여야 한다는 점.

Course 정보들의 일관성 유지 개념

```
// Course Schedule
// Format:: CourseName:Credit:StartDay:StartHour:Hour
CC:3:Tue:1:3
//
Java:3:Mon:1:3
//
DS:3:Tue:5:3
//
```

LOAD, SAVE button
explicitly

Course(), print() print(), size(), getCourse(i), CourseSchedule(fileName)



Update
automatically

Planning Course Schedule

과목명	학점	요일	시작교시	시간
My Courses				
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3

Show List
Add
Delete
Load
Save

Course List

List of Courses - Click to select

과목명	학점	요일	시작교시	시간
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3
OS	3	Fri	5	3

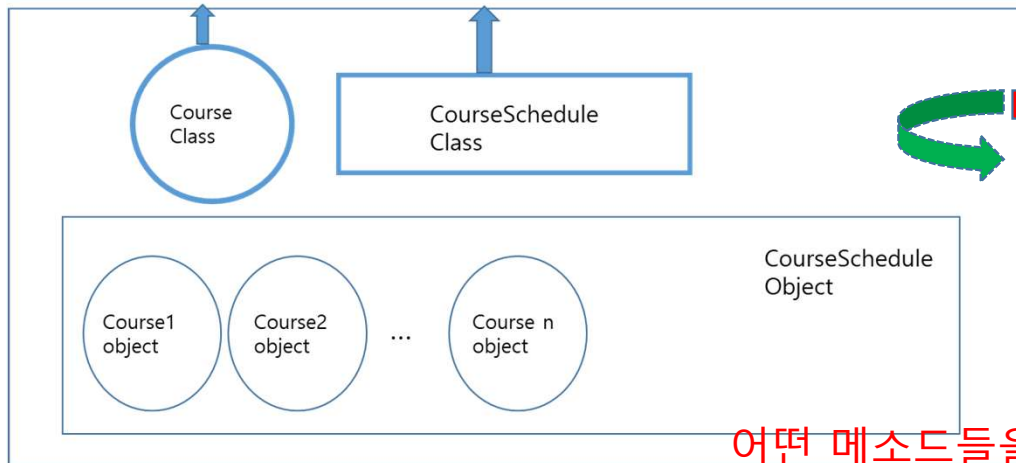
	Mon	Tue	Wed	Thr	Fri
1	Java	CC			
2	Java	CC			
3	Java	CC			
4					
5		DS			
6		DS			
7		DS			
8					
9					

Course 정보들의 일관성 유지 구현

```
// Course Schedule
// Format:: CourseName:Credit:StartDay:StartHour:Hour
CC:3:Tue:1:3
//
Java:3:Mon:1:3
//
DS:3:Tue:5:3
//
```

LOAD, SAVE button
explicitly

Course(), print() print(), size(), getCourse(i), CourseSchedule(fileName)



Planning Course Schedule

과목명	학점	요일	시작교시	시간
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3

Show List
Add
Delete
Load
Save

Course List

List of Courses - Click to select

과목명	학점	요일	시작교시	시간
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3
OS	3	Fri	5	3

	Mon	Tue	Wed	Thr	Fri
1	Java	CC			
2	Java	CC			
3	Java	CC			
4					
5		DS			
6		DS			
7		DS			
8					
9					

어떤 메소드들을 정의해서
상대방에게 서비스로 제공할 것인가?



기술적 사항

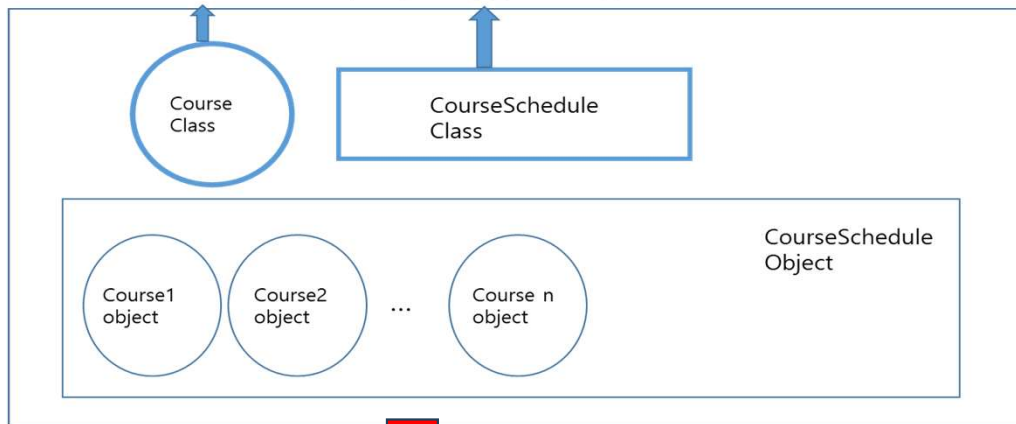
- 상대방이 제공하는 서비스를 사용하려면,
- 즉, 상대방의 메소드를 call 하려면,
- 상대방 객체의 화살표 값, 즉, reference variable 값을 알고 있어야 한다
- 이를 간단히 해결하기 위해서, 구조에 관한 고려 없이 static을 사용하면
 - 임의 접근에 대한 제어가 어렵고,
 - 프로그램 구조의 융통성이 없어질 수 있다.
- Class 간에 화살표 값을 전달하는 방법에 대한 고민이 있어야 함.

수강 신청 (Add) 처리

```
// Course Schedule
// Format:: CourseName:Credit:StartDay:StartHour:Hour
CC:3:Tue:1:3
//
Java:3:Mon:1:3
//
DS:3:Tue:5:3
//
```

LOAD, SAVE button
explicitly

Course(), print() print(), size(), getCourse(i), CourseSchedule(fileName)



- 어떤 인터페이스 메소드 정의해서 사용?
- 그 메소드는 어떻게 접근?

Planning Course Schedule

과목명	학점	요일	시작교시	시간
My Courses				
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3

Buttons: Show List, Add, Delete, Load, Save

	Mon	Tue	Wed	Thr	Fri
1	Java	CC			
2	Java	CC			
3	Java	CC			
4					
5		DS			
6		DS			
7		DS			
8					
9					

Course List

List of Courses - Click to select

과목명	학점	요일	시작교시	시간
Java	3	Mon	1	3
DS	3	Tue	5	3
CC	3	Tue	1	3
OS	3	Fri	5	3



구조의 중요성: Class화

- 화면 구성을 누가, 어디서 display?
 - mainFrame에서 statement들로? 과제 5에 한해서.
 - method로? 여전히 C-style
 - 구성 요소의 재사용 어려움 (예, 과목/강의 리스트)
 - JPanel을 확장시킨 GUI class를 아끼지 말고 사용!
- Course ↔ CoursePanel
 - 단순히 이름만 매칭시키는 것이 아님
 - 생성자 인자로!
 - method return value가 Course 객체
- 시간 여유가 있어야 구조 조정 가능. 일찍 시작이 필수!

코딩 스타일

- 평범이 우선. 개성은 나중에.
- 아직도 C 스타일
- 예측 가능한 코드
- Naming 규칙을 따르는 것이 좋음