



Object – Oriented System Design

Programming Assignment #01

● Programming Assignment

- Template : Assignment01
- 업로드 : 각자의 프로젝트 폴더 (기존의 Self-Test업로드 방식과 동일)
- Commit Message : Assignment01 (git commit -m "Assignment01")
- 제출기한 : 5월 23일 밤 12시까지

CSLAB

● TimeTable 만들기

- 학생들은 그들의 일주일 시간표를 기록하고 이용하곤 한다. 이번 과제에서 우리는 시간표 어플리케이션을 만들고 각자의 시간표를 저장할 것이다. 학생들의 스케줄은 요일과 교시에 따라 정렬된다. 요일은 월요일부터 금요일까지, 교시는 1교시부터 10교시까지 있다.

CSLAB

● Class Lecture

- `private String day` (강의 요일)
- `private int time` (강의 시간)
- `private String lectureName` (강의 이름)
- `private String professor` (교수 이름)
- `private String roomNumber` (강의 장소)
- `private int enrolled` (강의를 수강중인 학생 수)
- `private int maxEnrolled` (수강 정원)

- `public Lecture()`
 - 변수 `lectureName` = "----" , `professor` = "None" , `roomNumber` = "None"으로 초기화하는 생성자
- `public Lecture(String day, int time, String lectureName, String professor, String roomNumber, int maxEnrolled)`
 - 해당 변수들을 인자로 받은 값으로 초기화할 생성자
 - `enrolled`의 값은 0으로 초기화

CSLAB

● Class Lecture

- 각 변수들에 대한 accessor, mutator 메서드 선언
- **public boolean equals(Lecture s)** -> 같은 수업인지 확인한다
 - lectureName, professor, roomNumber 비교
- **public void incEnrolled()**
 - enrolled를 1증가시키는 메서드
- **public void decEnrolled()**
 - enrolled를 1감소시키는 메서드
- **public boolean checkLecture()** -> 강의를 추가할 수 있는지 확인하는 메서드
 - 강의명이 "----"인지 확인

CSLAB

● Class Student

- `private String studentName` (학생 이름)
- `private int studentCode` (학번)
- `Lecture timetable[][] = new Lecture[10][5]` (시간표)
- `public enum DAYS`
 - 월요일부터 금요일까지
- `public Student()`
 - 시간표를 모두 초기화하는 생성자
- `public Student(String studentName, int studentCode)`
 - 호출됨과 동시에 시간표 초기화
 - 인자로 받은 변수로 초기화하는 생성자
- 각 변수에 대한 `accessor, mutator` 선언

CSLAB

● Class Student

- **public void setScheduleInformation(Lecture lecture)**
 - 추가하려는 강의가 최대 정원이 초과되지 않았을 경우 인자로 받은 과목을 시간표에 추가.
 - Lecture 클래스의 checkLecture()클래스를 호출하여 강의를 추가할 수 있는지도 확인 할 것.
 - DAYS 활용하여 시간표 2차원 배열 index 접근할 것.
 - 현재 수강인원을 1 증가 시킴

CSLAB

● Class Student

- **public String showTimetable()**
 - 학생의 시간표를 String타입으로 리턴
 - DAYS를 활용할 것
 - 과목의 name만 출력
 - 중첩 반복문 사용
 - 아래 그림과 같이 정렬되도록 할 것.

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	OOP	----	----	----	----
2	----	security	OS	----	----
3	----	----	----	----	----
4	----	----	----	----	----
5	----	Digital logic	----	----	----
6	----	----	----	Network	Python
7	----	----	----	----	----
8	----	----	----	----	----
9	----	Ski-Board	----	----	----
10	----	----	----	----	----

● Class Student

- **public boolean equals(Student stu)**
 - 이 메서드를 호출한 객체와 인자로 받은 객체의 시간표를 비교
 - Lecture 클래스의 equals 메서드 호출
 - 학생들의 시간표를 비교하는데 사용하는 메서드
- **public void deleteSchedule(String day, int period)**
 - 인자로 받은 day와 period에 해당하는 강의 수강인원을 1 감소.
 - 인자로 받은 요일(String)과 시간(int)의 강의를 지움(default 상태로 만듦)
- **public Calendar setInputDate()**
 - Calendar 클래스를 이용
 - 키보드로부터 년, 월, 일을 입력으로 받아 Calendar클래스의 객체를 리턴

CSLAB

● Class Student

- **public String oneDaySchedule(String day)**
 - 인자로 받은 요일의 시간표만 String 형태로 리턴해주는 메서드
 - 해당 요일과 과목명만 리턴

```
MONDAY  
OOP  
--  
--  
--  
--  
--  
--  
--  
--  
--  
--  
--  
--
```

CSLAB

● Class TimeTableApp

• main 메서드

- 10개의 강의가 생성되어 있음.

```
Lecture lecture[] = new Lecture[10];
lecture[0] = new Lecture("MONDAY", 1, "OOP", "mr.park", "ITBT808", 20);
lecture[1] = new Lecture("TUESDAY", 2, "security", "mr.park", "ITBT816", 20);
lecture[2] = new Lecture("WEDNESDAY", 1, "Datastructure", "mr.park", "ITBT808", 25);
lecture[3] = new Lecture("THURSDAY", 6, "Network", "mr.yang", "ITBT503", 20);
lecture[4] = new Lecture("FRIDAY", 9, "computer architecture", "mr.han", "ITBT507", 20);
lecture[5] = new Lecture("TUESDAY", 5, "Digital logic", "mr.kim", "ITBT401", 15);
lecture[6] = new Lecture("WEDNESDAY", 2, "OS", "mr.yoon", "ITBT606", 15);
lecture[7] = new Lecture("THURSDAY", 3, "C-language", "mr.choi", "ITBT503", 20);
lecture[8] = new Lecture("FRIDAY", 6, "Python", "mr.han", "ITBT504", 15);
lecture[9] = new Lecture("TUESDAY", 9, "Ski-Board", "mr.kang", "ITBT404", 5);
```

- 스캐너를 통하여 학생수를 입력받고 학생수만큼 각각 이름과 학번을 입력받는다

```
학생수를 입력하세요
3
1 학생의 이름을 입력하세요
Wooye1
1 학생의 학번을 입력하세요
20191123
2 학생의 이름을 입력하세요
TheQ
2 학생의 학번을 입력하세요
20191124
3 학생의 이름을 입력하세요
Beenzino
3 학생의 학번을 입력하세요
20201010
```

SLAB

● Class TimeTableApp

• main 메서드

- 메인 메뉴 생성 (무한 반복문 사용)

학생의 시간표를 입력하려면 1번을 누르세요
특정 과목을 수강하는 학생수를 출력하려면 2번을 누르세요
학생들의 시간표를 비교하려면 3번을 누르세요.
학생의 시간표를 지우려면 4번을 누르세요
시간표를 출력하려면 5번을 누르세요
해당 학생의 특정 날짜 시간표를 출력하려면 6번을 누르세요
프로그램을 종료하려면 7번을 누르세요

• case 1

- 시간표를 추가하려는 학생의 학번을 입력으로 받아 추가할 수 있는 과목을 보여주고 과목 번호를 입력 시 시간표에 과목 추가
- setScheduleInformation()메서드 호출하여 활용

CSLAB

● Class TimeTableApp

• Case 1 예시

시간표를 입력할 학생의 학번을 입력하세요

20191123

다음 중 추가할 시간표의 번호를 입력하세요

- (1) OOP - 월요일 - 1교시
- (2) security - 화요일 - 2교시
- (3) Datastructure - 수요일 - 1교시
- (4) Network - 목요일 - 6교시
- (5) computer architecture - 금요일 - 8교시
- (6) Digital design - 화요일 - 5교시
- (7) OS - 수요일 - 2교시
- (8) C-language - 목요일 - 3교시
- (9) Python - 금요일 - 6교시
- (10) Ski-Board - 화요일 - 9교시

1

시간이 정상적으로 추가되었습니다.

LAB

● Class TimeTableApp

• Case 1-2 예시

- 수강정원이 5명인 강의의 수강인원이 5명인 경우 해당 강의 추가 시도시 아래와 같이 메시지를 출력.

```
(1) OOP- 월요일 - 1교시  
(2) security - 화요일 - 2교시  
(3) Datastructure - 수요일 - 1교시  
(4) Network - 목요일 - 6교시  
(5) computer architecture - 금요일 - 8교시  
(6) Digital design - 화요일 - 5교시  
(7) OS - 수요일 - 2교시  
(8) C-language - 목요일 - 3교시  
(9) Python - 금요일 - 6교시  
(10) Ski-Board - 화요일 - 9교시
```

10

강의를 추가할 수 없습니다. : 최대 수강 인원 초과

AB

● Class TimeTableApp

• case 2

- 과목명을 입력으로 받아 해당 과목을 몇명이 수강하는지 출력
- Lecture 배열을 이용할 것
- 출력 결과

```
과목의 이름을 입력하세요
```

```
OOP
```

```
OOP 과목을 수강하는 학생의 수는 2
```

CSLAB

● Class TimeTableApp

• case 3

- 2명의 학생의 학번을 키보드로 입력받아 시간표가 같은지 비교
- Student클래스의 equals()메서드 활용
- 출력 예시

```
시간표를 비교할 첫번째 학생의 학번을 입력하세요  
20191123  
두번째 학생의 학번을 입력하세요  
20201010  
두 학생의 시간표는 다릅니다.
```

CSLAB

● Class TimeTableApp

• case 4

- 시간을 지울 학생의 학번을 키보드로 입력 받음.
- 시간을 지울 요일과 시간을 키보드로 입력 받음.
- Student클래스의 deleteSchedule() 메서드 호출하여 삭제
- 삭제 후 Student 클래스의 showTimetable()메서드 호출하여 시간표 출력

```
4
시간표를 삭제할 학생의 학번을 입력하세요
20191123
지을 시간표의 요일을 입력하세요
monday
지을 시간표의 시간을 입력하세요
1
|
|          MONDAY          TUESDAY          WEDNESDAY          THURSDAY          FRIDAY
1          ----          ----          ----          ----          ----
2          ----          ----          ----          ----          ----
3          ----          ----          ----          ----          ----
4          ----          ----          ----          ----          ----
5          ----          ----          ----          ----          ----
6          ----          ----          ----          ----          ----
7          ----          ----          ----          ----          ----
8          ----          ----          ----          ----          ----
9          ----          ----          ----          ----          ----
10         ----          ----          ----          ----          ----
```

• Class TimeTableApp

• case 5

- 학생의 학번을 입력받아 해당 학생의 시간표를 출력하는 메서드
- Student클래스의 showTimeTable() 메서드 호출

5
시간표를 출력할 학생의 학번을 입력하세요
20191123

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
1	----	----	----	----	----
2	----	----	----	----	----
3	----	----	----	----	----
4	----	----	----	----	----
5	----	----	----	----	----
6	----	----	----	Network	----
7	----	----	----	----	----
8	----	----	----	----	----
9	----	----	----	----	----
10	----	----	----	----	----

CSLAB

● Class TimeTableApp

• case 6

- 학생의 학번과 년 월 일을 키보드로 입력받아 해당 요일의 시간표만 출력
- Student클래스의 setInputDate()메서드를 호출하여 Calendar객체에 날짜 정보 저장
- oneDaySchedule()메서드를 호출하여 특정 날짜의 시간표를 출력
- Calendar클래스의 메서드는 문서 마지막 참조

```
6
출력할 학생의 학번을 입력하세요
20191123
Enter the date:
20200514
|
THURSDAY
----
----
----
----
----
Network
----
----
----
----
```

SLAB

● Class TimeTableApp

- case 7

- 프로그램 종료

- default

- Try again 이란 메시지를 출력하며 메뉴 선택창 출력

```
10
```

```
Try again|
```

학생의 시간표를 입력하려면 1번을 누르세요

특정 과목을 수강하는 학생수를 출력하려면 2번을 누르세요

학생들의 시간표를 비교하려면 3번을 누르세요.

학생의 시간표를 지우려면 4번을 누르세요

시간표를 출력하려면 5번을 누르세요

해당 학생의 특정 날짜 시간표를 출력하려면 6번을 누르세요

프로그램을 종료하려면 7번을 누르세요

CSLAB

● Class TimeTableApp

- **private static void showClass()**

- 현재 추가할 수 있는 강의를 단순히 콘솔에 출력해주는 메서드

```
(1) OOP - 월요일 - 1교시  
(2) security - 화요일 - 2교시  
(3) Datastructure - 수요일 - 1교시  
(4) Network - 목요일 - 6교시  
(5) computer architecture - 금요일 - 8교시  
(6) Digital design - 화요일 - 5교시  
(7) OS - 수요일 - 2교시  
(8) C-language - 목요일 - 3교시  
(9) Python - 금요일 - 6교시  
(10) Ski-Board - 화요일 - 9교시
```

● Calendar 클래스

- Calendar 클래스는 자바에서 날짜와 시간에 관한 데이터를 손쉽게 처리할 수 있도록 제공하는 추상 클래스. 이러한 Calendar 클래스에는 날짜와 시간을 처리하기 위한 다양한 필드와 메소드가 포함. Calendar 클래스를 사용하기 위해선 'java.util.Calendar;'를 import해야 함

CSLAB

● Calendar 클래스

• Calendar 클래스의 주요 상수

상수	사용방법	설명
static int YEAR	Calendar.YEAR	현재 년도를 가져온다.
static int MONTH	Calendar.MONTH	현재 월을 가져온다. (1월은 0)
static int DATE	Calendar.DATE	현재 월의 날짜를 가져온다.
static int WEEK_OF_YEAR	Calendar.WEEK_OF_YEAR	현재 년도의 몇째 주
static int WEEK_OF_MONTH	Calendar.WEEK_OF_MONTH	현재 월의 몇째 주
static int DAY_OF_YEAR	Calendar.DAY_OF_YEAR	현재 년도의 날짜
static int DAY_OF_MONTH	Calendar.DAY_OF_MONTH	현재 월의 날짜 (DATE와 동일)
static int DAY_OF_WEEK	Calendar.DAY_OF_WEEK	현재 요일 (일요일은 1, 토요일은 7)
static int HOUR	Calendar.HOUR	현재 시간 (12시간제)
static int HOUR_OF_DAY	Calendar.HOUR_OF_DAY	현재 시간 (24시간제)
static int MINUTE	Calendar.MINUTE	현재 분
static int SECOND	Calendar.SECOND	현재 초

● Calendar 클래스

● Calendar 클래스 메소드

Calendar 클래스의 메소드	설명
boolean after(Object when)	when과 비교하여 현재 날짜 이후이면 true, 아니면 false를 반환한다.
boolean before(Object when)	when과 비교하여 현재 날짜 이전이면 true, 아니면 false를 반환한다.
boolean equals(Object obj)	같은 날짜값인지 비교하여 true, false를 반환한다.
int get(int field)	현재 객체의 주어진 값의 필드에 해당하는 상수 값을 반환한다. 이 상수값은 Calendar 클래스의 상수중에 가진다.
static Calendar getInstance()	현재 날짜와 시간 정보를 가진 Calendar 객체를 생성한다.
Date getTime()	현재의 객체를 Date 객체로 변환한다.
long getTimeInMillis()	객체의 시간을 1/1000초 단위로 변경하여 반환한다.
void set(int field, int value)	현재 객체의 특정 필드를 다른 값으로 설정한다.
void set(int year, int month, int date)	현재 객체의 년, 월, 일 값을 다른 값으로 설정한다.
void set(int year, int month, int date, int hour, int minute, int second)	현재 객체의 년, 월, 일, 시, 분, 초 값을 다른 값으로 설정한다.
void setTime(Date date)	date 객체의 날짜와 시간 정보를 현재 객체로 생성한다.
void setTimeInMillis(long mills)	현재 객체를 1/1000초 단위의 주어진 매개변수 시간으로 설정한다.
int getActualMinimum(int field)	현재 객체의 특정 필드의 최소 값을 반환한다.
int getActualMaximum(int field)	현재 객체의 특정 필드의 최대 값을 반환한다.