

Report

전자공학도의 윤리 강령 (IEEE Code of Ethics)

(출처: <http://www.ieee.org>)

나는 전자공학도로서, 전자공학이 전 세계 인류의 삶에 끼치는 심대한 영향을 인식하여 우리의 직업, 동료와 사회에 대한 나의 의무를 짊어 있어 최고의 윤리적, 전문적 행위를 수행할 것을 다짐하면서, 다음에 동의한다.

1. **공중의 안전, 건강 복리에 대한 책임:** 공중의 안전, 건강, 복리에 부합하는 결정을 할 책임을 질 것이며, 공중 또는 환경을 위협할 수 있는 요인을 신속히 공개한다.
2. **지위 남용 배제:** 실존하거나 예기되는 이해 상충을 가능한 한 피하며, 실제로 이해가 상충할 때에는 이를 이해 관련 당사자에게 알린다. (이해 상충: conflicts of interest, 공적인 지위를 사적 이익에 남용할 가능성)
3. **정직성:** 청구 또는 견적을 함에 있어 입수 가능한 자료에 근거하여 정직하고 현실적으로 한다.
4. **뇌물 수수 금지:** 어떠한 형태의 뇌물도 거절한다.
5. **기술의 영향력 이해:** 기술과 기술의 적절한 응용 및 잠재적 영향에 대한 이해를 높인다.
6. **자기계발 및 책무성:** 기술적 능력을 유지, 증진하며, 훈련 또는 경험을 통하여 자격이 있는 경우이거나 관련 한계를 전부 밝힌 뒤에만 타인을 위한 기술 업무를 수행한다.
7. **엔지니어로서의 자세:** 기술상의 업무에 대한 솔직한 비평을 구하고, 수용하고, 제공하며, 오류를 인정하고 수정하며, 타인의 기여를 적절히 인정한다.
8. **차별 안하기:** 인종, 종교, 성별, 장애, 연령, 출신국 등의 요인에 관계없이 모든 사람을 공평하게 대한다.
9. **도덕성:** 허위 또는 악의적인 행위로 타인, 타인의 재산, 명예, 또는 취업에 해를 끼치지 않는다.
10. **동료애:** 동료와 협력자가 전문분야에서 발전하도록 도우며, 이 윤리 헌장을 준수하도록 지원한다.

위 IEEE 윤리헌장 정신에 입각하여 report를 작성하였음을 서약합니다.

학 부: 전자공학부

제출일: 2024/04/09

과목명: 임베디드시스템 설계

교수명: 김영진

분 반: 수D 금D

학 번: 201620698 / 202126946

성 명: 김진태 / 이도윤

과제 1: Python Numpy 및 함수 기반의 수강 신청 프로그램 작성

1. 과제 소개

이번 과제는 Python을 통한 프로그래밍에 익숙해지는 것과 Python의 라이브러리 Numpy의 사용법을 익히는 것을 목표로 한다.

2. 프로젝트 개요

1) 프로젝트 소개

주어진 프로젝트는 수강 신청 프로그램을 개발하는 것이다. 수강 신청 프로그램에는 다음과 같은 요구 사항이 주어 져 있다.

- 로그인 및 사용자 인증 기능
- 관리자 기능
- 수강 신청자 기능

로그인 및 사용자 인증 기능은 사용자에게 사용자 이름과 비밀번호를 입력받아 사용자 유형을 판별하는 기능이다. 사용자 유형은 두 가지로, 관리자와 수강 신청자로 나뉘어진다.

프로그램 사용자가 관리자일 경우 과목 추가, 과목 삭제, 과목 정보 수정을 할 수 있다. 프로그램 사용자가 수강 신청자일 경우 관리자가 할 수 있는 행위는 할 수 없지만, 수강 과목 조회 및 과목 신청, 그리고 본인이 신청한 과목에 대한 요일별 수업 스케줄 확인이 가능하다.

2) 사용된 기술 및 라이브러리

- Python 프로그래밍 언어
- Numpy 라이브러리

Python 프로그래밍 언어를 기반으로 프로그램을 작성하며, Numpy를 통해 row와 column이 있는 데이터를 다루는 것을 목표로 한다. Numpy는 주로 선형대수학의 벡터를 포함한 수학적 연산에 사용되지만 이번 과제에서는 수학적 연산보다는 row와 column이 있는 데이터를 다루는 데에 목적을 둔다. 예를 들어 해당 프로젝트에서 수강 과목 데이터 저장을 Numpy를 통해 한다.

3. 프로그램 설계

1) 전반적인 구조

프로그램의 코드는 모듈화되어 있다. 코드의 전반적인 구조는 다음과 같다.

1. Main 함수

Main 함수를 통해 프로그램이 실행된다. Main 함수 내에서 사용자의 입력을 받고, 입력받은 내용에 따라 적절한 기능을 호출한다.

2. 관리자 기능 클래스 - CourseAdminSystem

과목을 추가, 삭제, 수정하는 기능을 담당한 클래스이다.

3. 수강자 기능 클래스 - CourseRegistrationSystem

과목을 조회하고, 수강 신청을 하거나 신청한 과목을 취소하는 기능을 담당하는 클래스이다.

4. 과목 데이터 관리 클래스 - CourseList

과목 데이터를 관리하는 클래스이다. 파일에서 과목 데이터를 로드하거나, 파일에 과목 데이터를 저장하는 기능을 수행한다.

5. 그 외 함수

is_student, is_admin과 같은 함수가 구현되어 있다.

6. 그 외 클래스

수업 시간이 적절한 입력인지 확인할 때 사용하는 클래스인 LectureTimeValidator가 구현되어 있다. 해당 클래스는 'Wed#D1#Thu#E1'과 같은 수업 시간 문자열을 전달받아 적절한 수업 시간 문자열인지 확인한다.

2) 함수 설명

1. Main 함수

Main 함수는 프로그램의 진입점이 되는 함수이다. Main 함수 내에서 사용자의 입력을 받고, 입력받은 내용에 따라 적절한 기능을 호출한다. 사용자의 선택에 따라 관리자 로그인, 수강자 로그인, 수강자 등록, 프로그램 종료를 할 수 있다. 각각의 기능은 해당 기능을 담당한 클래스를 호출하여 수행한다.

2. is_admin 함수

프로그램 사용자가 관리자로 로그인하려 할 때, 입력한 사용자 이름과 비밀번호가 관리자인지 판단하는 함수이다. 사용자 이름이 root이고 비밀번호가 0000이면 관리자로 판단하여 True를 return하고 root,0000이 아니면 False를 return한다.

3. is_student 함수

프로그램 사용자가 수강자로 로그인하려 할 때, 입력한 사용자 이름과 비밀번호가 수강자인지 판단하는 함수이다. 수강자 목록에 사용자 이름과 비밀번호와 일치하는 정보가 있다면 수강자로 판단하여 해당 수강자의 수강시스템 인스턴스를 return하고, 그 외의 입력에 대해서는 None를 return한다.

3) 클래스 설명

1. 관리자 기능 클래스 - CourseAdminSystem

과목을 추가, 삭제, 수정하는 기능을 담당하는 클래스이다. 해당 클래스의 인스턴스를 생성할 때, CourseList 클래스의 인스턴스를 인자로 받는다. CourseList 클래스를 통해 과목을 추가하거나 삭제, 수정하고 해당 데이터를 basics.txt 파일에 저장한다. 이 때, 과목을 추가할 때에는 중복된 과목을 추가할 수 없도록 구현되어 있다.

2. 수강자 기능 클래스 - CourseRegistrationSystem

과목을 조회하고, 수강 신청을 하거나 신청한 과목을 취소하는 기능을 담당하는 클래스이다. 수강 신청할 때에 수강자가 이미 신청한 과목인지와 시간표와 중복되지 않는지 확인하며, 최대 학점인 17학점이 넘게 수강 신청할 수 없도록 구현되어 있다. 수강을 취소할 때에는 해당 과목을 수강자가 신청하였는지 먼저 확인하고, 신청한 과목일 때에만 취소하도록 구현되어 있다. 또한 신청과 취소 모두 로그인 후 5분경과시 수행할 수 없으며 이를 위하여 시간을 갱신하는 기능 또한 구현하였다.

3. 과목 데이터 관리 클래스 - CourseList

과목 데이터를 관리하는 클래스이다. 같은 폴더 내에 있는 basics.txt 파일에서 과목 데이터를 로드한다. 만약 basics.txt 파일이 없다면 과목 데이터를 파일에서 로드하지 않고 빈 과목 목록을 생성한다. 과목 목록을 관리할 때에는 Numpy의 배열을 사용한다.

4) 데이터 구조 설명

1. Numpy 배열

해당 프로젝트에서는 Numpy 배열을 사용하여 과목 목록을 데이터화하고 처리한다. CourseList 클래스 내에서 과목 목록을 저장할 때에 Numpy 배열이 사용되며, np.loadtxt 함수를 통해 basics.txt 파일에서 과목 목록을 읽고, np.savetxt 함수를 통해 basics.txt 파일에 과목 목록을 저장한다.

2. 2차원 배열

해당 프로젝트에서 사용하는 Numpy 배열은 2차원 배열이다. 과목 목록을 저장하는 CourseList의 items 프로퍼티, 사용자의 시간표를 저장하는 CourseRegistrationSystem의 user_schedule 프로퍼티는 모두 2차원 배열로 구성되어 있다. 이 중에서 user_schedule 프로퍼티는 row가 시간대(예: A교시)이고 column이 요일(예: Mon)을 나타낸다.

3. 파이썬의 List 타입

사용자의 아이디와 비밀번호를 저장할 때에 student_list라는 파이썬의 List 타입으로 구현된 값을 사용한다.

4. Set 타입

CourseRegistrationSystem 클래스에서 이미 수강한 과목인지 확인하기 위해 과목 번호를 담는 Set 타입의 course_numbers_set 프로퍼티가 있다. 해당 프로퍼티 내에 수강 과목 번호가 담겨있다면 이미 수강한 과목이고, 담겨있지 않다면 아직 수강하지 않은 과목으로 처리하도록 구현하였다. 해당 기능을 List가 아닌 Set 타입으로 구현한 이유는 중복되는 값을 허용하지 않는다는 최적화된 타입이 Set 타입이기 때문이다.

4. 주요 기능 설명

1) 로그인 및 사용자 인증

로그인 및 사용자 인증과 관련된 함수는 다음 함수들이 구현되어 있다.

1. main

메인 함수에서 사용자의 입력을 받아 관리자로 로그인할지, 수강자로 로그인할지 아니면 수강자 등록을 할지를 정할 수 있다. 각 선택에 따라 적절한 사용자 이름과 비밀번호가 들어왔는지 확인하고, 적절하지 않은 사용자 이름 또는 비밀번호가 들어왔을 경우 다시 입력을 받도록 구현되어 있다. 이 때 적절한 사용자 이름과 비밀번호인지 확인할 때 수강자의 경우 `is_student`가 관리자의 경우 `is_admin` 함수가 사용된다.

2. is_student

`is_student` 함수는 `main` 함수에서 사용되는 함수로, 사용자가 수강자로 로그인을 선택하였을 때 입력한 사용자 이름과 비밀번호가 적합한지 판단하는 함수이다. 해당 함수는 사용자 이름과 비밀번호가 수강자 목록의 사용자 이름과 비밀번호 중 일치하는 것이 있는지 확인하고, 둘 모두 일치할 때에만 적합하다고 판단하고 수강자 시스템 인스턴스를 반환하고 둘 중 하나라도 일치하지 않으면 `None`을 반환한다.

3. is_admin

`is_admin` 함수는 `main` 함수에서 사용되는 함수로, 사용자가 관리자로 로그인하였을 때 입력한 사용자 이름과 비밀번호가 적합한지 판단하는 함수이다. 해당 함수는 사용자 이름이 `root`이고 비밀번호가 `0000`일 때에만 적합하다고 판단하고 그에 따라 `True, False`를 반환한다.

2) 수강 과목 목록 관리 기능

수강 과목 목록 관리 기능을 수행하는 `CourseList` 클래스에는 다음과 같은 함수들이 구현되어 있다.

1. load_courses

`load_courses` 메서드는 `basics.txt` 파일의 내용을 읽어와 `self.items`에 `ndarray` 타입으로 값을 지정해주는 메서드이다. `basics.txt` 파일이 경로에 존재하지 않을 경우 과목을 빈 배열로 초기화한다.

2. append_course

`append_course` 메서드는 새로운 과목을 `self.items` 마지막에 추가해주는 메서드이다.

3. find_course_index

`find_course_index` 메서드는 과목 번호를 인자로 전달받아 `self.items`에 일치하는 과목이 있는지 찾는 메서드이다. 일치하는 과목이 있다면 해당 과목의 인덱스를 `return`하고, 없다면 `None`을 `return`한다.

4. find_course

`find_course` 메서드는 과목 번호를 인자로 전달받아 `self.items`에 일치하는 과목이 있는지 찾아 `return`하는 메서드이다. 일치하는 과목이 없으면 `None`을 전달한다. 내부적으로는 `find_course_index` 함수를 사용해 과목의 인덱스를 찾고, `self.items[index]`를 `return`하도록 구현되어 있다.

5. has_course

`has_course` 메서드는 `self.items`에 과목 번호와 일치하는 과목이 있는지 찾는 메서드이다. 일치하는 과목이 있으면 `True`를, 없으면 `False`를 `return`한다.

6. delete_course

`delete_course` 메서드는 `self.items`에서 과목 번호와 일치하는 과목을 찾고 삭제하는 메서드이다. 만약 삭제할 과목이 없다면 과목을 찾을 수 없음을 출력한다.

7. edit_course

`edit_course` 메서드는 과목 번호와 일치하는 과목을 수정하는 메서드이다. 해당 메서드에서는 사용자의 입력을 받아 과목 번호, 과목명, 전필여부, 학점, 수업 시간, 수업 장소 등을 수정한다.

8. show_courses

`show_courses` 메서드는 등록된 과목들을 보여주는 메서드이다. 내부적으로는 `print(self.items)` 한 줄로 구현되어 있다.

2) 관리자 기능

관리자 기능을 수행하는 `CourseAdminSystem` 클래스에는 다음과 같은 함수들이 구현되어 있다.

1. insert_courses

`insert_courses` 메서드는 총 5개의 특정 과목을 `course_list`에 추가하는 메서드이다. 만약 과목 번호가 중복되는 과목

이 있다면 해당 과목은 추가하지 않는다. 내부적으로는 CourseList 클래스의 append_course 메서드를 활용한다.

2. delete_course

delete_course 메서드는 사용자에게 삭제하려는 과목 번호를 입력 받고 해당 과목을 과목 목록에서 삭제하는 메서드이다. 내부적으로는 CourseList 클래스의 delete_course 메서드를 활용한다.

3. edit_course

edit_course 메서드는 사용자에게 변경하려는 과목 번호를 입력받아 과목의 정보를 변경하는 메서드이다. 내부적으로는 CourseList 클래스의 edit_course 메서드를 활용한다.

4. run

run 메서드는 CourseAdminSystem 클래스를 통해 관리자가 작업을 수행할 수 있도록 작업을 시작하는 메서드이다. 해당 메서드를 실행하여 CLI 프로그램 사용자는 과목 추가, 삭제, 정보 변경을 할 수 있게 된다.

3) 수강 신청자 기능

수강 신청자 기능을 수행하는 CourseList 클래스에는 다음과 같은 함수들이 구현되어 있다.

1. show_all_courses

show_all_courses 메서드는 시스템에 등록된 모든 수업을 보여주는 메서드이다. 내부적으로는 print(self.course_list.items)를 출력하도록 구현되어 있다.

2. register_course

register_course 메서드는 수강자가 과목을 신청할 때 사용하는 메서드이다. 해당 메서드에서 신청하려는 과목 번호를 입력 받고, 수강자의 시간표에서 다른 과목과 시간대가 겹치거나, 최대 신청 학점인 17학점을 넘어서거나, 로그인 시간이 5분을 지난 경우 과목을 신청할 수 없도록 구현되어 있다. 만약 과목을 신청하는 데에 문제가 없다면 과목이 수강된다.

3. is_enrollment_allowed

is_enrollment_allowed 메서드는 해당 과목과 시간이 겹치는 다른 과목이 없는지 판단하는 메서드이다. 겹치는 시간이 있으면 True를, 없으면 False를 return한다.

4. cancel_course

cancel_course 메서드는 과목 신청을 취소하는 메서드이다. 사용자로부터 취소하려는 과목 번호를 입력받으며, 취소하려는 과목을 신청한 적이 없거나 로그인한 시간으로부터 5분이 지난 경우 과목을 취소할 수 없다. 그 외의 경우에는 과목 신청이 취소된다.

5. show_user_schedule

show_user_schedule 메서드는 수강자의 시간표를 보여주는 메서드이다. 월, 화, 수, 목, 금 순으로 신청한 과목의 이름과 장소, 그리고 해당 과목의 요일이 표기된다.

6. time_reset

time_reset 메서드는 수강자의 로그인 시간을 0초로 다시 갱신하는 메서드이다.

7. run

run 메서드는 CourseRegistrationSystem 클래스를 통해 수강자가 작업을 수행할 수 있도록 작업을 시작하는 메서드이다. 해당 메서드를 실행하여 CLI 프로그램 사용자는 과목 조회, 신청, 취소 등을 할 수 있게 되며 실행시 로그인시간을 초기화한다.

5. 프로그램 실행 예시

1) 시작 화면

```
프로그램 종료 : 0
관리자 로그인 : 1
수강자 로그인 : 2
수강자 등록 : 3

입력 : █
```

프로그램의 시작 화면이다. 이미지에 나온 것처럼 관리자로 로그인하려면 1을, 수강자로 로그인하려면 2를, 새로운 수강자 계정을 등록하려면 3을, 프로그램을 종료하려면 0을 입력하면 된다.

```
프로그램 종료 : 0
관리자 로그인 : 1
수강자 로그인 : 2
수강자 등록 : 3

입력 : 0
프로그램을 종료합니다 .
```

0을 입력하면 위 이미지와 같이 프로그램을 종료하며, 1, 2, 3을 입력하면 각각 관리자 시스템, 수강자 시스템, 수강자 등록이 실행된다.

```
프로그램 종료 : 0
관리자 로그인 : 1
수강자 로그인 : 2
수강자 등록 : 3

입력 : admin
```

그 외의 입력이 들어오면 위와 같이 잘못된 입력임을 표기한 후 다시 한 번 입력을 받는다.

2) 관리자 시스템 화면

```
입력 : 1
사용자 이름을 입력해주세요 : root
비밀번호를 입력해주세요 : 0000

입력한 정보가 틀립니다 .

프로그램 종료 : 0
관리자 로그인 : 1
수강자 로그인 : 2
수강자 등록 : 3
```

관리자 시스템을 실행하면 프로그램 사용자에게 사용자 이름과 비밀번호를 입력받는다. 이 때, 관리자의 사용자 이름과 비밀번호는 root, 0000으로 고정되어 있으며 사용자 이름이나 비밀번호가 일치하지 않으면 로그인되지 않고 초기 화면으로 돌아간다.

```
입력 : 1
사용자 이름을 입력해주세요 : root
비밀번호를 입력해주세요 : 0000
관리자로 로그인하셨습니다 . 수행하고자 하는 작업을 입력해주세요 .
0- 과목 추가 , 1- 과목 삭제 , 2- 과목 정보 변경 , 3- 개설 과목 조회 , 4- 작업 종료 . 입력하기 : █
```

위 이미지와 같이 정확한 사용자명 root와 비밀번호 0000을 입력하면 관리자 기능을 사용할 수 있다.

```
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 3
[['100' 'LogicCircuit' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#535']
['101' 'CapstoneDesign1' 'Required' '2' 'Fri#F4' 'Woncheon#538']
['102' 'DataStructure' 'Required' '3' 'Mon#C1#Thu#C1' 'Woncheon#340']
['104' 'CircuitTheory' 'Required' '3' 'Wed#B1#Thu#D1' 'Woncheon#535']
['105' 'DigitalSystemDesign' 'Elective' '3' 'Tue#E1#Thu#E1'
'Woncheon#228']
['108' 'EmbeddedSystemDesign' 'Elective' '3' 'Wed#D1#Fri#D1'
'Woncheon#538']
['109' 'SignalsSystems' 'Required' '3' 'Tue#A1#Thu#D1' 'Woncheon#535']
['110' 'ElectronicCircuit1' 'Required' '3' 'Mon#C1#Wed#C1'
'Woncheon#340']
['111' 'Electromagnetics' 'Required' '3' 'Mon#A1#Thu#D1' 'Woncheon#539']
['112' 'BasicLab' 'Required' '2' 'Fri#F4' 'Woncheon#334']
['113' 'Robotics' 'Elective' '3' 'Mon#B1#Wed#C1' 'Woncheon#538']]
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: █
```

위 화면은 과목 목록을 basics.txt 파일에서 읽어온 초기 상태의 과목 목록을 보여주는 모습이다. 관리자 시스템에 3번을 입력하여 개설 과목을 조회하였다.

```
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 0
과목이 추가되었습니다.
과목이 추가되었습니다.
과목이 추가되었습니다.
과목이 추가되었습니다.
과목이 추가되었습니다.
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 3
[['100' 'LogicCircuit' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#535']
['101' 'CapstoneDesign1' 'Required' '2' 'Fri#F4' 'Woncheon#538']
['102' 'DataStructure' 'Required' '3' 'Mon#C1#Thu#C1' 'Woncheon#340']
['104' 'CircuitTheory' 'Required' '3' 'Wed#B1#Thu#D1' 'Woncheon#535']
['105' 'DigitalSystemDesign' 'Elective' '3' 'Tue#E1#Thu#E1'
'Woncheon#228']
['108' 'EmbeddedSystemDesign' 'Elective' '3' 'Wed#D1#Fri#D1'
'Woncheon#538']
['109' 'SignalsSystems' 'Required' '3' 'Tue#A1#Thu#D1' 'Woncheon#535']
['110' 'ElectronicCircuit1' 'Required' '3' 'Mon#C1#Wed#C1'
'Woncheon#340']
['111' 'Electromagnetics' 'Required' '3' 'Mon#A1#Thu#D1' 'Woncheon#539']
['112' 'BasicLab' 'Required' '2' 'Fri#F4' 'Woncheon#334']
['113' 'Robotics' 'Elective' '3' 'Mon#B1#Wed#C1' 'Woncheon#538']
['120' 'AutomaticControl' 'Elective' '3' 'Mon#B1#Thu#B1' 'Woncheon#539']
['130' 'Multimedia' 'Elective' '3' 'Wed#B1#Fri#B1' 'Woncheon#540']
['140' 'ComputerArchitecture' 'Elective' '3' 'Mon#E1#Thu#E1'
'Woncheon#510']
['150' 'Probability' 'Elective' '3' 'Mon#D1#Wed#D1' 'Woncheon#228']
['160' 'ElectronicCircuit1' 'Required' '3' 'Wed#B1#Fri#B1'
'Woncheon#538']]
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: █
```

관리자 시스템에서 과목 추가를 선택한 후에 개설된 과목을 살펴보면 위와 같이 120, 130, 140, 150, 160 과목 번호에 해당되는 5개의 과목이 추가된 것을 확인 할 수가 있다.

```
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
160 test Elective 3 Wed#B1#Fri#B1 Woncheon#538
이미 존재하는 과목번호 입니다
```

과목을 추가할 때에는 프로젝트에서 교수님이 정해주신 120, 130, 140, 150, 160번에 해당하는 강의를 추가하는 방법과, 직접 강의의 정보를 입력하여 추가하는 방법이 있다. 두 가지 경우 모두 과목 번호가 이미 존재하는 경우 과목이 추가되지 않는다. 위의 이미지는 직접 과목 정보를 입력하여 추가하는 경우이며, 160번 과목 번호가 이미 추가되어 있어 추가하지 못하는 모습이다.


```

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
aa test Elective 3 Wed#B1#Fri#B1 Woncheon#538
과목 번호 입력이 숫자가 아닙니다. 과목 번호는 숫자로 입력해야 합니다.
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
170 test Electiv 3 Wed#B1#Fri#B1 Woncheon#538
전필여부의 입력이 잘못되었습니다.
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
170 test Elective a Wed#B1#Fri#B1 Woncheon#538
학점 입력이 숫자가 아닙니다. 학점은 숫자로 입력해야 합니다.
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
170 test Elective 3 Wed#B1#Fri# Woncheon#538
수업 시간은 다음과 같은 포맷으로 입력해야 합니다. 예) Mon#D1#Wed#D1

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 0
0 : 추가할 과목 불러오기, 1 : 직접 정보 입력
입력: 1
과목번호 과목명 전필여부 학점 시간 장소
170 test Elective 3 Wed#B1#Fri#B1 Woncheon#538
수업 장소는 다음과 같은 포맷으로 입력해야 합니다. 예) Woncheon#538

```

위 이미지에서는 직접 추가할 과목에 대해 포맷이 맞지 않게 입력한 경우 입력을 다시 받는 것을 보여주고 있다. 직접 추가할 과목 정보를 입력할 때에 위 이미지와 같이 각각의 형식들과 다르게 입력을 한 경우에도 과목은 추가되지 않는다. 직접 과목 정보를 입력할 때에는 전필여부가 'Elective' 또는 'Required'여야 하며, 수업 시간 또한 Mon#D1#Wed#D1과 같은 형태로 입력되어야 한다. 마찬가지로 과목 장소 또한 Woncheon#538과 같은 포맷으로 입력되어야 한다.

```

입력: 1
과목번호 과목명 전필여부 학점 시간 장소
170 test Elective 3 Wed#B1#Fri#B1 Woncheon#538
과목이 추가되었습니다

```

```

['150' 'Probability' 'Elective' '3' 'Mon#D1#Wed#D1' 'Woncheon#228']
['160' 'ElectronicCircuit1' 'Required' '3' 'Wed#B1#Fri#B1'
'Woncheon#538']
['170' 'test' 'Elective' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']]

```

위 이미지와 같이 올바른 형식을 입력하였을 때 과목이 추가되는 것을 확인할 수 있다.


```

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 2
변경하려는 과목번호를 입력해주세요: 170
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 0
과목 번호를 입력해주세요: 171
변경된 정보: ['171' 'test' 'Elective' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 1
과목명을 입력해주세요: test2
변경된 정보: ['171' 'test2' 'Elective' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: Required
입력이 잘못되었습니다. 다시 입력해주세요.
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 2
전필 여부를 입력해주세요 (Elective, Required): Required
변경된 정보: ['171' 'test2' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 3
학점을 입력해주세요: 2
변경된 정보: ['171' 'test2' 'Required' '2' 'Wed#B1#Fri#B1' 'Woncheon#538']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 4
수업 시간을 입력해주세요: Wed#B2#Fri#B2
변경된 정보: ['171' 'test2' 'Required' '2' 'Wed#B2#Fri#B2' 'Woncheon#538']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 5
수업 장소를 입력해주세요: Woncheon#535
변경된 정보: ['171' 'test2' 'Required' '2' 'Wed#B2#Fri#B2' 'Woncheon#535']
변경하고자 하는 과목 정보를 선택하여 주십시오.
0- 과목 번호, 1- 과목명, 2- 전필여부, 3- 학점, 4- 수업 시간, 5- 수업 장소, 6- 작업 종료. 입력: 6
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기:

```

```

['160' 'ElectronicCircuit1' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']
['171' 'test2' 'Required' '2' 'Wed#B2#Fri#B2' 'Woncheon#535']]

```

위 이미지는 과목 정보 변경을 수행하는 이미지이다. 과목을 직접 입력하여 추가할 때와 마찬가지로 포맷이 맞지 않으면 과목이 추가되지 않는다.

과목 정보 변경 기능에서 위 이미지와 같이 각각의 정보들을 변경하였을 때 변경이 적용되어있는 것을 확인이 가능하다.

```

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 1
삭제하려는 과목번호를 입력해주세요: 171
과목이 삭제되었습니다.
과목 번호 171을(를) 삭제하였습니다.
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 3
[['100' 'LogicCircuit' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#535']
['101' 'CapstoneDesign1' 'Required' '2' 'Fri#F4' 'Woncheon#538']
['102' 'DataStructure' 'Required' '3' 'Mon#C1#Thu#C1' 'Woncheon#340']
['104' 'CircuitTheory' 'Required' '3' 'Wed#B1#Thu#D1' 'Woncheon#535']
['105' 'DigitalSystemDesign' 'Elective' '3' 'Tue#E1#Thu#E1' 'Woncheon#228']
['108' 'EmbeddedSystemDesign' 'Elective' '3' 'Wed#D1#Fri#D1' 'Woncheon#538']
['109' 'SignalsSystems' 'Required' '3' 'Tue#A1#Thu#D1' 'Woncheon#535']
['110' 'ElectronicCircuit1' 'Required' '3' 'Mon#C1#Wed#C1' 'Woncheon#340']
['111' 'Electromagnetics' 'Required' '3' 'Mon#A1#Thu#D1' 'Woncheon#539']
['112' 'BasicLab' 'Required' '2' 'Fri#F4' 'Woncheon#334']
['113' 'Robotics' 'Elective' '3' 'Mon#B1#Wed#C1' 'Woncheon#538']
['120' 'AutomaticControl' 'Elective' '3' 'Mon#B1#Thu#B1' 'Woncheon#539']
['130' 'Multimedia' 'Elective' '3' 'Wed#B1#Fri#B1' 'Woncheon#540']
['140' 'ComputerArchitecture' 'Elective' '3' 'Mon#E1#Thu#E1' 'Woncheon#510']
['150' 'Probability' 'Elective' '3' 'Mon#D1#Wed#D1' 'Woncheon#228']
['160' 'ElectronicCircuit1' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#538']]

```

위 이미지와 같이 과목 삭제 기능에서 과목 번호를 입력하여 과목 삭제가 가능하다.

```

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 1
삭제하려는 과목번호를 입력해주세요: 170
과목이 삭제되었습니다.
과목 번호 170을(를) 찾을 수 없습니다.

```

만약 삭제하고자 하는 과목 번호가 없으면 과목이 삭제되지 않는다.

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 4

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

관리자 시스템에서 종료를 선택하면 다시 관리자 로그인, 수강자 로그인, 수강자 등록을 받는 초기 화면으로 돌아간다.

3) 수강자 등록 화면

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 3
사용자 이름을 입력해주세요: test
비밀번호를 입력해주세요: 1234

등록이 완료되었습니다.

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 3
사용자 이름을 입력해주세요: root
비밀번호를 입력해주세요: 0000

이미 존재하는 사용자 이름입니다.

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 3
사용자 이름을 입력해주세요: test2
비밀번호를 입력해주세요: 1234

등록이 완료되었습니다.

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 3
사용자 이름을 입력해주세요: test
비밀번호를 입력해주세요: 1234

이미 존재하는 사용자 이름입니다.

수강자 시스템을 이용하려면 수강자의 등록이 필요하다. 수강자를 등록할 때에는 관리자 명인 root와 겹치거나, 이미 존재하는 사용자 명일 때에는 등록이 되지 않도록 구현하였다.

4) 수강자 시스템 화면

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 2
사용자 이름을 입력해주세요: test3
비밀번호를 입력해주세요: 1234

입력하신 정보를 찾을 수 없습니다.

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

입력: 2
사용자 이름을 입력해주세요: test
비밀번호를 입력해주세요: 1232

입력하신 정보를 찾을 수 없습니다.

로그인을 시도할 때 등록되지 않은 사용자 명이나 틀린 비밀번호를 입력하면 로그인이 되지 않는다.

```

프로그램 종료 : 0
관리자 로그인 : 1
수강자 로그인 : 2
수강자 등록 : 3

입력 : 2
사용자 이름을 입력해주세요 : test
비밀번호를 입력해주세요 : 1234
수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 :

```

위와 같이 등록된 사용자 이름과 비밀번호와 일치하는 계정을 입력하면 수강자로 로그인에 성공한다.

```

수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 0
[['100' 'LogicCircuit' 'Required' '3' 'Wed#B1#Fri#B1' 'Woncheon#535']
['101' 'CapstoneDesign1' 'Required' '2' 'Fri#F4' 'Woncheon#538']
['102' 'DataStructure' 'Required' '3' 'Mon#C1#Thu#C1' 'Woncheon#340']
['104' 'CircuitTheory' 'Required' '3' 'Wed#B1#Thu#D1' 'Woncheon#535']
['105' 'DigitalSystemDesign' 'Elective' '3' 'Tue#E1#Thu#E1'
'Woncheon#228']
['108' 'EmbeddedSystemDesign' 'Elective' '3' 'Wed#D1#Fri#D1'
'Woncheon#538']
['109' 'SignalsSystems' 'Required' '3' 'Tue#A1#Thu#D1' 'Woncheon#535']
['110' 'ElectronicCircuit1' 'Required' '3' 'Mon#C1#Wed#C1'
'Woncheon#340']
['111' 'Eelctromagnetics' 'Required' '3' 'Mon#A1#Thu#D1' 'Woncheon#539']
['112' 'BasicLab' 'Required' '2' 'Fri#F4' 'Woncheon#334']
['113' 'Robotics' 'Elective' '3' 'Mon#B1#Wed#C1' 'Woncheon#538']]

```

수강자 시스템에서 과목 조회 기능 사용시 관리자 시스템에서 추가한 과목들이 추가되어 있는 것을 확인할 수 있다.

```

수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 3
Mon:
Tue:
Wed:
Thu:
Fri:
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 1
신청하려는 과목 번호를 입력해주세요 : 100
로그인 시간으로 5분이 지나 과목을 수강할 수 없습니다.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 4
시간이 갱신되었습니다

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 1
신청하려는 과목 번호를 입력해주세요 : 100
LogicCircuit 과목이 수강되었습니다. 앞으로 필수로 들어야 하는 전필 과목 수는 3개입니다.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기 : 3
Mon:
Tue:
Wed: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)
Thu:
Fri: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)

```

위의 이미지를 보면 먼저 시간표 확인 기능으로 각 요일별로 신청한 과목의 정보를 확인 할 수 있는 것을 확인할 수 있다. 그리고 신청시 로그인 시간이 지났으면 신청이 되지 않는 것 또한 확인할 수 있다.

로그인 시간인 5분이 지났을 때에 로그인 시간 갱신 기능을 사용하면 로그인 시간이 현재 시점으로 갱신된다. 해당 기능을 사용 후 다시 과목을 신청했을 때 신청이 정상적으로 동작하며 들어야 하는 전필 개수 또한 함께 출력이 되는 것을 볼 수 있다.

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 102
 DataStructure 과목이 수강되었습니다. 앞으로 필수로 들어야 하는 전필 과목 수는 1개입니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 104
 시간대가 겹쳐 과목을 신청할 수 없습니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 103
 입력하신 과목은 존재하지 않습니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 109
 SignalsSystems 과목이 수강되었습니다. 전필 과목은 최소 갯수인 4개를 충족하였습니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 105
 DigitalSystemDesign 과목이 수강되었습니다. 전필 과목은 최소 갯수인 4개를 충족하였습니다.

그리고 과목 신청 시 학점 초과, 시간표 중복, 이미 신청한 과목들에 대해서는 신청이 되지 않는 것을 확인이 가능하며 또한 존재하지 않는 과목 번호를 입력했을 경우 또한 신청이 되지 않는다.

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
 신청하려는 과목 번호를 입력해주세요: 100
 이미 신청한 과목입니다

위 이미지는 이미 신청한 과목에 대해서 중복 신청하지 못하는 모습을 보여준다.

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 2
 취소하려는 과목 번호를 입력해주세요: 100
 로그인 시간으로 5분이 지나 과목을 취소할 수 없습니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 4
 시간이 갱신되었습니다

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 2
 취소하려는 과목 번호를 입력해주세요: 100
 LogicCircuit 과목이 취소되었습니다.

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 3
 Mon: Robotics - Woncheon#538 (Mon#B1#Wed#C1) DataStructure - Woncheon#340 (Mon#C1#Thu#C1)
 Tue: SignalsSystems - Woncheon#535 (Tue#A1#Thu#D1) DigitalSystemDesign - Woncheon#228 (Tue#E1#Thu#E1)
 Wed: Robotics - Woncheon#538 (Mon#B1#Wed#C1)
 Thu: DataStructure - Woncheon#340 (Mon#C1#Thu#C1) SignalsSystems - Woncheon#535 (Tue#A1#Thu#D1) DigitalSystemDesign - Woncheon#228 (Tue#E1#Thu#E1)
 Fri: CapstoneDesign1 - Woncheon#538 (Fri#F4) CapstoneDesign1 - Woncheon#538 (Fri#F4) CapstoneDesign1 - Woncheon#538 (Fri#F4) CapstoneDesign1 - Woncheon#538 (Fri#F4)

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1

과목 취소 또한 5분이 지난 경우 실행할 수 없으며 과목을 취소하고 시간표를 확인하면, 시간표에서 해당 과목이 사라진 것을 확인할 수 있다.

0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 2
 취소하려는 과목 번호를 입력해주세요: 140
 취소하려는 과목은 신청하지 않은 과목입니다.
 0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 2
 취소하려는 과목 번호를 입력해주세요: 170
 입력하신 과목은 존재하지 않습니다.

그리고 신청하지 않은 과목이나 존재하지 않는 과목번호를 입력받았을 경우에는 과목은 삭제되지 않는다.

```

사용자 이름을 입력해주세요: test
비밀번호를 입력해주세요: 1234
수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
신청하려는 과목 번호를 입력해주세요: 102
DataStructure 과목이 수강되었습니다. 앞으로 밑수로 들어야 하는 전말 과목 수는 3개입니다.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
신청하려는 과목 번호를 입력해주세요: 130
Multimedia 과목이 수강되었습니다. 앞으로 밑수로 들어야 하는 전말 과목 수는 3개입니다.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 3
Mon: DataStructure - Woncheon#340 (Mon#C1#Thu#C1)
Tue:
Wed: Multimedia - Woncheon#540 (Wed#B1#Fri#B1)
Thu: DataStructure - Woncheon#340 (Mon#C1#Thu#C1)
Fri: Multimedia - Woncheon#540 (Wed#B1#Fri#B1)
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 5
작업을 종료합니다

```

```

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

```

```

입력: 2
사용자 이름을 입력해주세요: test2
비밀번호를 입력해주세요: 1234
수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 1
신청하려는 과목 번호를 입력해주세요: 100
LogicCircuit 과목이 수강되었습니다. 앞으로 밑수로 들어야 하는 전말 과목 수는 3개입니다.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 3
Mon:
Tue:
Wed: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)
Thu:
Fri: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)

```

그리고 위의 사진과 같이 서로 다른 수강자들이 과목들을 신청한뒤 작업을 종료 후 다시 로그인을 하면

```

사용자 이름을 입력해주세요: test
비밀번호를 입력해주세요: 1234
수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 3
Mon: DataStructure - Woncheon#340 (Mon#C1#Thu#C1)
Tue:
Wed: Multimedia - Woncheon#540 (Wed#B1#Fri#B1)
Thu: DataStructure - Woncheon#340 (Mon#C1#Thu#C1)
Fri: Multimedia - Woncheon#540 (Wed#B1#Fri#B1)
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 5
작업을 종료합니다

```

```

입력: 2
사용자 이름을 입력해주세요: test2
비밀번호를 입력해주세요: 1234
수강자로 로그인하셨습니다. 수행하고자 하는 작업을 입력해주세요.
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기: 3
Mon:
Tue:
Wed: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)
Thu:
Fri: LogicCircuit - Woncheon#535 (Wed#B1#Fri#B1)
0- 과목 조회, 1- 과목 신청, 2- 과목 취소, 3- 요일별 수업 스케줄 확인, 4-로그인 시간 갱신, 5- 작업 종료. 입력하기:

```

신청한 과목들이 유지되는 것을 확인 할 수 있다.

```

과목 데이터베이스 파일이 없습니다. 과목을 빈 목록으로 초기화합니다.
1

```

```

프로그램 종료: 0
관리자 로그인: 1
수강자 로그인: 2
수강자 등록: 3

```

```

0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기: 3
[]
0- 과목 추가, 1- 과목 삭제, 2- 과목 정보 변경, 3- 개설 과목 조회, 4- 작업 종료. 입력하기:

```

만약 파이썬 파일이 있는 폴더에 basics.txt 파일이 없을 경우 '과목 데이터베이스 파일이 없습니다. 과목을 빈 목록으로 초기화합니다.'라고 띄우고 개설 과목을 조회하면 빈 배열이 나타난다.

6. 결론

프로젝트 요약

1) 기능성

함수를 활용하여 로그인 및 사용자 인증 기능을 구현하였다. 관리자 기능은 CourseAdminSystem 클래스를 통해, 수강 신청자 기능은 CourseRegistrationSystem 클래스를 통해 구현하였다.

예외 처리 또한 구현하였다. basics.txt 파일이 존재하지 않아 과목 데이터베이스를 읽어올 수 없을 경우, '과목 데이터베이스 파일이 없습니다. 과목을 빈 목록으로 초기화합니다.'라고 띄우고 개설 과목을 조회하면 빈 배열이 나타난다. 그 외 사용자 입력이 잘못되었을 경우 다시 입력받도록 구현하였다.

관리자 로그인 계정은 아이디는 root, 비밀번호는 0000으로 고정되어있다.

2) 코드 품질

코드의 재사용성을 늘리기 위해 코드를 모듈화하였다. 해당 프로젝트의 코드는 총 6개의 파이썬 파일로 구성되어 있으며 각각 관리자 시스템(course_admin_system), 수강 과목 목록(course_list), 수강 신청 시스템(course_registration_system), 강의 시간 포맷 검사기(lecture_time_validator), 터미널과 관련된 함수가 포함된 shell_func 파일, 그리고 프로그램의 시작점인 main 파일로 구성되어 있다. 각 파일에는 해당 파일의 역할에 맞는 함수 또는 클래스가 정의되어 있다.

코드 작성자 외에도 코드를 읽는 다른 사람이 이해할 수 있도록 주석을 작성하였다. 각 클래스의 역할이 무엇인지, 각 함수의 기능이 무엇인지를 작성하였고 코드의 흐름에 대해서도 필요하다면 작성하였다.

3) 창의성 및 추가 기능

추가 기능으로 '관리자가 직접 추가하고자 하는 과목을 입력하는 기능'과 '세션 5분 시간 제한 갱신', '사용자 추가' 기능을 구현하였다.

'관리자가 직접 추가하고자 하는 과목을 입력하는 기능'은 프로젝트 요구 사항에서 주어진 5과목을 자동으로 추가하는 것 외에도 관리자가 과목 정보를 입력하여 과목을 추가하는 기능이다.

'세션 5분 시간 제한 갱신'은 학교 홈페이지에 세션을 갱신하는 기능이 있듯이 수강자가 5분 시간 제한이 지난 후에도 세션을 갱신할 수 있도록 추가한 기능이다.

'사용자 추가' 기능은 회원 가입과 유사한 기능이라 말할 수 있다. 사용자의 이름과 비밀번호를 입력받아 새로운 사용자로 추가하며, 이미 같은 사용자 이름이 존재할 경우 사용자는 추가되지 않는다.

4) 프로젝트 문서화

프로젝트 내용에 대해 해당 보고서에 과제 소개 및 프로젝트 소개, 사용된 기술 및 라이브러리 소개, 프로그램 설계 - 전반적 구조, 클래스, 함수, 데이터 구조를 설명하였다. 그리고 프로그램 실행 예시를 통해 어떻게 프로그램이 동작하는지 이미지와 텍스트로 설명하였다.

7. 참고 문헌

프로젝트 개발에 참고한 자료 및 라이브러리는 다음과 같다.

1. Numpy 공식 문서 - https://numpy.org/doc/stable/user/absolute_beginners.html
2. Python 공식 문서 (클래스) - <https://docs.python.org/ko/3/tutorial/classes.html>
3. 아주대학교 임베디드 시스템 설계 강의 자료