|  |  |  |
| --- | --- | --- |
| **기능 명세서** | 문서번호 | TO.004 |
| 작성자 | 2조 |
|  |  |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **Package com.Enrolement.Main** | **메인** |
| **클래스** | **Main** | **메인 역할** |
| **필드** | Scanner sc | 입력을 받기 위한 스캐너 |
| Enrolement e | 수강신청 객체를 생성하기 위한 변수. |
| **메소드** | main | 프로그램의 메인메서드 |
| **패키지** | **Package com.Enrolement.Util** | **Util기능을 모아놓은 패키지** |
| **클래스** | **ExData** | **외부 데이터로부터 불러올 데이터를 저장할 클래스** |
| **필드** | List<ProfessorDTO**>**professorList | 교수관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<StudentDTO**>**studentList | 학생관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<DepartmentDTO**>**departmentList | 학과관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<GeneralSubjectDTO**>**generalSubjectList | 교양과목관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<MajorSubjectDTO**>**majorSubjectList | 전공과목관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<ClassroomDTO**>**classroomList | 강의실관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<PreviousSemesterDTO**>**previousSemesterList | 이전학기성적.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<InterestSubjectListDTO**>**interestSubjectListList | 관심과목목록.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<ClassroomTimeDTO>classroomTimeList | 강의실강의시간.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<OpinionDTO>opinionList | 의견게시판.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<NoticeDTO>noticeList | 공지사항.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<FinalEnrolementDTO>finalEnrolementList | 최종수강신청.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<SemesterManagementDTO>semesterManagementList | 학기관리.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<StudentLoginDTO> studentLoginList | 학생로그인.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<AdminLoginDTO> adminLoginList | 관리자로그인.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| List<PointUseDTO> pointUseList | 포인트사용목록.dat에서 데이터를 얻어와 객체를 저장할 리스트 |
| **클래스** | **UtilAboutFile** | **업무중 사용되는 IO저장/불러오기 기능 모음** |
| **메소드** | load() | 외부 자료들을 불러오는 메소드 |
| **필드** | BufferedReader reader | 외부 파일을 읽어들이는데 사용 |
| String line | 줄별로 읽어들인 자료를 담음 |
| String[] temp | 읽어들인 자료를 담음 |
|  | String professroPath | 교수관리.dat파일이 저장된 경로를 저장 |
| **(정적)**  **필드** | String studentPath | 학생관리.dat파일이 저장된 경로를 저장 |
| String departmentPath | 학과관리.dat파일이 저장된 경로를 저장 |
| String generalSubjectPath | 교양과목.dat파일이 저장된 경로를 저장 |
| String majorSubjectPath | 전공과목.dat파일이 저장된 경로를 저장 |
| String classroomPath | 강의실관리.dat파일이 저장된 경로를 저장 |
| String previousSemesterPath | 이전학기성적.dat파일이 저장된 경로를 저장 |
| String interestSubjectPath | 관심과목목록.dat파일이 저장된 경로를 저장 |
| String classroomTimePath | 강의실강의시간.dat파일이 저장된 경로를 저장 |
| String opinionPath | 의견게시판.dat파일이 저장된 경로를 저장 |
| String noticePath | 공지사항.dat파일이 저장된 경로를 저장 |
| String finalEnrolementPath | 최종수강신청.dat파일이 저장된 경로를 저장 |
| String smemsterManagementPath | 학기관리.dat파일이 저장된 경로를 저장 |
| String studentLoginPath | 학생로그인.dat파일이 저장된 경로를 저장 |
| String adminLoginPath | 관리자로그인.dat파일이 저장된 경로를 저장 |
| pointUsePath | 포인트사용목록.dat파일이 저장된 경로를 저장. |
| **메소드** | professorSave() | 교수정보를 파일에 저장하는 메소드 |
| studentSave() | 학생정보를 파일에 저장하는 메소드 |
| deptSave() | 학과정보를 파일에 저장하는 메소드 |
| generalSubjectSave() | 교양과목 정보를 파일에 저장하는 메소드 |
| majorSubjectSave() | 전공과목 정보를 파일에 저장하는 메소드 |
| classroomSave() | 강의실 정보를 파일에 저장하는 메소드 |
| previousSemesterSave() | 이전학기의 성적들의 정보를 파일에 저장하는 메소드 |
| interestSubjectListSave() | 관심과목들의 정보를 파일에 저장하는 메소드 |
| classroomTimeSave() | 강의실과 강의시간의 정보를 파일에 저장하는 메소드 |
| opinionSave() | 의견함에 있는 정보를 파일에 저장하는 메소드 |
| noticeSave() | 공지사항에 공지한 정보를 파일에 저장하는 메소드 |
| finalEnrolementSave() | 학생의 최종 수강신청 리스트 정보를 파일에 저장하는 메소드 |
| semesterManagementSave() | 해당 학기의 정보를 파일에 저장하는 메소드 |
| studnetLoginSave() | 학생들의 아이디와 비밀번호 정보를 파일에 저장하는 메소드 |
| adminLoginSave() | 관리자 아이디와 비밀전호를 저장 |
| pointUseSave() | 포인트 사용목록을 저장. |
| **클래스** | **UtilAboutCodeFind** | **학생,교수,학과 등 코드를 통해 객체를 얻기위한 유틸클래스** |
| **(정적)**  **필드** | StudentDTO student | 학생객체를 저장하기 위한 변수 |
| ProfessorDTO professor | 교수객체를 저장하기 위한 변수 |
| ClassroomDTO classroom | 강의실객체를 저장하기 위한 변수 |
| DepartmentDTO department | 학과객체를 저장하기 위한 변수 |
| GeneralSubjectDTO generalSubject | 교양과목객체를 저장하기 위한 변수 |
| MajorSubjectDTO majorSubject | 전공과목객체를 저장하기 위한 변수 |
| InterestSubjectListDTO interestSubject | 관심과목객체를 저장하기 위한 변수 |
| **메소드** | findStudent(String studentId) | 학번을 통해 학생객체를 반환하기위한 메소드 |
| findProfessor(String professorCode) | 교수코드를 통해 교수 객체를 반환하기 위한 메소드 |
| findDepartment(String deptCode) | 학과코드를 통해 학과 객체를 반환하기 위한 메소드 |
| findClassroom(String classnum) | 강의실 호수를 통해 강의실 객체를 반환하기 위한 메소드 |
| findGeneralSubject(String subjectCode) | 교양과목코드를 통해 교양과목 객체를 반환하기 위한 메소드 |
| findMajorSubject(String subjectCode) | 전공과목 코드를 통해 전공과목 객체를 반환하기 위한 메소드 |
| findInterestSubject(String subjectCode) | 과목코드를 넣으면 관심과목 객체를 반환하기 위한 메소드 |
| change(MajorSubjectDTO m) | 전공과목객체를 넣으면 관심과목 객체로 변환하기 위한 메소드 |
| Change(GeneralSubjectDTO g) | 교양과목 객체를 넣으면 관심과목 객체로 변환하기 위한 메소드 |
| changeF(GeneralSubjectDTO g) | 교양과목 객체를 넣으면 최종수강신청객체로 변환하기 위한 메소드 |
| changeF(MajorSubjectDTO g) | 전공과목 객체를 넣으면 최종수강신청객체로 변환하기 위한 메소드 |
| changeF(InterestSubjectListDTO i) | 관심과목객체를 넣으면 최종수강신청 객체로 변환하기 위한 메소드 |
| **클래스** | **UtilAboutCodeFind** | **학생,교수,학과 등 코드를 통해 객체를 얻기위한 유틸클래스** |
| **메소드** | interestSubjectApplyCount(String subjectCode) | 관심과목의 신청개수를 리턴해주는 메소드 |
| finalSubjectApplyCount(String subjectCode) | 최종신청과목의 코드를 넘겨주어 신청된 개수를 리턴해주는 메소드 |
| **패키지** | **Package com.Enrolement.Enrolement** | 수강신청 시스템을 구현하는 패키지 |
| **클래스** | **Enrolement** | **수강신청 시스템을 구현** |
| **메소드** | 생성자 | UtilAboutFile클래스에 있는 load()메소드를 실행해 파일에 있는 데이터를 메모리상에 올린다. |
| **필드** | Scanner sc | 스캐너 입출력 담당 |
| **필드**  **(정적)** | Boolean enrolementServer1 | 예비수강신청의 서버 개폐여부 |
| Boolean enrolementServer2 | 수강신청의 서버 개폐여부 |
| Boolean programOnOff | 프로그램의 종료 플래그 |
| StudentDTO loginStudent | 현재 로그인한 학생의 객체를 담는 변수 |
| Void ruclose() | 프로그램 종료시키는 메소드 |
| **메소드** | void programStart() | 본격적인 프로그램을 시작할 메소드 |
| void startScreen() | 프로그램 시작 전 화면을 출력할 메소드 |
| Int login() | 로그인 ui와 내부로직을 담은 메소드 |
| pause() | 잠시 대기할 상황을 표현할 메소드 |
| **클래스** | **AdminSide** | **관리자 영역을 담당하는 클래스** |
| **메소드** | 생성자 | 관리자영역 나타냄 |
| Void endFinalEnrolement() | 최종적으로 각각의 학생들이 수강신청을 종료한다. |
| Void mockFinalEnrolement() | 모의 수강신청을 구현한 메소드 |
| Void studentOpinionCheck() | 학생의 의견함을 출력하는 메소드 |
| Void eroleNoticement() | 공지사항을 입력하고 저장하느 메소드 |
| Void enrolementStart() | 등록 시작 UI메소드 |
| Void adminLoginUI | 관리자 로그인 UI를 나타내는 메소드 |
| Pause\*() | 잠시 중지하려는 메소드 |
| **클래스** | **FinalEnrolement extends ShowInterestSubjectList** | **최종적인 수강신청 단계를 구현한 클래스** |
| **필드** | Scanner sc | 스캐너 입출력 변수 |
| List<PointUseDTO> pointuseList | 포인트사용내역 객체를 저장할 메소드.. |
| **메소드** | Void doEnrolement() | 실제 과목을 등록하려는 메소드 |
| Void finalEnrolementCancel() | 최종수강신청을 취소하려는 메소드 |
| Void doEnrolementFromAllSubject(String num) | 전체과목에서 한꺼번에 수강신청을 진행하는 메소드 |
| Void doEnrolementFromInterest() | 관심과목리스트에서 수간신청을 진행하는 메소드 |
| doEnUI() | 수강신청 UI |
| Pause() | 잠시 중지하기 위한 메소드 |
| isMyPointOK(int payPoint) | 내가 지불하려는 포인트가 부족하지 않은지 판별하는 메소드 |
| **클래스** | **FirstEnrolement** | **예비수강신청을 구현하는 클래스** |
| **메소드** | firstEnrolement() | 예비수강신청의 뼈대를 구현하는 메소드 |
| prepareEnrolementCancel() | 예비수강신청을 취소하는 것을 구현하는 메소드 |
| generalSubjectApply() | 교양과목을 신청하는 메소드 |
| showMajorSubject() | 전공과목 목록을 리스트로 표현해주는 메소드 |
| majorUI() | 전공 신청 UI |
| majorFirstEnrolementSaveOneTime() | 한번에 전공과목을 신청하는 메소드 |
| majorFirstEnrolementSaveOneAndOne() | 한번씩 입력하여 하나씩 전공을 신청하는 메소드 |
| Search() | 과목명입력하고 반환하는 메소드 |
| firstEnrolementUI() | 예비수강신청 UI |
| Pause() | 잠시 중지하기 위한 메소드 |
| **클래스** | **ShowInterestSubjectList** | **관심과목 리스트를 가진 클래스** |
| **메소드** | String showInterestSubject() | 스트링빌더로 로그인한 학생의 관심과목 리스트를 반환. |

|  |  |  |
| --- | --- | --- |
| **패키지** | **com.enrolement.DTO** | **수강신청 시스템 관련 데이터 모음** |
| **클래스** | **ProfessorDTO** | **교수 데이터 저장 클래스** |
| **필드** | String professorCode | 교수코드를 담는 변수 |
| String deptCode | 학과코드를 담는 변수 |
| String professorName | 교수의 이름정보를 담는 변수 |
| Int age | 교수의 나이정보를 담는 변수 |
| String phoneNumber | 교수의 전화번호 정보를 담는 변수 |
| String studyroom | 교수의 연구실 정보를 담는 변수 |
| **클래스** | **StudentDTO** | **학생 데이터 저장 클래스** |
| **필드** | String studentID | 학생의 학번정보를 담는 변수 |
| String studentName | 학생의 이름정보를 담는 변수 |
| String deptCode | 학생이 속한 학과의 학과코드를 담는 변수 |
| String grade | 학생의 학년정보를 담는 변수 |
| Int age | 학생의 나이정보를 담는 변수 |
| String phoneNumber | 학생의 전화번호를 담는 변수 |
| String address | 학생의 주소정보를 담는 변수 |
| Int applyCredit | 학생이 신청한 과목의 총 학점을 담는 변수 |
| Int holdingPoint | 학생이 현재 보유하고 있는 포인트 정보를 담는 변수 |
| **클래스** | **DepartmentDTO** | **학과 데이터 저장 클래스** |
| **필드** | String deptCode | 학과코드를 담는 변수 |
| String deptName | 학과의 이름데이터를 담는 변수 |
| String location | 학과의 학과사무실 위치정보를 담는 변수 |
| **클래스** | **GeneralSubjectDTO** | **교양과목 데이터 저장 클래스** |
| **필드** | String subjectCode | 교양과목의 과목코드를 담는 변수 |
| String deptCode | 교양과목의 소속인 교양학부를 담는 변수 |
| String professorCode | 교양과목의 담당 교수의 교수코드를 담는 변수 |
| String classNum | 교양과목이 행해지는 강의실의 호수를 담는 변수 |
| String subjectName | 교양과목의 과목명을 담는 변수 |
| Int grades | 해당 교양과목의 부여 학점을 담는 변수 |
| Int checkNum | 수강가능한 인원수를 담는 변수 |
| String schejuleTime | 교양과목의 강의시간 데이터를 담는 변수 |
| Int interestNum | 교양과목이 학생들에게 선택받은 횟수를 나타내는 변수 |
| **클래스** | **MajorSubjectDTO** | **전공과목 데이터 저장 클래스** |
| **필드** | String subjectCode | 전공과목의 과목코드를 담는 변수 |
| String deptCode | 전공과목이 소속된 학과의 학과코드를 담는 변수 |
| String professorCode | 전공과목을 가르치는 담당 교수의 교수코드를 담는 변수 |
| String subjectName | 과목명을 담는 코드 |
| Int grades | 과목에게 부여된 학점을 담는 변수 |
| String grade | 해당 과목을 든는 학생집단의 학년을 담는 코드 |
| Int checkNum | 수강가능한 인원을 담는 변수 |
| String schejuleTime | 강의시간에 대한 데이터를 담는 변수 |
| String classNum | 강의실 호수에 대한 데이터를 담는 변수 |
| **클래스** | **ClassroomDTO** | **강의실 데이터 저장 클래스** |
| **필드** | String classNum | 강의실의 호수 데이터를 담는 변수 |
| String deptCode | 강의실의 소속 학과데이터를 담는 학과코드 변수 |
| Int capacity | 강의실의 수용인원 데이터를 담는 변수 |
| **클래스** | **PreviousSemesterDTO** | **전학기 성적 데이터 모음 클래스** |
| **필드** | String studentID | 학생의 학번데이터를 담는 변수 |
| Double lastTotalScore | 학생의 이전 데이터를 담는 변수 |
| **클래스** | **InterestSubjectListDTO** | **관심과목 데이터 모음 클래스** |
| **필드** | String studentID | 학생의 학번데이터를 저장할 변수 |
| String subjectCode | 학생이 예비수강한 과목 데이터를 저장할 변수. |
| **클래스** | **ClassroomTimeDTO** | **관람 등급 데이터 모음 클래스** |
| **필드** | String classNum | 강의실의 호수를 담는 변수 |
| String schedule | 강의실이 이용되는 시간정보(강의시간)를 담는 변수 |
| **클래스** | **OpinionDTO** | **의견게시판 데이터 모음 클래스** |
| **필드** | String studentID | 학생들을 구분할 학번을 담는 변수 |
| String opinion | 학생들이 제출한 의견을 보관하는 변수 |
| **클래스** | **NoticeDTO** | **공지사항 데이터 모음 클래스** |
| **필드** | String notice | 관리자가 입력한 공지사항을 저장하는 변수. |
| **클래스** | **FinalEnrolementDTO** | **최종 수강신청 데이터 모음 클래스** |
| **필드** | String studentID | 학생을 구분할 학번데이터를 담는 변수 |
| String subjectCode | 과목을 나타내는 과목코드를 저장하는 변수 |
| **클래스** | **SemesterManagementDTO** | **학기 데이터 모음 클래스** |
| **필드** | String semester | 현재 수강신청 중인 학기를 저장하는 변수(1/2) |
| **클래스** | **StudentLoginDTO** | **학생들의 아이디와 비밀번호 모음 클래스** |
| **필드** | String id | 아이디 = 학번을 저장하는 변수 |
| String pwd | 비밀번호 = 핸드폰 뒷자리 를 저장하는 변수 |
| **클래스** | **AdminLoginDTO** | **관리자의 아이디와 비밀번호 소유 클래스** |
| **필드** | String id | 관리자 아이디를 저장하는 변수 |
| String pwd | 관리자 비밀번호를 저장하는 변수 |
| **클래스** | **PointUseDTO** | **학생들이 포인트를 사용한 정보를 소유하는 클래스** |
|  | String studentID | 학생들의 학번을 저장하는변수 |
| String subjectCode | 과목의 과목코드를 저장하는 변수 |
| Int usePoint | 사용한 포인트를 저장하는 변수 |

|  |  |  |
| --- | --- | --- |
| **패키지** | **package com.Enrolement. MockEnrolement** | **모의 수강신청 패키지** |
| **클래스** | **MockEnrolement** | **학생들의 모의 수강신청을 구현하는 클래스** |
| **필드** | Random ran | 랜덤객체 생성 |
| **메소드** | Enrolement(studentDTO student) | 학생객체를 매개변수로 넣으면 임의로 수강신청을 진행하는 메소드 |

|  |  |  |
| --- | --- | --- |
| **패키지** | **package com.Enrolment.Administor** | **관리자페이지 부분을 모아놓은 패키지** |
| **클래스** | **Student 클래스** | **학생에 관한 CURD동작** |
| **필드** | static Scanner scan | 키보드에서 입력된 값을 받아온다. |
| static booleanmainloop | 기능을 반복을 묻는 의사 |
| static UIStudent ui | Ui를 사용하기 위한 객체 |
| **메소드** | (Student)Main() | 학생클래스에서 동작하는 모든 메소드를 가진다.(=Main) |
| **필드** | Int sel | 스캐너에서 받아온 번호 |
| **메소드** | addStudent() | 한명의 학생을 등록하기 : 학생.dat파일에 학생 한줄 추가 |
| **필드** | String studentName | 등록하기위해 입력받은 학생이름값 저장 |
| String patName | 학생이름 유효성 검사를 위한 정규식 저장 |
| String studentID | 등록을 위해 입력받은 학생 학번값 저장 |
| String patID | 학생학번 유효성 검사를 위한 정규식 저장 |
| Iterator<StudentDTO> stuIter | 학생 한명씩의 정보를 가져올 리스트 탐색을 위한 iterator |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| String grade | 등록을 위해 입력받은 학생 학년값 저장 |
| String patGrade | 학년값 유효성 검사를 위한 정규식 저장 |
| String phoneNumber | 등록을 위해 입력받은 학생 전번값 저장 |
| String patTel | 전번값 유효성 검사를 위한 정규식 저장 |
| Iterator<StudentDTO> stuIterTel | 학생들 한명씩의 객체를 탐색 |
| String age | 등록을 위한 학생 나이값 저장 |
| Int ageInt | 유효성검사시 학생나이 비교연산 사용하기위한 변수 |
| String address | 등록을 위해 입력받은 주소값 저장 |
| String patAddress | 주소값 유효성 검사를 위한 정규식 저장 |
| Iterator<DepartmentDTO> deptIter | 입력받은 학과명으로 코드를 저장하기위해 학과객체 탐색 |
| DepartmentDTO deptDTO | 학과 객체 하나씩을 저장 >>조건이용 |
| StudentDTO stuDTO | 새로등록할 학생객체를 저장하기위함 |
| **메소드** | upDateStudent() | 학생 정보 하나가 선택되어 수정하기 : 학생.dat파일 내용 한줄이 삭제->추가 |
| **필드** | String Name | 수정하기위해 입력받은 학생이름값 저장 |
| Iterator<StudentDTO> nameiter | 학생 각 객체의 이름을 찾기위한 객체탐색 itorator |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| String num | 수정하기 위해 입력받은 학생학번값을 저장 |
| Iterator<StudentDTO> IDiter | 각각의 학생객체 학번을 탐색하기위한 iterator |
| String deptName | 수정하기 위해 입력받은 학과명을 저장 |
| String deptCode | 학과명을받아 학과객체의 학과명과 일치하는 코드저장 |
| Iterator<DepartmentDTO> deptIter | 입력받은 학과명으로 코드를 저장하기위해 학과객체 탐색 |
| DepartmentDTO deptDTO | 학과 객체 하나씩을 저장 >>조건이용 |
| **메소드** | deleteStudent() | 존재하는 학생의 정보 삭제하기 : 학생.dat파일의 내용 한줄이 삭제 |
| **필드** | String name | 키보드에서 받아온 삭제할 학생의 이름 |
| Iterator<StudentDTO> nameiter | 학생 각 객체의 이름을 찾기위한 객체탐색 itorator |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| String id | 키보드에서 받아온 삭제할 학생의 학번 |
| **메소드** | listStudent() | (학과별) 학생정보 목록띄우기 : 학생.dat파일의 내용을 전체 , 혹은 학과별 필터로 읽어오기 |
| **필드** | String deptName | 키보드에서 받아온 목록 띄울 학과명 |
| String deptCode | 입력학과명과 객체의 학과명이 같을 때 그학과 코드저장 |
| Iterator<DepartmentDTO> deptIter | 입력받은 학과명으로 코드를 저장하기위해 학과객체 탐색 |
| DepartmentDTO deptDTO | 학과 객체 하나씩을 저장 >>조건이용 |
| Iterator<StudentDTO> stuiter | 학생객체의 코드값과 입력코드값이 같은가 비교하기위해 |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| **메소드** | searchStudent() | 특정학생의 정보를 찾아 띄우기 : 학생.dat파일에서 학번으로 학생을 찾아 정보 읽어오기 |
| **필드** | String name | 키보드에서 받아온 특정학생 이름 |
| Iterator<StudentDTO> nameiter | 학생 각 객체의 이름을 찾기위한 객체탐색 itorator |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| String num | 수정하기 위해 입력받은 학생학번값을 저장 |
| Iterator<StudentDTO> IDiter | 각각의 학생객체 학번을 탐색하기위한 iterator |
| StudentDTO stuDTO | Iterator로 읽어온 객체 하나씩을 저장 |
| **클래스** | **UIStudent 클래스** | **학생 관리에서의 모든 UI** |
| **메소드** | studMenu() | 학생관리자 시스템 첫화면 메뉴출력 (추가/수정/삭제/목록/종료) |
| **메소드** | endComment() | 작업이 끝난후 띄워지는 멘트 |
| **필드** | publici static final int ADD | 추가가 끝났을때의 멘트 |
| publici static final int UPDATE | 수정이 끝났을때의 멘트 |
| publici static final int DELETE | 삭제가 끝났을때의 멘트 |
| publici static final int LIST | 목록띄우기 끝났을때의 멘트 |
| publici static final int SEARCH | 검색이 끝났을때의 멘트 |
| **메소드** | eachElement() | 학생 목록의 요소 출력 >>ex) [학생 이름] : |
| **필드** | publici static final int NAME | 입력시 이름라벨 |
| publici static final int ID | 입력시 학번라벨 |
| publici static final int GRADE | 입력시 학년라벨 |
| publici static final int TEL | 입력시 전화번호라벨 |
| publici static final int AGE | 입력시 나이라벨 |
| publici static final int ADDRESS | 입력시 주소라벨 |
| publici static final int CODE | 입력시 학과라벨 |
| **메소드** | updateUI() | 학생수정 타이틀 |
| **클래스** | **Dept 클래스** | **학과에 관한 CURD 동작** |
| **필드** | static Scanner scan | 키보드에서 입력된 값을 받아온다. |
| static booleanmainloop | 기능을 반복을 묻는 의사 |
| static UIDept ui | Ui를 사용하기 위한 객체 |
| **메소드** | (Department)Main() | 학생클래스에서 동작하는 모든 메소드를 가진다.(=Main) |
| **필드** | Int sel | 스캐너에서 받아온 번호 (실행할 기능 선택) |
| **메소드** | addDept() | 학과를 하나 등록하기 : 학과.dat파일에 학과 한줄 추가 |
| **필드** | String deptCode | 등록하기위해 키보드에서 받아온 학과 코드값저장 |
| String patCode | 코드값 유효성검사를 위한 정규식 저장 |
| Iterator<DepartmentDTO> deptIter | 코드값 중복 유효성검사를 위한 학과객체 탐색iterator |
| DepartmentDTO deptDTO | iterator에서 탐색한 각각의 학과 객체 저장 |
| String deptName | 등록하기위해 키보드에서 받아온 학과명을 저장 |
| String patName | 학과명 유효성검사를 위한 정규식 저장 |
| String location | 등록하기 위해 키보드에서 받아온 학과사무실 위치 저장 |
| String patLoc | 사무실위치 유효성 검사를 위한 정규식 저장 |
| DepartmentDTO deptDTO | 새로운 학과객체를 저장하기위한 객체 |
| **메소드** | upDateDept() | 학과의 정보 하나가 선택되어 수정하기 : 학과.dat파일 내용에 한줄이 삭제->추가 |
| **필드** | Iterator<DepartmentDTO> nameiter | 선택을위해 학과목록을 띄우기위한 iterator |
| DepartmentDTO deptDTO | 학과객체 하나하나를 탐색하기위한 객체저장공간 |
| String deptCode | 수정할 학과코드값을 받아 저장 |
| **메소드** | deleteDept() | 존재하는 학과의 정보 삭제하기 : 학과.dat파일의 내용 한줄이 삭제 |
| **필드** | Iterator<DepartmentDTO> nameiter | 선택을위해 학과목록을 띄우기위한 iterator |
| DepartmentDTO deptDTO | 학과객체 하나하나를 탐색하기위한 객체저장공간 |
| String id | 삭제할 학과코드값을 받아 저장 |
| **메소드** | listDept() | 전체 학과의 정보목록을 띄우기 : 학과.dat파일의 내용 전체를 읽어오기 |
| **필드** | Iterator<DepartmentDTO> nameiter | 선택을위해 학과목록을 띄우기위한 iterator |
| DepartmentDTO deptDTO | 학과객체 하나하나를 탐색하기위한 객체저장공간 |
| **메소드** | searchDepartment() | 특정 학과의 정보를 찾아 띄우기 : 학과.dat파일에서 코드로 학과를 찾아 정보 읽어오기 |
| **필드** | String deptname | 검색할 학과의 이름값 받아와 저장 |
| Iterator<DepartmentDTO> nameiter | 선택을위해 학과목록을 띄우기위한 iterator |
| DepartmentDTO deptDTO | 학과객체 하나하나를 탐색하기위한 객체저장공간 |
| **클래스** | **UIdept 클래스** | **학과관리에서의 모든 UI** |
| **메소드** | deptMenu() | 학과 관리자 시스템 첫화면 메뉴 출력 (추가/수정/삭제/목록/종료) |
| **메소드** | endComment() | 작업이 끝난후 띄워지는 멘트 |
| **필드** | publici static final int ADD | 추가가 끝났을때의 멘트 |
| publici static final int UPDATE | 수정이 끝났을때의 멘트 |
| publici static final int DELETE | 삭제가 끝났을때의 멘트 |
| publici static final int LIST | 목록띄우기 끝났을때의 멘트 |
| publici static final int SEARCH | 검색이 끝났을때의 멘트 |
| **메소드** | eachElement() | 학과 목록의 요소 출력 >>ex) [학과 명] : |
| **필드** | publici static final int NAME | 입력시 이름라벨 |
| publici static final int CODE | 입력시 코드라벨 |
| publici static final int POSITION | 입력시 학과사무실 위치라벨 |
| **클래스** | **Professor** | **교수 정보의 메소드를 모아놓는 클래스** |
| **필드** | Scannerscan=new Scanner(System.in); | 사용자로부터 입력받아 값을 저장하는 변수 |
| UIprofessor ui | 교수님 화면을 담을 변수 |
| String sel | 관리자의 입력을 담을 변수 |
| boolean loop | 메인화면의 실행 루프 |
| ArrayList<Professor>professor | 교수 정보를 담을 컬렉션 |
| String professorCode | 교수 코드를 담을 변수 |
| String deptCode | 과목 코드를 담을 변수 |
| String professorName | 교수 이름을 담을 변수 |
| String professorName | 교수 이름을 담을 변수 |
| int age | 교수 나이를 담을 변수 |
| String phoneNumber | 핸드폰 번호 담을 변수 |
| String studyroom | 교수 연구실 담을 변수 |
| professorMain() | 교수 생성자 |
| regexCode(Pratten.matches) | 유효성 검사 |
| Int sel | 메뉴 버튼을 담을 변수 |
| Int Add | 메뉴 추가를 담을 변수 |
| Int Update | 메뉴 수정을 담을 변수 |
| Int Delete | 메뉴 삭제를 담을 변수 |
| Int List | 메뉴 목록을 담을 변수 |
| **메소드** | addProfessor () | 교수 정보 추가하는 메소드 |
| updateProfessor () | 교수 정보 수정하는 메소드 |
| deleteProfessor () | 교수 정보 제거하는 메소드 |
| listProessor () | 교수 정보 목록하는 메소드 |
| searchProfessor () | 교수 정보 검색하는 메소드 |
| Professormenu() | 교수 메뉴 호출 메소드 |
| startSearch() | 교수 검색 시작 메소드 |
| EndSearch() | 교수 검색 종료 메소드 |
| Start() | 교수 시작 메소드 |
| End() | 교수 종료 메소드 |
| **클래스** | **UIProfessor** | **교수 정보를 보여주는위해 모아놓은 클래스** |
| **필드** | Scanner scan | 사용자로부터 키보드 입력을 받을 스캐너 |
| Public static final int ADD | 교수 추가 정적 변수 |
| public static final int UPDATE | 교수 수정 정적 변수 |
| public static final int LIST | 교수 목록 정적 변수 |
| public static final int SEARCH | 교수 검색 정적 변수 |
| **메소드** | professorMenu() | 교수관리 메뉴 출력 |
| endComment() | 작업이 끝난후 띄워지는 메시지 |
| startSearch() | 교수 관리 메뉴 시작 메소드 |
| endSearch() | 교수 관리 메뉴 종료 메소드 |
| **클래스** | professorMenu() | 교수관리 메뉴 출력 |
| **메소드** | Check() |  |
| checkProfessorName() | 교수코드 유효성 검사 |
| checkProfessorNumber() | 교수 핸드폰 번호 유효성 검사 |
| checkProfessorName() | 교수 핸드폰 번호 유효성 검사 |
| checkProfessorCode() | 교수 코드 유효성 검사 |
| **클래스** | Major | 전공 과목 |
| **필드** | Scanner scan | 사용자로부터 키보드 입력을 받을 스캐너 |
| UI ui | UI를 사용하기 위한 객체 |
| String departCode | 학과코드를 담을 변수 |
| String subjectCode | 과목코드를 담을 변수 |
| String professorCode | 교수코드를 담을 변수 |
| String subjectName | 과목명을 담을 변수 |
| String grades | 학점을 담을 변수 |
| String grade | 학년을 담을 변수 |
| String scheduleTime | 강의시간을 담을 변수 |
| String classNum | 강의실을 담을 변수 |
| MajorSubjectDTO dto | 새로등록할 전공과목객체를 저장하기위함 |
| Iterator<MajorSubjectDTO> iter | 전공과목 탐색 |
| int searchNum | 검색 선택 번호를 담을 변수 |
| Int deleteSelect | 삭제 다시 확인 선택 번호를 담을 변수 |
| Int searchSelect | 검색 선택 |
| SubjectMenu sub | 과목메인메뉴 객체선언 |
| Int select | 완료 후 메뉴 선택 |
| **메소드** | majorAdd() | 전공과목 등록 메소드 |
| majorUpdate() | 전공과목 수정 메소드 |
| majorDelete() | 전공과목 삭제 메소드 |
| majorList() | 전공과목 목록 출력 메소드 |
| deptSearch() | 학과코드로 검색 메소드 |
| professorSearch() | 교수코드로 검색 메소드 |
| SubjectSearch() | 과목코드로 검색 메소드 |
| **클래스** | General | 교양 과목 |
| **필드** | Scanner scan | 사용자로부터 키보드 입력을 받을 스캐너 |
| UI ui | UI를 사용하기 위한 객체 |
| String departCode | 학과코드를 담을 변수 |
| String subjectCode | 과목코드를 담을 변수 |
| String professorCode | 교수코드를 담을 변수 |
| String subjectName | 과목명을 담을 변수 |
| String grades | 학점을 담을 변수 |
| String scheduleTime | 강의시간을 담을 변수 |
| String classNum | 강의실을 담을 변수 |
| GeneralSubjectDTO subDTO | 새로등록할 전공과목객체를 저장하기위함 |
| Iterator< GeneralSubjectDTO> iter | 교양과목 탐색 |
| int searchNum | 검색 선택 번호를 담을 변수 |
| Int deleteSelect | 삭제 다시 확인 선택 번호를 담을 변수 |
| Int searchSelect | 검색 선택 |
| SubjectMenu sub | 과목메인메뉴 객체선언 |
| Int select | 메뉴 선택 |
| **메소드** | generalAdd() | 교양과목 등록 메소드 |
| generalUpdate() | 교양과목 수정 메소드 |
| generalDelete() | 교양과목 삭제 메소드 |
| generalList() | 교양과목 목록 출력 메소드 |
| deptSearch() | 학과코드로 검색 메소드 |
| professorSearch() | 교수코드로 검색 메소드 |
| SubjectSearch() | 과목코드로 검색 메소드 |
| **클래스** | SubjectUI | 과목 전체 UI |
| **필드** | Public static final int UiAdd | 과목 등록 |
| public static final int UiUpdate | 과목 수정 |
| Public static final int UiDelete | 과목 삭제 |
| Public static final int UiList | 과목 목록 |
| **메소드** | SubjectMenu() | 과목 메뉴 메소드 |
| UiMajorSubjectMain() | 전공과목 메뉴 메소드 |
| UiGeneralSubjectMain() | 교양과목 메뉴 메소드 |
| UiSearchMenu() | 검색 메뉴 메소드 |
| MajorTitle(int n)  >>인덱스 값 : 메뉴 선택 숫자 | 각 전공CRUD의 헤드라인 메소드 |
| GeneralTitle(int n)  >>인덱스 값 : 메뉴 선택 숫자 | 각 교양 CRUD의 헤드라인 메소드 |
| SubjectCompletMenu(int n)  >>인덱스 값 : 메뉴 선택 숫자 | 각 CRUD의 완료 메소드 |
| **클래스** | SubjectMenu | 과목 메인 메뉴 |
| **필드** | Scanner scan | 사용자로부터 키보드 입력을 받을 스캐너 |
| UI ui | UI를 사용하기 위한 객체 |
| Major major | 전공과목 기능을 사용하기 위한 객체 |
| General general | 교양과목 기능을 사용하기 위한 객체 |
| Int select | 메뉴 선택 변수 |
| **메소드** | Menu() | 과목 메뉴 메소드 |
| **클래스** | classroom | 강의실 메뉴의 각 출력값 출력 |
| **필드** | UIClass ui | UI를 사용하기 위한 객체 |
| String classNum | 강의실 호수 |
| String deptCode | 학과 코드 |
| String Capacity | 수용인원 |
| String room | 강의실 정보 선택 변수 |
| Scanner scan | 관리자로부터 키보드 입력을 받을 스캐너 |
| String input | 입력받은 문자열을 담을 변수 |
| Int select | 완료 후 메뉴 선택해주는 변수 |
| ClassroomDTO clrDTO | 새로 등록할 강의실객체를 저장하기 위한 변수 |
| Loop | 루프변수 |
| classroomList | 강의실 데이터의 모든 값을 가지고 있는 리스트 |
| String subjectCode | 강의실 호수를 입력받는 변수 |
| String regex | 강의실 정보들에 대한 유효성검사 조건 변수 |
| Iterator<classroomDTO> iter | 강의실 검색 |
| **메소드** | classroom() | 강의실 클래스에서 동작하는 모든 메소드를 가진다. |
| classroomAdmin() | 강의실 관리 메인메소드 |
| addClass() | 강의실을 추가시키는 메소드 |
| updateClass() | 강의실을 수정하는 메소드 |
| deleteClass() | 필요없는 강의실을 삭제하는 메소드 |
| searchClass() | 강의실의 목록을 보여주게하는 메소드 |
| ClassroomSave | 강의실을 저장시켜주는 메소드 |
| loadData() | 강의실 데이터에 값을 가져야 객체를 만듬 |
| **클래스** | UIclass | 강의실 정보를 보여주기 위해 출력 |
| **필드** | Public static final int add | 강의실 추가 변수 |
| Public static final int update | 강의실 수정 변수 |
| Public static final int delete | 강의실 삭제 변수 |
| Public static final int search | 강의실 호수 검색 변수 |
| Public static final int search2 | 강의실 학과 검색 변수 |
| **메소드** | classTitle() | 강의실 변수의 제목 출력ui |
| classMore() | 강의실 추가 피드백UI |
| classComplete | 강의실 정보 입력 완료 ui |
| classMenu() | 강의실 관리의 첫화면 출력 및 메뉴별 메소드 호출 |
| endComment() | 작업 처리후 마지막 띄워지는 메시지 |
| **클래스** | Check | 유효성 검사 |
| **메소드** | checkClass | 강의실 호수 유효성 검사 |
| checkdeptcode | 학과 코드 유효성 검사 |