- RegistrationSystem: RegistrationSystem __listOfSemester: List -__registrationSystem1: RegistrationSystem __gano: Float + __init__() __log: Logger __totalCreditsTaken: List __points: Float + __init__(RegistrationSystem) _totalCreditsComplated: Integer + instructorLogin(): Void - instructorMenu(Student): Void + __init__() showLecture(Student): Void + addSemester(Semester): Void + showStudents(Student): Void + totalCreditsComplatedCalculator(List): Integer - signOut(): Void + totalCreditsTakenCalculator(List): Integer term: Term + pointsCalculator(): Integer __termYear: TermYear __init__() StudentRegistrationSystem AdvisorRegistrationSystem - registrationSystem: RegistrationSystem __ registrationSystem1: RegistrationSystem __registrationSystem: RegistrationSystem __log: Logger log: Logger _gradeLectureList: List - scanner : Scanner objects1: ObjectCreator __creditsTaken: Integer - RegistrationSystem: RegistrationSystem __init__(RegistrationSysem) __creditsCompleted: Integer -scanner : Scanner + studentLogin(): Void + __init__(RegistrationSystem) + studentMenu(Student): Void __yano: Float + advisorLogin(): Void + showTranscript(Student): Void + advisorMenu(Student): Void + item: attribute - term: Term _singOut(): Void + singOut(): Void + __init__() «interface» SessionID LectureID <u>UniqeID</u> __id: Integer __LectureCode: String + __init__(Integer) <constructor>>LectudeID() + setID(int): void + setID(String): void Implement Implement + getID(): String + getID(): String InstructorID StudentID __departmentCode: Integer _departmentCode: Integer __orderOfEntry: Integer __yearCode: Integer __orderOfPlacement: Integer + __init__(Any...) + digitFixer(Integer): String + __init__(Any...) + setID(int, int): void + __digitFixer(Integer): String + setID(int, int, int): void + getID(): String - __namePool: NamePool + getID(): String + setID(String): void + ___init___() Person _firstName: String lastName: String fullName: String + ___init___() generateAdvisors(): void Inherit Inherit Student Instructor · id: StudentID _id: InstructorID __advisor: Advisor __schedule: Schedule _schedule: Schedule _dateOfEntry: DateTime __transcript: Transcript · _instructorType: InstructorType __dateOfEntry: DateTime - - scanner: Scanner __debt: Debt + ___init___() __registrationApplication: LectureRegistrationApplication + chackScheduleForLecture(Schedule, Lecture): Boolean + showStudents(): Void -__scanner: Scanner + showLectures(): Void + __init__() + sendForApproval(List): Void + canTakeLecture(Lecture, Transcript): Boolean Inherit + hasPreqLectureTaken(Lecture, List): Boolean + takenPoint(Lecture, List): Boolean +availableLessons(): List Advisor +availableLessons(Student): List _listOfStudents: List + checkScheduleForLecture(Schedule, Lecture): Boolean _listOfApplications: List - showChosenLectureSessions(ArrayList<LectureSession>): void - scanner: Scanner parseSelectionCommand(String,ArrayList<LectureSession>): voi + __init__() + makeRegistrationMenu(): void + approveAplication(LectureRegistirationApplication, LectureSession): Void + registrationStatusMenu(): void + rejectApplication(LectureRegistirationApplication, LectureSession): Void + debtMenu(): void + applicationOperations(Integer): void + showSchedule(): void + showApplications(): void + showTranscript(): void

Transcript

_student: Student

InstructorRegistrationSystem

- scanner : Scanner

Debt student: Student amount: Float

LectureRegistrationApplication __sessionsSentForApproval: Map<LectureSession, ApprovalState> __advisor: Advisor _student: Student

RegistrationSystem __simulatin: Simulation __log: Logger + ___init___() + menu(): Void

Lecture

+ removeLectureSession(LectureSession): Void

DataManager

+ searchAdvisor(String, FilterType): ArrayList<Advisor>

+ searchStudent(String, FilterType): ArrayList<Student>

+ searchLecture(String, FilterType): ArrayList<Lecture>

+ findInstructer(String, FilterType): Optional<Instructer>

+ findStudent(String, FilterType): Optional<Student>

+ findLecture(String, FilterType): Optional<Lecture>

DataManagerInstance

+ findAdvisor(String, FilterType): Optional<Advisor>

+ searchInstructor(String, FilterType): ArrayList<Instructor>

__singleInstance: DataManagerInstance

-jsonOperator: JsonOperator

- cacheList: ArrayList<Person>

+<<<constructor>>>_init_(self)

__listOfLectures: list()

__listOfPeople: list()

jsonOperator: JsonOperator

+<<<constructor>>>_init_(self)

+ findLecture(self, key, filterType): Lecture

+ findStudent(self, key, filterType): Student

+ findAdvisor(self, key, filterType): Advisor

+ searchLectures(self, key, filterType): list()

+ searchStudents(self, key, filterType): list()

+ searchAdvisors(self, key, filterType): list()

+ searchLecturesOfTerm(self, key, term): list()

addStudents(self, studentsList): void

searchLecturesUntilTerm(self, key, term, termYear): list()

Simulation

listOfPeople: ArrayList<Person>

· listOfLectures: ArrayList<Lecture>

+ addPrerequisiteLecture(Lecture): Void

+ removePrerequisiteLecture(): Void

Logger

__id: LectureID

__name: String

__credit: Integer

__sessions: List

-__quota: Integer

__term: Term

+ ___init___()

-_loggers: Dictionary (String, Logger)

randomOutputFile: RandomAccessFile

outputFile: outputFile

+ __init__(String)

+ error(String): Void

+ error(String, any...): void

+ info(String, any...): void

+ getLogger(String): Logger

_log(String, String): void

__prerequisite: Lecture

__termYear: TermYear

__lectureType: LectureType

Schedule

- __person: Person
- __listOfLectureSessions: List

_init__()

- __sessionType: SessionType
- +showSchedule(Logger): void

__sessionID: SessionID __lecture: Lecture

LectureSession

- __sessionHours: List
- __instructor: Instructor
- __listOfAssistans: List __listOfStudents: list()
- __init__()

Semester

- __points: Float
- __listOfLecturesTaken: Dictionary

- termYear: TermYear

- + addLecture(Lecture, LetterGrade): Void
- __creditsTakenCalculator(Dictionary): Integer
- __creditsCompletedCalculator(Dictionary): Integer
- __pointsCalculator(Dictionary): Float
- + removeLecture(Lecture,LectureGrade): void

JsonOperator

__lectureJsonDicts: List

- __lectureObjectsList: List
- __advisorJsonDicts: IList
- __advisorObjectList: List
- __studentJsonDicts: List
- -_studentObjectList: List
- -__transcriptJsonDicts: List
- -__transcriptObjectList: List

metaData: Dictionary

+ readMetaData(): Void

- + readNamePool(): Void
- + readLectures(): Void
- generateLectureObjects(): Void
- __generateLectureSessions(LectureSessions, Lecture): List

+ readAdvisors(): Void

- + readStudents(): void
- generateStudentObjects(): void
- + readTranscripts(): void
- + pairObjects(): void
- __pairLectures(self): void
- __pairStudents(): void
- + pairAdvisors(self): void
- + pairTranscripts(self, student): void
- + saveStudent(self, student): void
- + saveAdvisor(self, advisor): void
- + saveLecture(self, lecture): void
- + saveTranscript(self, transcript): void
- __findLecture(String): None
- __findAdvisor(String): None
- __findStudent(String): None
- _strDateTime(String): DateTime

Classname

- + field: type
- + generate(List, Term): List
- + lectureTakenCaculaterFromType(Student, Lecture):

StudentGenerator

+ field: type

- + generate(Integer, Integer): Student
- studentIdGenerator(
- + studentDebtGenerator(): Float

- + startSimulation(List): Void

+ loadFiles(self): void

+ saveFiles(self): void

· listOfStudents: List

- year: Integer

term: Term

+ run(): List

- + skipTerm(List, Term): Void
- + takeAutoAnsverForLRA(List): Void
- + fillSemesterFromLRA(List): Void
- + takeAutoLetterGradeForSemester(List): Void
- + emptyLRA(List): Void
- + item: attribute