

Student
- name: String - logString: String - studentID: StudentID - registrationOrder: int - grades: ArrayList<PassedCourseGrade> - currentCourse: ArrayList<Course> - advisor: Advisor - schedule: Schedule - transcript: Transcript - registrationController: RegistrationController
+ Student(name: String, registrationYear: int, registrationOrder: int, registrationController: RegistrationController) + getPassedCourses(): ArrayList<Course> + isPassed(course: Course): boolean + newCourses(courseSection: CourseSection) + getGrade(course: Course) + selectAndSendToAdvisor() + getLogString(): String + setLogString(logString: String) + getName(): String + getStudentID(): StudentID + getGrade(): ArrayList<PassedCourseGrade> + getTakenCourses(): ArrayList<Course> + getYear(): int + getAdvisor(): Advisor + setAdvisor(advisor: Advisor) + getSchedule(): WeeklySchedule + setSchedule(schedule: WeeklySchedule) + getTranscript(): Transcript

WeeklySchedule
- stuSchedule: CourseSection[][] - student: Student
+ WeeklySchedule(student: Student) + addToSchedule(courseSection: CourseSection) + isCollision(courseSection: CourseSection): boolean

StudentID
- rank: int - year: int - department: String= "1501" {readOnly}
+ StudentID(year: int, rank: int) - registrationYear(): String + getStudentID(): String + toString(): String

CourseSection
- course: Course - full: boolean - sectionHour: int - students: ArrayList<Student> - courseSchedule[][]: boolean - quota: int - numberOfQuotaFail: int - numberOfPrereqFail: int - numberOfCollisionFail: int - numberOfCreditFail: int - failedQuota: String - failedPrereq: String - failedCollision: String - failedCredit: String
+ CourseSection(course: Course) + setCourseSchedule() + addStudent(student: Student) + getQuota(): int + isFull(): boolean + getCourse(): Course + getCourseSectionCode(): String + setFull(full: boolean) + setSectionHour() + getCourseSchedule(): boolean[][] + getNumberOfQuotaFail(): int + getNumberOfPrereqFail(): int + getNumberOfCollisionFail(): int + getNumberOfCreditFail(): int + setNumberOfCollisionFail(numberOfCollisionFail: int) + setNumberOfPrereqFail(numberOfPrereqFail: int) + setNumberOfCreditFail(numberOfCreditFail: int) + getFailedQuota(): String + getFailedPrereq(): String + getFailedCollision(): String + getFailedCredit(): String + setFailedPrereq(failedPrereq: String) + setFailedCollision(failedCollision: String) + setFailedCredit(failedCredit: String)

PreviousCourseGrade
- decimalGrade: int - course: Course
+ PassedCourseCode(course: Course, decimalGrade: int) + convertLetterGrade(): String + getCourse(): Course

Transcript
- student: Student
+ Transcript() + calculateCredits(): int + calculateGPA(): double

Advisor
- name: String
+ approveCourse(student: Student, courseSection: CourseSection) + getName(): String

RegistrationController
- semester: String - year: int - courses: ArrayList<Course> - courseSections: ArrayList<CourseSection> - numberOfStudents: int - numberOfAdvisors: int - advisors: ArrayList<Advisor> - students: ArrayList<Student>
+ RegistrationController() - startRegistration() + getNamesList(): ArrayList<String> - createAdvisors() - createStudents() - assignAdvisorsToStudents() - addPreviousCourses() - selectCourses() - showOutput() - showStatistics() + createNewCourseList(student: Student): ArrayList<CourseSection> - findCourse(courseCode: String): Course + setNumberOfStudents(numberOfStudents: int) + setNumberOfAdvisors(numberOfAdvisors: int)

Course
- courseCode: String - courseName: String - semester: String - quota: int - credits: int - year: int - sectionHours[2]: int - requiredCredits: int - prerequisite: Course - passingCourseProb: double
+ Course(courseCode: String, courseName: String, semester: String, quota: int, credits: int, theoretic: int, lab: int, year: int, requiredCredits: int, prerequisite: Course, passingCourseProb: double) + getPassingCourseProb(): double + getCourseCode: String + getSemester(): String + getQuota(): int + getCredits(): int + getYear(): int + getRequiredCredits(): int + getPrerequisite(): Course + setPrerequisite(prerequisite: Course) + getSectionHours(): int + setSectionHours(theoretic: int, lab: int)