

GenerateStudent

-course ff rate: float -max number of selection for courses: int -student: List<Student> -courses: List<Courses> -advisors: List<Advisors> -firstSemesterCourses: List<String> -secondSemesterCoursesHash: HashMap<String, List<String>> -thirdSemesterCoursesHash: HashMap<String, List<String>> -fourthSemesterCoursesHash: HashMap<String, List<String>> -fifthSemesterCourses: HashMap<String, List<String>> -sixthSemesterCoursesHash: HashMap<String, List<String>> -seventhSemesterCoursesHash: HashMap<String, List<String>> -eighthSemesterCoursesHash: HashMap<String, List<String>> +addCourseNames(): void +generateYear(student: Student): void +semesterSetter(s: Student, semester: String): void +setCoursesList(s: Student): void +assignFailedCourses(currentSemesterFailed: List<FailedCourses>, courseCode: String): void +prerequisiteControlAndLock(courseCode: String, lockedCourses: HashMap<String, List<String>>): +addCompletedCourses(currentSemesterCompleted: List<CompletedCourses>.courseCode: String,

grade: String, finishedSemester: int): void

currentSemester: int): void

+ unlock Locked Courses And Set Available (s: Student, completed Courses: List < Completed Courses>,

lockedCourses: HashMap<String, List<String>>): void

+checkAvailableCourse(s: Student, currentSemesterCompleted: List<CompletedCourses>, currentSemesterFailed: List<FailedCourses>, lockedCourses: HashMap<String, List<String>>,

+simulateFailedCourses(s: Student, currentSemesterCompleted: List<CompletedCourses>,

currentSemesterCourses: HashMap<String, List<String>>):void

+checkCourseGiven(s: Student): void

+remove_unamed_courses(s: Student): void

+checkCourseHasPrerequisite(courseCode: String): void

+checkPrerequisiteCourseIsGiven(s: Student, courseCode: String, semester: int): boolean

+courselsGivenAlready(s: Student, courseCode: String): boolean

+setStudentAdvisor(s: Student): void

+caseTwo(): void +caseThree(): void +caseFour(): void otherCases: void

+simulateSemester(s: Student, semester: String): void

- CalculateAvailables - +set_attributes(courses: Courses[]): void +putAvailableCoursesCaseTwo(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseThree(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseFour(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseFive(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseSix(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseSeven(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseEigth(courses: Courses[],studentCourseTook:List<String +putAvailableCoursesCaseEigth(courses: Courses[],studentCourseTook:List<String +calculatedCoursesResetter(): void +set_available_courses_for_each_student(students, courses, advisors, UE, TE, FTE, NTE, max_number_of_selection_for_courses) +set_students_for_each_advisor(students, advisors) +set_students_for_each_courses(students, courses)