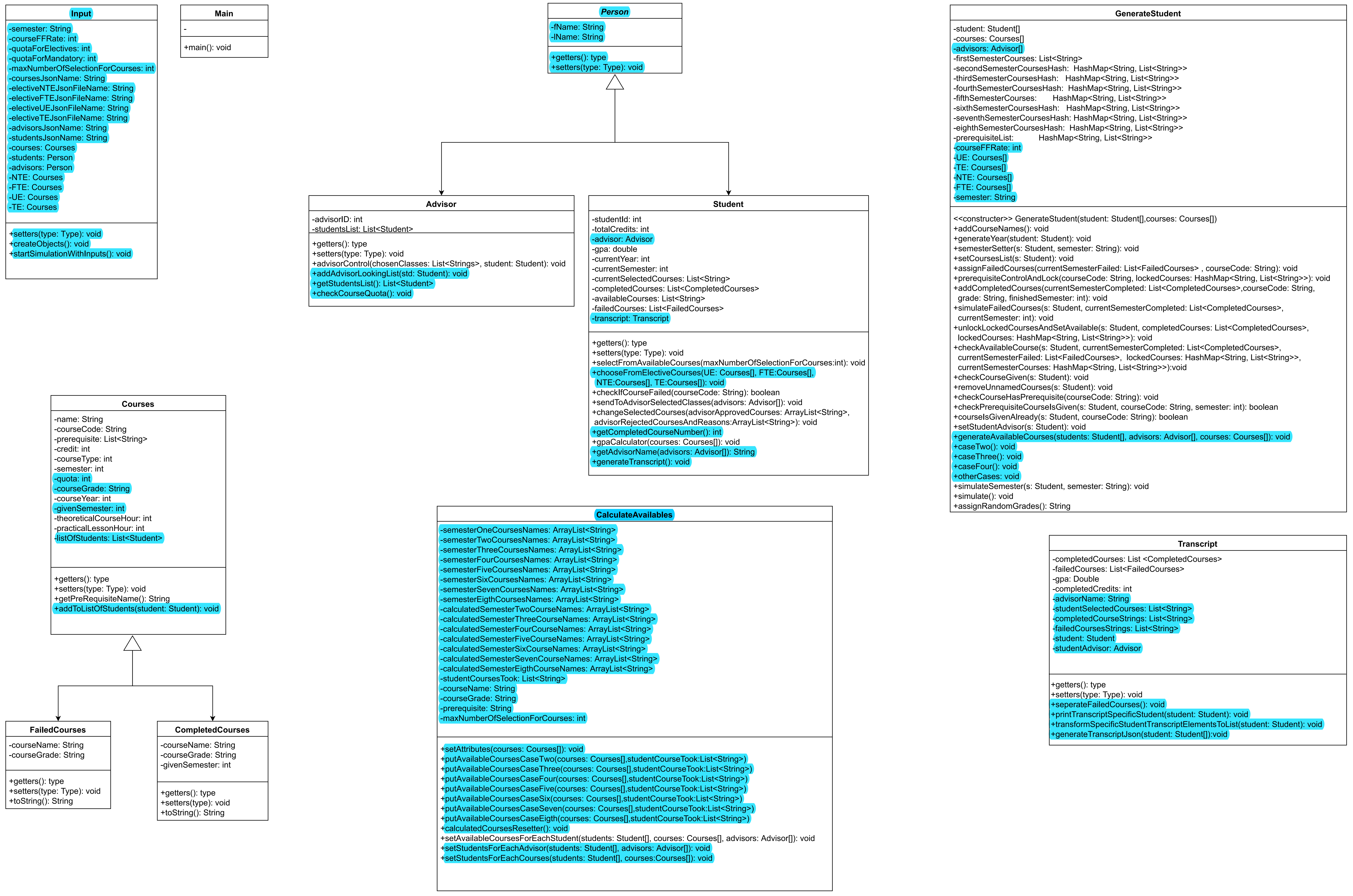
Input
-semester: String
-courseFFRate: int
-quotaForElectives: int
-quotaForMandatory: int
-maxNumberOfSelectionForCourses: int
-coursesJsonName: String
-electiveNTEJsonFileName: String
-electiveFTEJsonFileName: String
-electiveUEJsonFileName: String
-electiveTEJsonFileName: String
-advisorsJsonName: String
-studentsJsonName: String
-courses: Courses
-students: Person
-advisors: Person
-NTE: Courses
-FTE: Courses
-UE: Courses
-TE: Courses
+setters(type: Type): void
+createObjects(): void
+startSimulationWithInputs(): void
Main
-
+main(): void
Person
-fName: String
-lName: String
+getters(): type
+setters(type: Type): void
Advisor
-advisorID: int
-studentsList: List<Student>
+getters(): type
+setters(type: Type): void
+advisorControl(chosenClasses: List<Strings>, student: Student): void
+addAdvisorLookingList(std: Student): void
+getStudentsList(): List<Student>
+checkCourseQuota(): void
Student
-studentId: int
-totalCredits: int
-advisor: Advisor
-gpa: double
-currentYear: int
-currentSemester: int
-currentSelectedCourses: List<String>
-completedCourses: List<CompletedCourses>
-availableCourses: List<String>
-failedCourses: List<FailedCourses>
-transcript: Transcript
+getters(): type
+setters(type: Type): void
+selectFromAvailableCourses(maxNumberOfSelectionForCourses:int): void
+chooseFromElectiveCourses(UE: Courses[], FTE:Courses[], NTE:Courses[], TE:Courses[]): void
+checkIfCourseFailed(courseCode: String): boolean
+sendToAdvisorSelectedClasses(advisors: Advisor[]): void
+changeSelectedCourses(advisorApprovedCourses: ArrayList<String>, advisorRejectedCoursesAndReasons:ArrayList<String>): void
+getCompletedCourseNumber(): int
+gpaCalculator(courses: Courses[]): void
+getAdvisorName(advisors: Advisor[]): String
+generateTranscript(): void
GenerateStudent
-student: Student[]
-courses: Courses[]
-advisors: Advisor[]
-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>>
-prerequisiteList: HashMap<String, List<String>>
-courseFFRate: int
-UE: Courses[]
-TE: Courses[]
-NTE: Courses[]
-FTE: Courses[]
-semester: String
<<constructer>> GenerateStudent(student: Student[],courses: Courses[])
+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>>): void
+addCompletedCourses(currentSemesterCompleted: List<CompletedCourses>,courseCode: String, grade: String, finishedSemester: int): void
+simulateFailedCourses(s: Student, currentSemesterCompleted: List<CompletedCourses>, currentSemester: int): void
+unlockLockedCoursesAndSetAvailable(s: Student, completedCourses: List<CompletedCourses>, lockedCourses: HashMap<String, List<String>>): void
+checkAvailableCourse(s: Student, currentSemesterCompleted: List<CompletedCourses>, currentSemesterFailed: List<FailedCourses>, lockedCourses: HashMap<String, List<String>>, currentSemesterCourses: HashMap<String, List<String>>):void
+checkCourseGiven(s: Student): void
+removeUnnamedCourses(s: Student): void
+checkCourseHasPrerequisite(courseCode: String): void
+checkPrerequisiteCourseIsGiven(s: Student, courseCode: String, semester: int): boolean
+courseIsGivenAlready(s: Student, courseCode: String): boolean
+setStudentAdvisor(s: Student): void
+generateAvailableCourses(students: Student[], advisors: Advisor[], courses: Courses[]): void
+caseTwo(): void
+caseThree(): void
+caseFour(): void
+otherCases: void
+simulateSemester(s: Student, semester: String): void
+simulate(): void
+assignRandomGrades(): String
Courses
-name: String
-courseCode: String
-prerequisite: List<String>
-credit: int
-courseType: int
-semester: int
-quota: int
-courseGrade: String
-courseYear: int
-givenSemester: int
-theoreticalCourseHour: int
-practicalLessonHour: int
-listOfStudents: List<Student>
+getters(): type
+setters(type: Type): void
+getPreRequisiteName(): String
+addToListOfStudents(student: Student): void
FailedCourses
-courseName: String
-courseGrade: String
+getters(): type
+setters(type: Type): void
+toString(): String
CompletedCourses
-courseName: String
-courseGrade: String
-givenSemester: int
+getters(): type
+setters(type): void
+toString(): String
CalculateAvailables
-semesterOneCoursesNames: ArrayList<String>
-semesterTwoCoursesNames: ArrayList<String>
-semesterThreeCoursesNames: ArrayList<String>
-semesterFourCoursesNames: ArrayList<String>
-semesterFiveCoursesNames: ArrayList<String>
-semesterSixCoursesNames: ArrayList<String>
-semesterSevenCoursesNames: ArrayList<String>
-semesterEigthCoursesNames: ArrayList<String>
-calculatedSemesterTwoCourseNames: ArrayList<String>
-calculatedSemesterThreeCourseNames: ArrayList<String>
-calculatedSemesterFourCourseNames: ArrayList<String>
-calculatedSemesterFiveCourseNames: ArrayList<String>
-calculatedSemesterSixCourseNames: ArrayList<String>
-calculatedSemesterSevenCourseNames: ArrayList<String>
-calculatedSemesterEigthCourseNames: ArrayList<String>
-studentCoursesTook: List<String>
-courseName: String
-courseGrade: String
-prerequisite: String
-maxNumberOfSelectionForCourses: int
+setAttributes(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>)
+calculatedCoursesResetter(): void
+setAvailableCoursesForEachStudent(students: Student[], courses: Courses[], advisors: Advisor[]): void
+setStudentsForEachAdvisor(students: Student[], advisors: Advisor[]): void
+setStudentsForEachCourses(students: Student[], courses:Courses[]): void
Transcript
-completedCourses: List <CompletedCourses>
-failedCourses: List<FailedCourses>
-gpa: Double
-completedCredits: int
-advisorName: String
-studentSelectedCourses: List<String>
-completedCourseStrings: List<String>
-failedCoursesStrings: List<String>
-student: Student
-studentAdvisor: Advisor
+getters(): type
+setters(type: Type): void
+seperateFailedCourses(): void
+printTranscriptSpecificStudent(student: Student): void
+transformSpecificStudentTranscriptElementsToList(student: Student): void
+generateTranscriptJson(student: Student[]):void