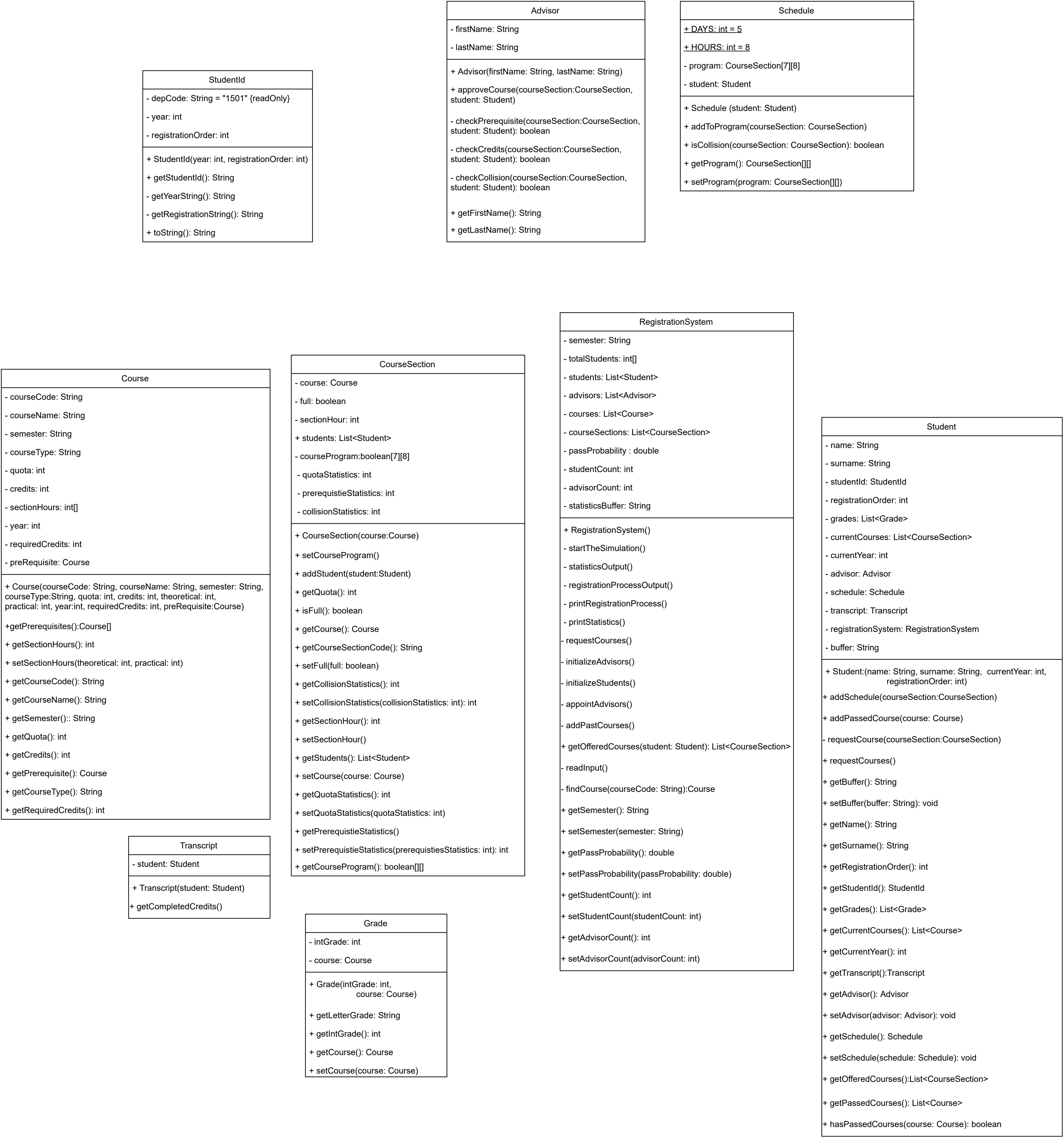

StudentId

- depCode: String = "1501" {readOnly}
- year: int
- registrationOrder: int

+ StudentId(year: int, registrationOrder: int)
+ getStudentId(): String
- getYearString(): String
- getRegistrationString(): String
+ toString(): String

Advisor

- firstName: String
- lastName: String

+ Advisor(firstName: String, lastName: String)
+ approveCourse(courseSection:CourseSection, student: Student)
- checkPrerequisite(courseSection:CourseSection, student: Student): boolean
- checkCredits(courseSection:CourseSection, student: Student): boolean
- checkCollision(courseSection:CourseSection, student: Student): boolean
+ getFirstName(): String
+ getLastName(): String

Schedule

+ DAYS: int = 5
+ HOURS: int = 8
- program: CourseSection[7][8]
- student: Student

+ Schedule (student: Student)
+ addToProgram(courseSection: CourseSection)
+ isCollision(courseSection: CourseSection): boolean
+ getProgram(): CourseSection[][]
+ setProgram(program: CourseSection[][])

Course

- courseCode: String
- courseName: String
- semester: String
- courseType: String
- quota: int
- credits: int
- sectionHours: int[]
- year: int
- requiredCredits: int
- preRequisite: Course

+ Course(courseCode: String, courseName: String, semester: String, courseType:String, quota: int, credits: int, theoretical: int, practical: int, year:int, requiredCredits: int, preRequisite:Course)

+getPrerequisites():Course[]

+ getSectionHours(): int
+ setSectionHours(theoretical: int, practical: int)
+ getCourseCode(): String
+ getCourseName(): String
+ getSemester():: String
+ getQuota(): int
+ getCredits(): int
+ getPrerequisite(): Course
+ getCourseType(): String
+ getRequiredCredits(): int

CourseSection

- course: Course
- full: boolean
- sectionHour: int
+ students: List<Student>
- courseProgram:boolean[7][8]
- quotaStatistics: int
- prerequistieStatistics: int
- collisionStatistics: int

+ CourseSection(course:Course)
+ setCourseProgram()
+ addStudent(student:Student)
+ getQuota(): int
+ isFull(): boolean
+ getCourse(): Course
+ getCourseSectionCode(): String
+ setFull(full: boolean)
+ getCollisionStatistics(): int
+ setCollisionStatistics(collisionStatistics: int): int
+ getSectionHour(): int
+ setSectionHour()
+ getStudents(): List<Student>
+ setCourse(course: Course)
+ getQuotaStatistics(): int
+ setQuotaStatistics(quotaStatistics: int)
+ getPrerequistieStatistics()
+ setPrerequistieStatistics(prerequistiesStatistics: int): int
+ getCourseProgram(): boolean[][]

RegistrationSystem

- semester: String
- totalStudents: int[]
- students: List<Student>
- advisors: List<Advisor>
- courses: List<Course>
- courseSections: List<CourseSection>
- passProbability : double
- studentCount: int
- advisorCount: int
- statisticsBuffer: String

+ RegistrationSystem()
- startTheSimulation()
- statisticsOutput()
- registrationProcessOutput()
- printRegistrationProcess()
- printStatistics()
- requestCourses()
- initializeAdvisors()
- initializeStudents()
- appointAdvisors()
- addPastCourses()
+ getOfferedCourses(student: Student): List<CourseSection>
- readInput()
- findCourse(courseCode: String):Course
+ getSemester(): String
+ setSemester(semester: String)
+ getPassProbability(): double
+ setPassProbability(passProbability: double)
+ getStudentCount(): int
+ setStudentCount(studentCount: int)
+ getAdvisorCount(): int
+ setAdvisorCount(advisorCount: int)

Student

- name: String
- surname: String
- studentId: StudentId
- registrationOrder: int
- grades: List<Grade>
- currentCourses: List<CourseSection>
- currentYear: int
- advisor: Advisor
- schedule: Schedule
- transcript: Transcript
- registrationSystem: RegistrationSystem
- buffer: String

+ Student:(name: String, surname: String, currentYear: int, registrationOrder: int)
+ addSchedule(courseSection:CourseSection)
+ addPassedCourse(course: Course)
- requestCourse(courseSection:CourseSection)
+ requestCourses()
+ getBuffer(): String
+ setBuffer(buffer: String): void
+ getName(): String
+ getSurname(): String
+ getRegistrationOrder(): int
+ getStudentId(): StudentId
+ getGrades(): List<Grade>
+ getCurrentCourses(): List<Course>
+ getCurrentYear(): int
+ getTranscript():Transcript
+ getAdvisor(): Advisor
+ setAdvisor(advisor: Advisor): void
+ getSchedule(): Schedule
+ setSchedule(schedule: Schedule): void
+ getOfferedCourses():List<CourseSection>
+ getPassedCourses(): List<Course>
+ hasPassedCourses(course: Course): boolean

Transcript

- student: Student

+ Transcript(student: Student)
+ getCompletedCredits()

Grade

- intGrade: int
- course: Course

+ Grade(intGrade: int, course: Course)
+ getLetterGrade: String
+ getIntGrade(): int
+ getCourse(): Course
+ setCourse(course: Course)