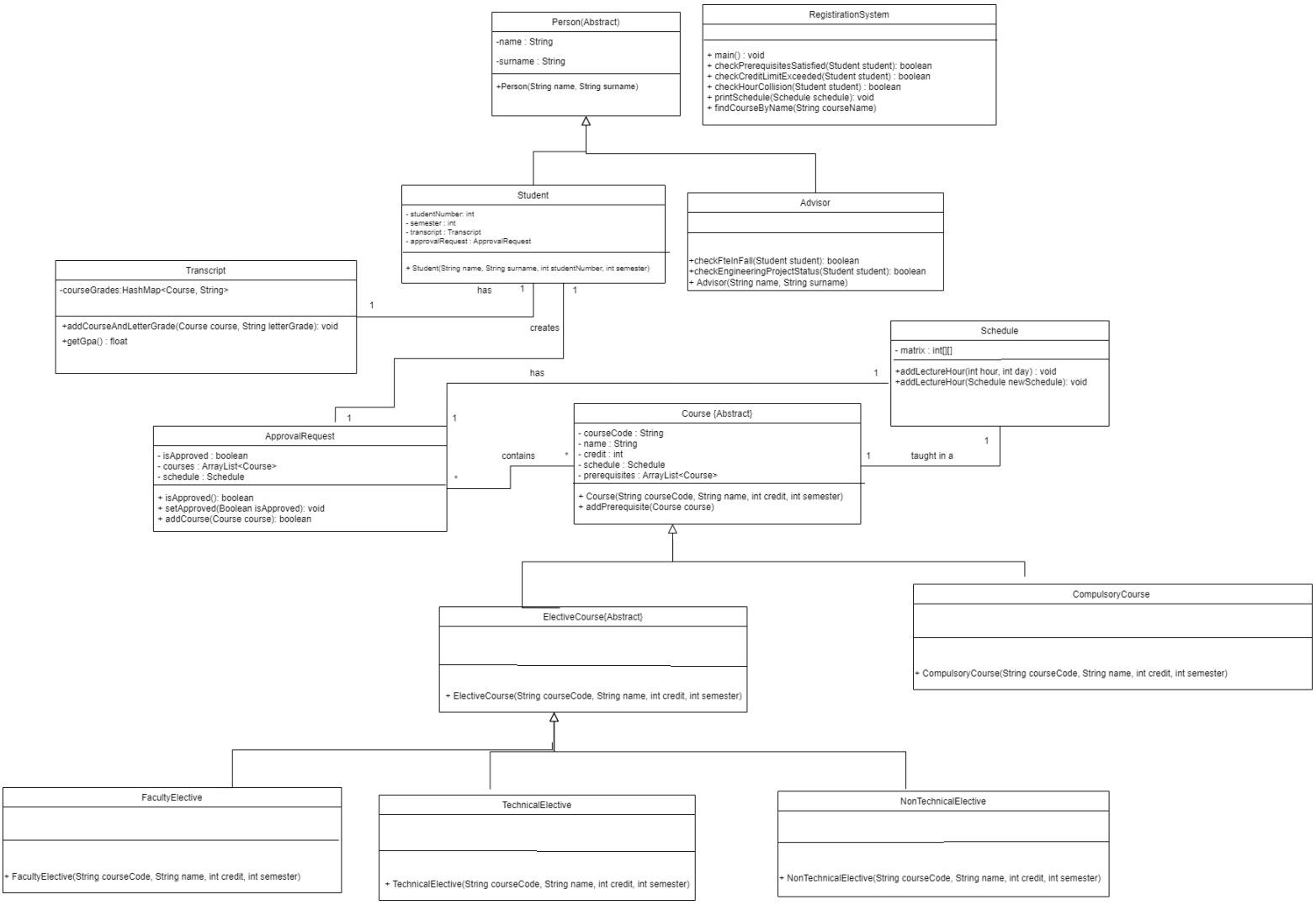
Person(Abstract)
-name : String
-surname : String
+Person(String name, String surname)
RegistirationSystem
+ main() : void
+ checkPrerequisitesSatisfied(Student student): boolean
+ checkCreditLimitExceeded(Student student) : boolean
+ checkHourCollision(Student student) : boolean
+ printSchedule(Schedule schedule): void
+ findCourseByName(String courseName)
Student
- studentNumber: int
- semester : int
- transcript : Transcript
- approvalRequest : ApprovalRequest
+ Student(String name, String surname, int studentNumber, int semester)
Advisor
+checkFteInFall(Student student): boolean
+checkEngineeringProjectStatus(Student student): boolean
+ Advisor(String name, String surname)
Transcript
-courseGrades:HashMap<Course, String>
+addCourseAndLetterGrade(Course course, String letterGrade): void
+getGpa() : float
has
1
1
1
creates
Schedule
- matrix : int[][]
+addLectureHour(int hour, int day) : void
+addLectureHour(Schedule newSchedule): void
has
1
1
1
ApprovalRequest
- isApproved : boolean
- courses : ArrayList<Course>
- schedule : Schedule
+ isApproved(): boolean
+ setApproved(Boolean isApproved): void
+ addCourse(Course course): boolean
contains
*
*
Course {Abstract}
- courseCode : String
- name : String
- credit : int
- schedule : Schedule
- prerequisites : ArrayList<Course>
+ Course(String courseCode, String name, int credit, int semester)
+ addPrerequisite(Course course)
1
taught in a
1
ElectiveCourse{Abstract}
+ ElectiveCourse(String courseCode, String name, int credit, int semester)
CompulsoryCourse
+ CompulsoryCourse(String courseCode, String name, int credit, int semester)
FacultyElective
+ FacultyElective(String courseCode, String name, int credit, int semester)
TechnicalElective
+ TechnicalElective(String courseCode, String name, int credit, int semester)
NonTechnicalElective
+ NonTechnicalElective(String courseCode, String name, int credit, int semester)