SDJ1 Exam Assignment (Plan B)

addGrade:
Only add the grade, if it is for a course that is in the student's list of courses.

hasValidCoronaPassport:
A Corona passport is valid if the current date is not before the date the passport was issued, and at the same time any of these three cases are true:
- The reason is "Vaccinated".
- The reason is "Tested" and the passport date is not more than 3 days ago.
- The reason is "Infected" and the passport date is more than 14 days ago and not more than 180 days ago.

Reason for a Corona passport can be one of the following: "Vaccinated", "Tested", or "Infected"

Date
- day : int
- month : int
- year : int

+ Date(day : int, month : int, year : int)
+ isBefore(date2 : Date) : boolean
- isLeapYear() : boolean
- daysInMonth() : int
- nextDay() : void
+ numberOfDaysUntil(date2 : Date) : int
+ today() : Date
+ copy() : Date
+ equals(obj : Object) : boolean
+ toString() : String

Constructor:
A lesson is not allowed to be created with an invalid time.

getDuration:
The method returns a Time object with hour, minute, and second set to the amount of time between the start and end of the lesson.

hasValidTime:
A lesson is not allowed to start before 08:20:00, end after 21:10:00, or have a start time that is after the end time.

getDateTimeString:
The method returns the date, start time, and end time, in the format dd/MM/yyyy hh:mm:ss - hh:mm:ss
E.g. 10/05/2021 12:45:00 - 16:05:00

isPDF:
A PDF file has the extension "pdf" or "PDF".

CoronaPassport
- reason : String

+ CoronaPassport(reason : String, date : Date)
+ getReason() : String
+ getDate() : Date
+ toString() : String

Resource
- fileName : String
- extension : String

+ Resource(fileName : String, extension : String)
+ isPDF() : boolean
+ equals(obj : Object) : boolean
+ toString() : String

Student
- studentNumber : int
- name : String

+ Student(studentNumber : int, name : String)
+ getStudentNumber() : int
+ getName() : String
+ addCourse(course : Course) : void
+ addGrade(grade : int, course : Course) : void
+ addCoronaPassport(passport : CoronaPassport) : void
+ removeCoronaPassport() : void
+ getCoronaPassport() : CoronaPassport
+ hasValidCoronaPassport() : boolean
+ getGradeAverage() : double
+ getAllGrades() : Grade[]
+ getAllCourses() : Course[]

Course
- name : String

+ Course(name : String)
+ getName() : String
+ getNumberOfLessons() : int
+ addLesson(lesson : Lesson) : void
+ removeLesson(lesson : Lesson) : void
+ hasLessonOnDate(date : Date) : boolean
+ getAllLessons() : ArrayList<Lesson>
+ getAllTopics() : ArrayList<String>
+ getOnlineLessons() : ArrayList<OnlineLesson>
+ getOnlineLessonsWithCameraRequired() : ArrayList<OnlineLesson>
+ getDiscordLessons() : ArrayList<DiscordLesson>
+ getZoomLessonsWithBreakoutRooms() : ArrayList<ZoomLesson>
+ getAllResources() : ArrayList<Resource>
+ toString() : String

Lesson
- topic : String

+ Lesson(topic : String, date : Date, start : Time, end : Time, res : Resource[])
+ getTopic() : String
+ getDate() : Date
+ getResources() : Resource[]
+ getAllPDFs() : ArrayList<Resource>
+ getDuration() : Time
+ hasValidTime(startTime : Time, endTime : Time) : boolean
+ getDateTimeString() : String
+ equals(obj : Object) : boolean
+ toString() : String

Time
- hour : int
- minute : int
- second : int

+ Time(h : int, m : int, s : int)
+ Time(totalTimeInSeconds : int)
+ convertToSeconds() : int
+ isBefore(time2 : Time) : boolean
+ timeUntil(time2 : Time) : Time
+ copy() : Time
+ equals(obj : Object) : boolean
+ toString() : String

Grade
- grade : int

+ Grade(grade : int, course : Course)
+ getGrade() : int
+ getCourse() : Course
+ copy() : Grade
+ toString() : String

toString:
The method returns the grade and the name of the course.

OnlineLesson
- cameraRequired : boolean

+ OnlineLesson(topic : String, date : Date, start : Time, end : Time, res : Resource[], camera : boolean)
+ isCameraRequired() : boolean
+ *nameOfSoftware() : String*
+ equals(obj : Object) : boolean
+ toString() : String

nameOfSoftware:
The method returns the name of the software used for the lesson. E.g. Zoom or Discord

DiscordLesson
- channelName : String

+ DiscordLesson(topic : String, date : Date, start : Time, end : Time, res : Resource[], camera : boolean, channel : String)
+ getChannelName() : String
+ nameOfSoftware() : String
+ equals(obj : Object) : boolean
+ toString() : String

ZoomLesson
- link : String
- breakoutRooms : boolean

+ ZoomLesson(topic : String, date : Date, start : Time, end : Time, res : Resource[], camera : boolean, link : String, breakout : boolean)
+ getLink() : String
+ usesBreakoutRooms() : boolean
+ nameOfSoftware() : String
+ equals(obj : Object) : boolean
+ toString() : String

Relationships/labels: - coronaPassport 0..1, - date 1, - courses 0..*, - schedule 0..*, - resources 0..*, - start 1, - end 1, - grades 0..*, - course 1, - date 1
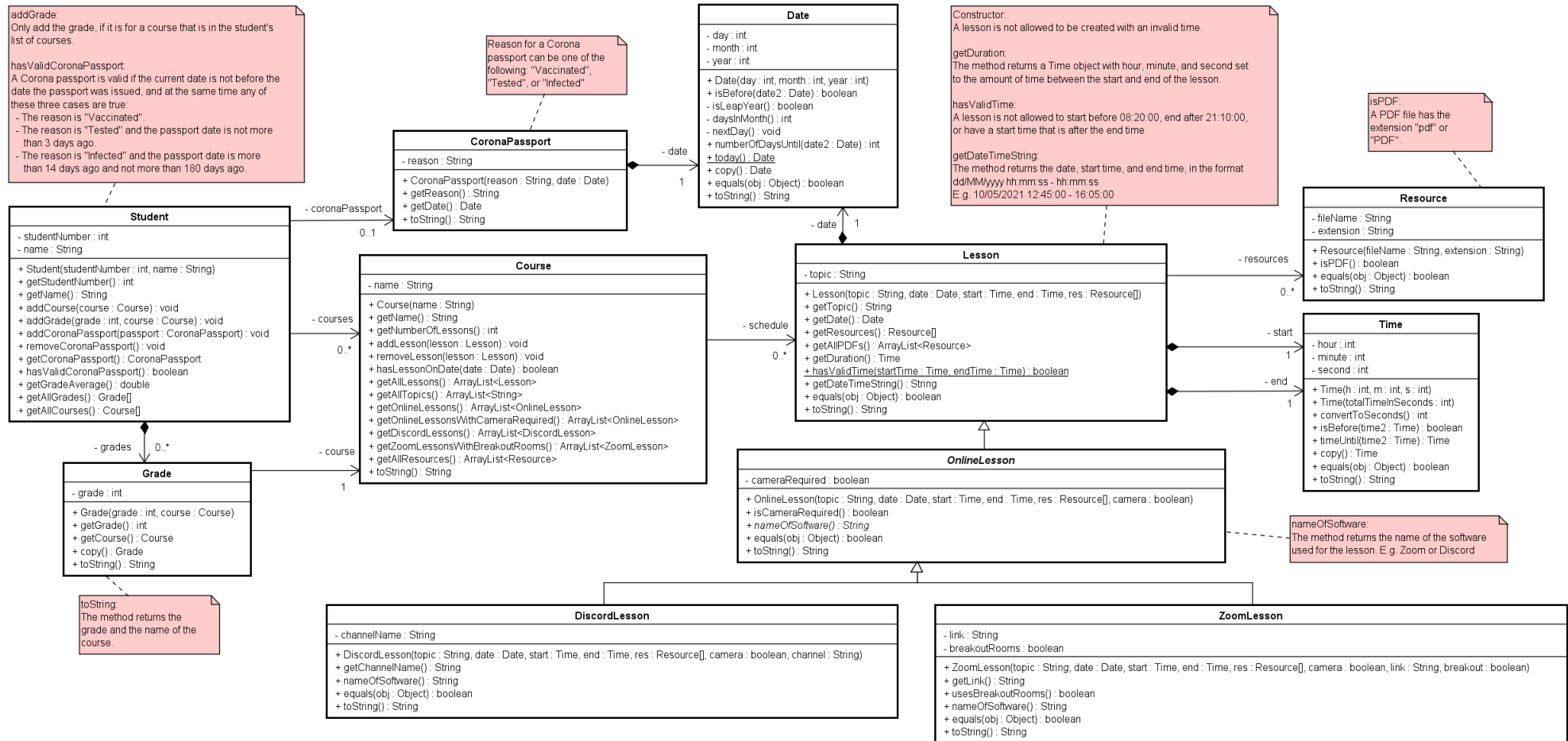
1. Implement the classes in the UML class diagram.

2. Create a test class, with a main method, where you test the functionality of the implemented classes.