

CourseOfStudy
+ name: string + studentNumber: int + semester: int + cosUsed: bool + visitedCourses: List<Course>

Block
+ timespan: TimeSpan + blockNumber: int + blockCourses: List<Course> + day: DayEnum
+ Block(TimeSpan zeit, int blockNumber, DayEnum day)

Helper
+ ContainsAllItems<T>(this IEnumerable<T> a, this IEnumerable<T> b) : bool + CountStudents() : int + MapCourseOfStudyNameToCourseOfStudy(string courseOfStudyName) + TimeCheck(DateTime a, DayEnum d, Prof p) : bool + SetBlock(OptionalCourse o) : void + PrintCourse(string courseofStudyName) : void + PrintPossibleOptionalCourses(CourseOfStudy courseOfStudy) : void + PrintOptionalCourseData(Block b, Day day, OptionalCourse o) : void

Prof
+ name: string + profUsed: bool + courses: List<string> + busy: List<TimeSpanDay>

Room
+ name: string + seats: int + roomUsed: bool + roomEquipment: List<string>

TimeSpanDay
+ start: DateTime + end: DateTime + day: DayEnum
+ TimeSpanDay(DateTime start, DateTime end)

TimeSpan
+ start: DateTime + end: DateTime
+ TimeSpan(DateTime start, DateTime end)

OptionalCourse
+ name: string + description: string + room: Room + studentNumber: int + prof: Prof + block: Block + timeSpanDay: TimeSpanDay + optionalCourseUsed: bool + equipment: List<string> + shouldPrint: bool

Course
+ name: string + description: string + studentNumber: int + room: Room + prof: Prof + block: Block + courseUsed: bool + equipment: List<string> + coursesOfStudy: List<CourseOfStudy>

<<enumeration>> DayEnum
Montag Dienstag Mittwoch Donnerstag Freitag

Day
+ day: DayEnum + blocks: List<Block>
+ Day(DayEnum day)

CoursesOfStudy Json
"name" "studentNumber" "semester" "visitedCourses" "visitedCourses:name"

TimeTableGenerator
+ week: List<Day> + times: TimeSpan[] + courses: List<Course> + optionalCourses: List<OptionalCourse> + rooms: List<Room> + profs: List<Prof> + coursesOfStudy: List<CourseOfStudy>
+ LoadJsonRooms () : void + LoadJsonProf () : void + LoadJsonCoursesOfStudy () : void + LoadJsonCourses () : void + LoadvisitedCourses () : void + LoadProfstoCourses () : void + LoadOptionalCourses () : void + Initializer () : void + GenerateOptionalTimetable () : void + GenerateTimetable () : void + SortBy () : void

Module Json
"name" "description" "equipment" "coursesOfStudy" "coursesOfStudy:name"

Prof Json
"name" "courses" "busy:start" "busy:end" "busy:day"

Room Json
"name" "seats" "roomEquipment"

OptionalCourse Json
"name" "description" "equipment" "timeSpanDay" "timeSpanDay:start" "timeSpanDay:end" "timeSpanDay:day"