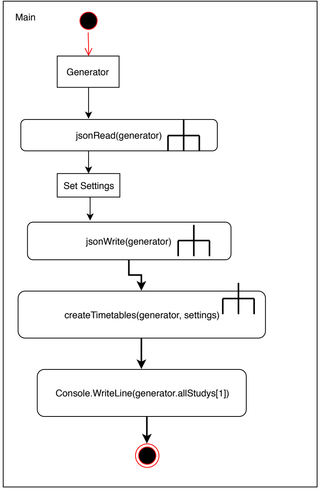
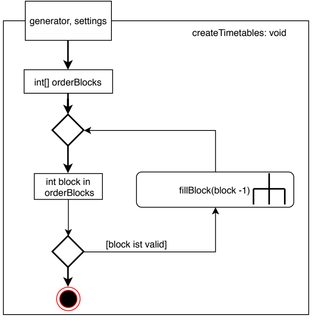
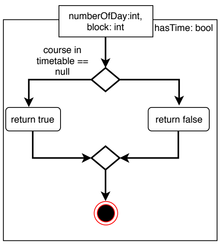
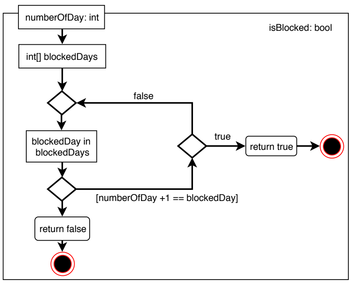
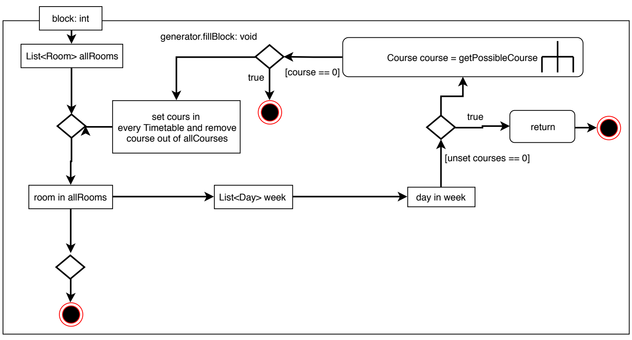
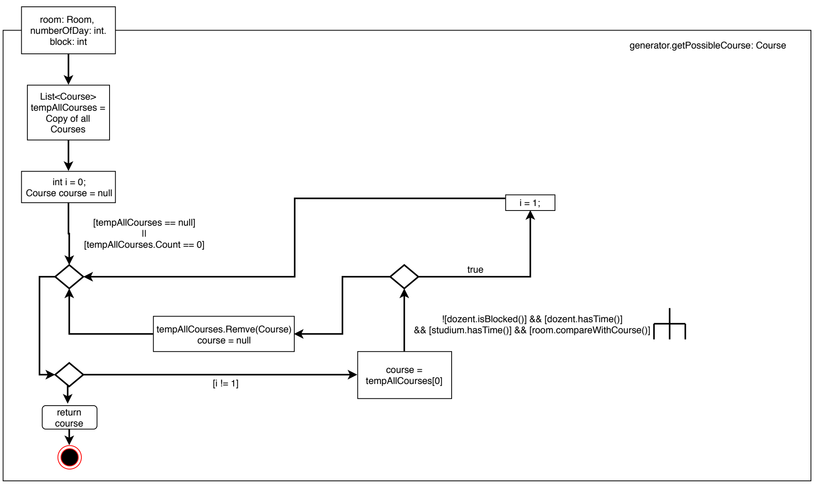

Main
Generator
jsonRead(generator)
Set Settings
jsonWrite(generator)
createTimetables(generator, settings)
Console.WriteLine(generator.allStudys[1])

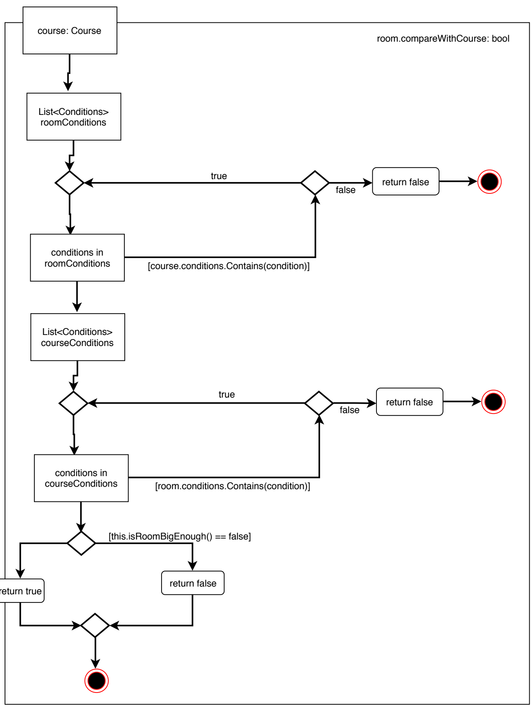

generator, settings
createTimetables: void
int[] orderBlocks
int block in orderBlocks
fillBlock(block -1)
[block ist valid]

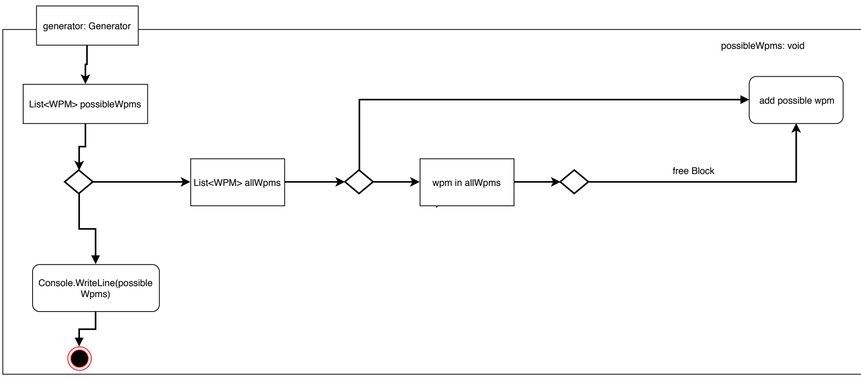

numberOfDay:int, block: int
hasTime: bool
course in timetable == null
return true
return false


numberOfDay: int
isBlocked: bool
int[] blockedDays
false
blockedDay in blockedDays
true
return true
[numberOfDay +1 == blockedDay]
return false


block: int
generator.fillBlock: void
List<Room> allRooms
Course course = getPossibleCourse
[course == 0]
true
set cours in every Timetable and remove course out of allCourses
true
return
[unset courses == 0]
room in allRooms
List<Day> week
day in week


room: Room, numberOfDay: int. block: int
generator.getPossibleCourse: Course
List<Course> tempAllCourses = Copy of all Courses
int i = 0; Course course = null
[tempAllCourses == null] || [tempAllCourses.Count == 0]
i = 1;
true
tempAllCourses.Remve(Course) course = null
![dozent.isBlocked()] && [dozent.hasTime()] && [studium.hasTime()] && [room.compareWithCourse()]
course = tempAllCourses[0]
[i != 1]
return course


course: Course
room.compareWithCourse: bool
List<Conditions> roomConditions
true
false
return false
conditions in roomConditions
[course.conditions.Contains(condition)]
List<Conditions> courseConditions
true
false
return false
conditions in courseConditions
[room.conditions.Contains(condition)]
[this.isRoomBigEnough() == false]
return true
return false


generator: Generator
possibleWpms: void
List<WPM> possibleWpms
add possible wpm
List<WPM> allWpms
wpm in allWpms
free Block
Console.WriteLine(possible Wpms)