



Main:Console

Main:Organization

Main:Lesson

SpaceCatalog

- spaceCatalog: ArrayList<Space>= new ArrayList<>();

+ addSpace(space:Space): void
+ spaceExists(id:String): boolean
+ printSpaces(): void
+ findSpace(spaceld: String): Space

Space

- personLimit: int
- address: string
- type: string
- isRented: boolean
- city: string
- spaceld: String
- lessons: ArrayList<Lesson>
- idcounter: int=1

+Space(String address,String type,isRented: boolean, city:String, personLimit: int)
- generateUniqueId(): String
+ getAddress(): string
+ getType(): string
+ isRented(): boolean
+ getCity(): String
+ getPersonLimit(): int
+ getSpaceld(): String
+ setAddress(address:String): void
+ setType(type:String)
+ setIsRented(isRented:boolean): void
+ setCity(city:String): void
+ setPersonLimit(limit:int): void
+ setSpaceld(id: String): void

Schedule

- startTime: LocalTime
- endTime: LocalTime
- startDate: LocalDate
- endDate: LocalDate
- day: String
- lessons: ArrayList<Lesson>
- space: Space
- timeFormatter: DateTimeFormatter
- dateFormatter: DateTimeFormatter

+ Schedule(startTime: String, endTime: String, startDate: String, endDate: String, space: Space, day: String)
+ getSpace(): Space
+ setSpace(space: Space): void
+ getStartTime(): LocalTime
+ getEndTime(): LocalTime
+ getStartDate(): LocalDate
+ getEndDate(): LocalDate
+ getDay(): String
+ setStartTime(startTime: String): void
+ setEndTime(endTime: String): void
+ setStartDate(startDate: String): void
+ setEndDate(endDate: String): void
+ setDay(day: String): void
+ getLessons(): List<Lesson>
+ timeslotAvailable(other: Schedule): boolean
+ printSchedule(): void

ScheduleCatalog

- scheduleCatalog: ArrayList<Schedule>= new ArrayList<>();

+ addSchedule(schedule:Schedule): void
+ getSpacesForSchedule(schedule:Schedule): void
+ checkScheduleConflict(newSchedule: Schedule): boolean
+ printSpacesSchedule(spaceld: String): void