

AttendeeManager		
m	AttendeeManager()	
m	createAttendee(String, String, Boolean)	Attendee
m	createSpeaker(String, String)	Speaker
m	registeredSpeaker(String)	boolean
m	inSystem(String, String)	boolean
m	addEventToSpeakerList(Speaker, String)	void
m	removeEventFromSpeakerList(Speaker, String)	void
m	getItinerary(Attendee)	ArrayList<String>
m	getSpeakingList(Speaker)	ArrayList<String>
m	changeSpeaker(String, String)	void
m	checkIsOrganizer(Attendee)	Boolean
m	signUp(Attendee, String)	void
m	dropOut(Attendee, String)	void
m	usernameToAttendeeObject(String)	Optional<Attendee>
m	eventToAttendees(Event)	ArrayList<Attendee>
p	allAttendees	ArrayList<Attendee>
p	allSpeakers	ArrayList<Speaker>

EventManager		
m	EventManager()	
m	createEvent(String, String, int, String, int, int, int, int)	void
m	freeSpeakerCheck(ArrayList<LocalDateTime>, String)	boolean
m	freeTitleCheck(String)	boolean
m	nameToEvent(String)	Optional<Event>
m	eventToAttendees(Event)	ArrayList<String>
m	signUp(Event, String)	void
m	dropOut(Event, String)	void
m	changeSpeaker(Event, String)	boolean
m	cancelEvent(Event)	void
m	main(String[])	void
p	events	ArrayList<Event>

RoomManager		
m	RoomManager()	
m	addRoom(int, int)	boolean
m	idToRoom(int)	Room
m	checkRoomInSystem(int)	boolean
m	checkIfRoomAvailable(Room, LocalDateTime, LocalDateTime)	boolean
m	book(Room, String, LocalDateTime, LocalDateTime)	void
m	checkConflict(LocalDateTime, LocalDateTime, LocalDateTime[])	boolean
m	unbook(Room, String)	void
m	main(String[])	void
p	rooms	ArrayList<Room>

MessageManager		
f	messages	ArrayList<Message>
m	MessageManager()	
m	createMessage(String, String, String)	Message
m	createMessage(ArrayList<String>, String, String)	Message
m	reply(Message, String, String)	Message
m	getSendBy(String)	ArrayList<String>
m	getReceivedBy(String)	ArrayList<String>
m	getAllMessagesFrom(String, String)	ArrayList<String>
m	recipientsBuilder(ArrayList<String>)	String