<<abstract>>

Schedule # events HashMap<String, List<List<Date>>> # scheduleName: String # scheduleDate: Stirng # author: String # status: String # scheduleID: String # deletedEvents HashMap<String, List<Date>> - type: String="Default"(final) + Schedule(scheduleName: String, scheduleDate: String, author: String, status: String) + checkDateFormat(String startTimeString, String endTimeString): boolean + getEvents(): HashMap<String, List<List<Date>>> + getScheduleDate(): String + getAuthor(): String + getStatus(): String + getScheduleID(): String + setStatus(status: String): void + setEvents(events: HashMap<String, List<List<Date>>>): void + getType(): String + getDeletedEvents(): HashMap<String, List<Date>> + SetDeletedEvents(): void + toString(): String

- type: String = "Daily"(final)

+ DailySchedule(scheduleName: String, scheduleDate: String, author: String, status: String)
+ getType(): String
+ checkDateFormat(startTimeString: String, endTimeString: String): boolean
+ toString(): String

ScheduleManager schedulesList: HashMap<String, List<Schedule>> - scheduleTempMap: HashMap<String, String> - schedulesFriend: HashMap<String, List<Schedule>> - factory: ScheduleFactory + ScheduleManager() + setSchedulesList(scheduleList: HashMap<String, List<Schedule>>): void + setScheduleTempMap(scheduleTempMap: HashMap<String, String>): void + setSchedulesFriend(schedulesFriend: HashMap<String, List<Schedule>>): void + getSchedulesList(): HashMap<String, List<Schedule>> + getScheduleTempMap(): HashMap<String, String> + getSchedulesFriend(): HashMap<String, List<Schedule>> + deleteSchedule(scheduleID: String): boolean + changeStatus(scheduleID: String, status: String): void - changeStatusHelper(s: Schedule, status: String): void + addEvent(scheduleID: String, eventName: String, startTimeString: String, endTimeString: String, t: TemplateManager): void + addValidEvent(DateList: List<Date>, s: Schedule, eventName: String): void + validEvent(s: Schedule, t: TemplateManager, start: Date, end: Date): boolean + validBetweenTime(s: Schedule, start: Date, end: Date, temp: Template): void + validDuration(start: Date, end: Date, min: double, max: double): void + deleteEvent (scheduleID: String, EventName: String, startTimeString: String, endTimeString: String): boolean + deleteValidEvent(listOfDates: List<List<Date>>, dateST: Date, dateET: Date, EventName: String, events: HashMap<String, List<List<Date>>>, s: Schedule): boolean + createSchedule(status: String, scheduleName: String, scheduleDate: String, author: String, t: TemplateManager, templateId: String): String + getAllUserSchedule(userID: String): List<Schedule> + getAllFriendsSchedule(userID: String): List<Schedule> + getScheduleEvents(scheduleID: String, userID: String): HashMap<String, List<List<Date>>> + getAllPublicSchedule(): List<Schedule> + checkScheduleBelongs(userID String, scheduleID String): boolean + getScheduleByID(scheduleID: String): Schedule - isDateValid(date: String, dateFormat: String): Date + getDeletedEvents(scheduleID: String): HashMap<String, List<Date>> + addDeletedEvents(scheduleID: String, eventName: String, newDeleted: List<Date>): void + removeDeletedEvents(scheduleID: String, eventName: String): void + clearDeletedEvents(): void + shareScheduleToFriends(scheduleId: String, um: UserManager): void

ScheduleSystem - sm: ScheduleManager - um: UserManager - tm: TemplateManager - deletedEventName: HashMap<String, List<String>> - previousStatus: ArrayList<String> - deletedSchedule: ArrayList<Schedule> - deletedTempID: ArrayList<String> + ScheduleSystem(sm: ScheduleManager, um: UserManager, tm: TemplateManager) + createSchedule(userID: String, status: String, scheduleName: String, scheduleDate: String, templateID: String): boolean + deleteSchedule(scheduleID: String, userId: String): boolean + addDeletedScheduleAndTemp(s: Schedule, tempID: String): void + recoverSchedule(): void + addDeletedTempID(tempID: String): void + clearDeletedSchedulesAndEvents(): void + addPreviousStatus(scheduleID: String): void + restoreStatus(scheduleId: String): void + clearPreviousStatus(): void + addDeletedEvent(eventName: String): void + recoverEvent(scheduleId: String, userId: String): void + displayPublicSchedule(): void + displayUserSchedule(userID: String): void + displayFriendsSchedule(userID: String): void + displayScheduleEvents(scheduleID: String, userID: String): void + checkNoSchedule(): boolean + GetScheduleRecords(): HashMap<Integer, String> + updateFriends(userID: String, friID: String): void - updateOneFriend(userID: String, friID: String): void + checkStatusInputValid(userType: String, sc: Scanner): String + enterDatebyTemplateType(templateId: String): void + getStartEndbyType(scheduleId: String, sc: Scanner): ArrayList<String> + chooseMySchedule(userId: String): boolean + generateNewEvent(scheduleId: String, sc: Scanner): void

ScheduleUI -facade: Facade - sc: Scanner -userType: String -userId: String -templateId: String +ScheduleUI(facade:Facade, scanner:Scanner) +setUserId(userId: String): void +setUserType(userType: String): void +runSchedule(templateId: String): void -runScheduleOption1(): boolean -runScheduleOption5(): void -runScheduleOption6(): void -editYourSchedule(): void -editOthersSchedule(): void -ifUnfreeze(s: Schedule): boolean +editSchedule(scheduleId: String): void -editScheduleOption2(scheduleId: String, eventName: String): boolean -editScheduleOption3(scheduleId: String): void

+ getSchedule(type: String, scheduleDate: String, scheduleName: String, author: String, status: String): Schedule
+ setUserId(userId: String): void
- isDateValid(date: String, dateFormat: String): void