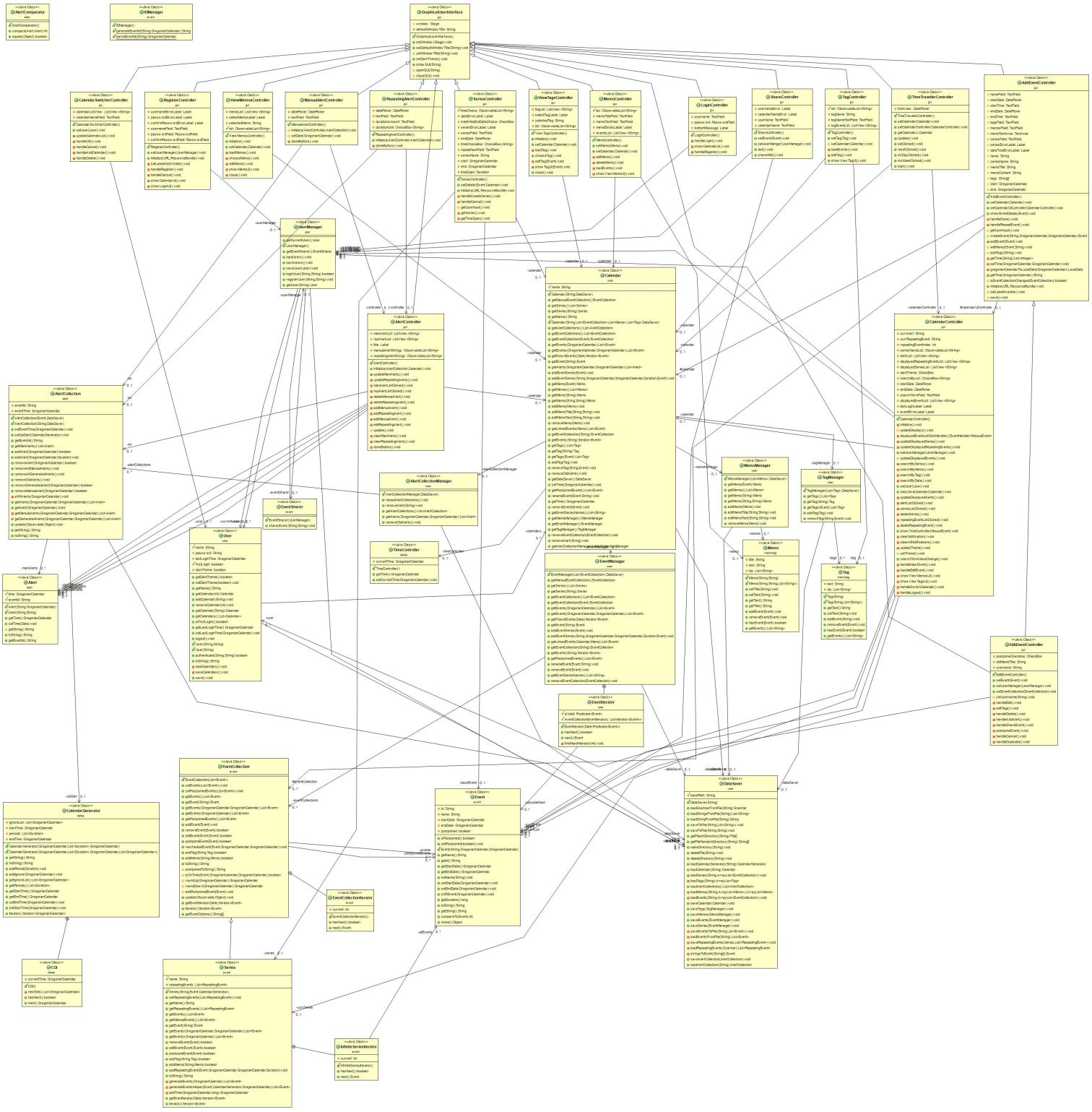
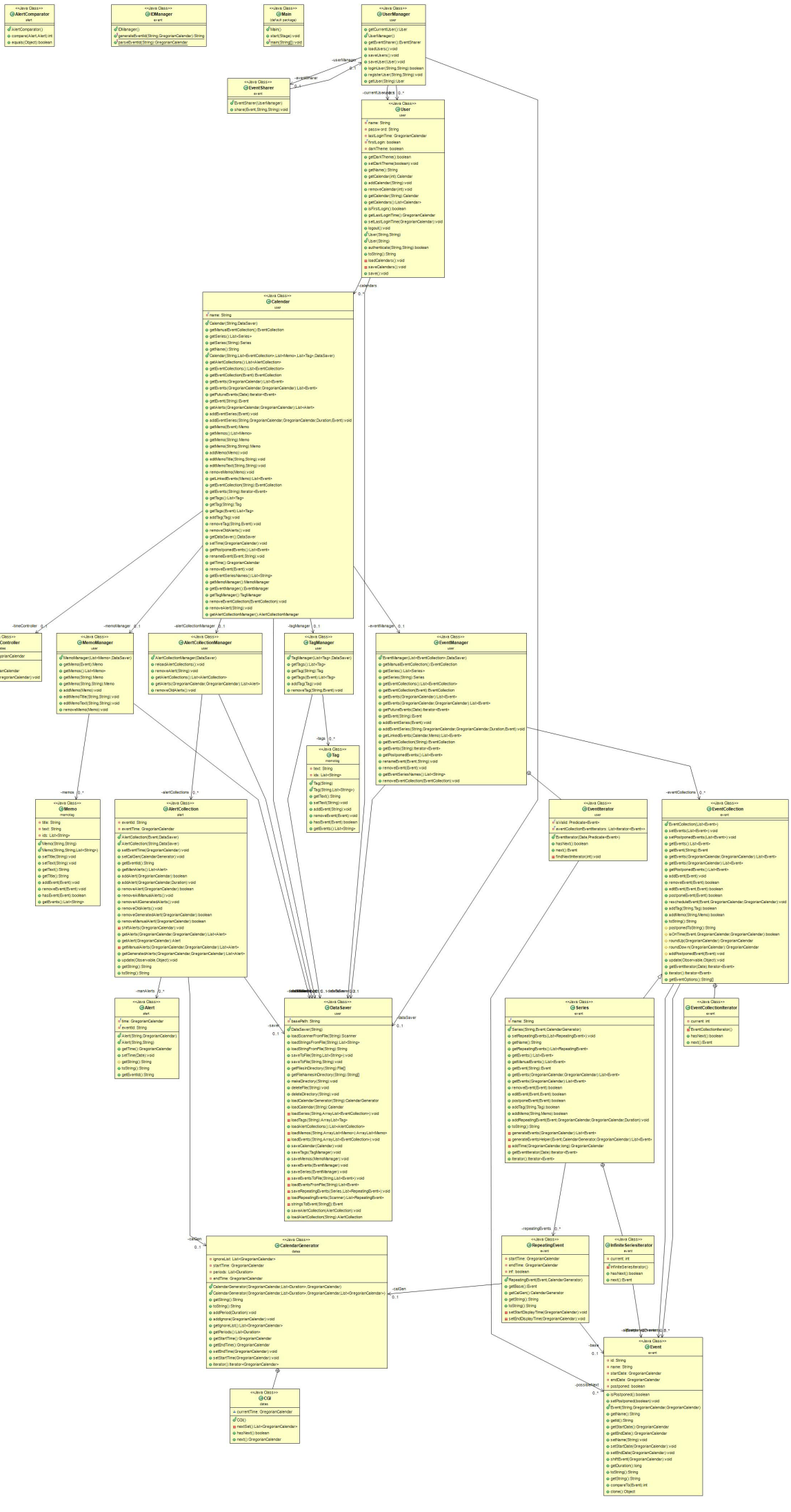

AlertComparator
IDManager
GraphicalUserInterface
CalendarSwitcherController
RegisterController
ViewMemosController
ManualAlertController
RepeatingAlertController
SeriesController
ViewTagsController
MemoController
LoginController
ShareController
TagController
TimeTravelerController
AddEventController
UserManager
Calendar
AlertController
CalendarController
AlertCollection
AlertCollectionManager
EventSharer
User
TimeController
MemoManager
TagManager
Memo
Tag
EditEventController
EventManager
EventIterator
EventCollection
CalendarGenerator
Event
DataSaver
EventCollectionIterator
CG
Series
InfiniteSeriesIterator


AlertComparator
alert
AlertComparator()
compare(Alert,Alert): int
equals(Object): boolean
EMManager
event
EMManager()
generateEventId(String,GregorianCalendar):String
parseEventId(String):GregorianCalendar
Main
(default package)
Main()
start(Stage): void
main(String[]): void
UserManager
user
getCurrentUser():User
UserManager()
getEventSharer():EventSharer
loadUsers():void
saveUsers():void
saveUser(User):void
loginUser(String,String):boolean
registerUser(String,String):void
getUser(String):User
-userManager
0..1
-eventSharer
0..1
EventSharer
event
EventSharer(UserManager)
share(Event,String,String):void
-currentUsers
0..*
User
user
name: String
passwrd: String
lastLoginTime: GregorianCalendar
firstLogin: boolean
darkTheme: boolean
getDarkTheme():boolean
setDarkTheme(boolean):void
getName():String
getCalendar(int):Calendar
addCalendar(String):void
removeCalendar(int):void
getCalendar(String):Calendar
getCalendars():List<Calendar>
isFirstLogin():boolean
getLastLoginTime():GregorianCalendar
setLastLoginTime(GregorianCalendar):void
logout():void
User(String,String)
User(String)
authenticate(String,String):boolean
toString():String
loadCalendars():void
saveCalendars():void
save():void
-calendars
0..*
Calendar
user
name: String
Calendar(String,DataSaver)
getManualEventCollection():EventCollection
getSeries():List<Series>
getSeries(String):Series
getName():String
Calendar(String,List<EventCollection>,List<Memo>,List<Tag>,DataSaver)
getAlertCollections():List<AlertCollection>
getEventCollections():List<EventCollection>
getEventCollection(Event):EventCollection
getEvents(GregorianCalendar):List<Event>
getEvents(GregorianCalendar,GregorianCalendar):List<Event>
getFutureEvents(Date):Iterator<Event>
getEvent(String):Event
getAlerts(GregorianCalendar,GregorianCalendar):List<Alert>
addEventSeries(Event):void
addEventSeries(String,GregorianCalendar,GregorianCalendar,Duration,Event):void
getMemo(Event):Memo
getMemos():List<Memo>
getMemo(String):Memo
getMemo(String,String):Memo
addMemo(Memo):void
editMemoTitle(String,String):void
editMemoText(String,String):void
removeMemo(Memo):void
getLinkedEvents(Memo):List<Event>
getEventCollection(String):EventCollection
getEvents(String):Iterator<Event>
getTags():List<Tag>
getTag(String):Tag
getTags(Event):List<Tag>
addTag(Tag):void
removeTag(String,Event):void
removeOldAlerts():void
getDataSaver():DataSaver
setTime(GregorianCalendar):void
getPostponedEvents():List<Event>
renameEvent(Event,String):void
getTime():GregorianCalendar
removeEvent(Event):void
getEventSeriesNames():List<String>
getMemoManager():MemoManager
getEventManager():EventManager
getTagManager():TagManager
removeEventCollection(EventCollection):void
removeAlert(String):void
getAlertCollectionManager():AlertCollectionManager
-timeController
0..1
-memoManager
0..1
-alertCollectionManager
0..1
-tagManager
0..1
-eventManager
0..1
TimeController
dates
currentTime: GregorianCalendar
TimeController()
getTime():GregorianCalendar
setCurrentTime(GregorianCalendar):void
MemoManager
user
MemoManager(List<Memo>,DataSaver)
getMemo(Event):Memo
getMemos():List<Memo>
getMemo(String):Memo
getMemo(String,String):Memo
addMemo(Memo):void
editMemoTitle(String,String):void
editMemoText(String,String):void
removeMemo(Memo):void
AlertCollectionManager
user
AlertCollectionManager(DataSaver)
reloadAlertCollections():void
removeAlert(String):void
getAlertCollections():List<AlertCollection>
getAlerts(GregorianCalendar,GregorianCalendar):List<Alert>
removeOldAlerts():void
TagManager
user
TagManager(List<Tag>,DataSaver)
getTags():List<Tag>
getTag(String):Tag
getTags(Event):List<Tag>
addTag(Tag):void
removeTag(String,Event):void
EventManager
user
EventManager(List<EventCollection>,DataSaver)
getManualEventCollection():EventCollection
getSeries():List<Series>
getSeries(String):Series
getEventCollections():List<EventCollection>
getEventCollection(Event):EventCollection
getEvents(GregorianCalendar):List<Event>
getEvents(GregorianCalendar,GregorianCalendar):List<Event>
getFutureEvents(Date):Iterator<Event>
getEvent(String):Event
addEventSeries(Event):void
addEventSeries(String,GregorianCalendar,GregorianCalendar,Duration,Event):void
getLinkedEvents(Calendar,Memo):List<Event>
getEventCollection(String):EventCollection
getEvents(String):Iterator<Event>
getPostponedEvents():List<Event>
renameEvent(Event,String):void
removeEvent(Event):void
getEventSeriesNames():List<String>
removeEventCollection(EventCollection):void
-tags
0..*
Tag
memotag
text: String
ids: List<String>
Tag(String)
Tag(String,List<String>)
getText():String
setText(String):void
addEvent(String):void
removeEvent(Event):void
hasEvent(Event):boolean
getEvents():List<String>
-memos
0..*
Memo
memotag
title: String
text: String
ids: List<String>
Memo(String,String)
Memo(String,String,List<String>)
setTitle(String):void
setText(String):void
getText():String
getTitle():String
addEvent(Event):void
removeEvent(Event):void
hasEvent(Event):boolean
getEvents():List<String>
-alertCollections
0..*
AlertCollection
alert
eventId: String
eventTime: GregorianCalendar
AlertCollection(Event,DataSaver)
AlertCollection(String,DataSaver)
setEventTime(GregorianCalendar):void
setCalGen(CalendarGenerator):void
getEventId():String
getManualAlerts():List<Alert>
addAlert(GregorianCalendar):boolean
addAlert(GregorianCalendar,Duration):void
removeAlert(GregorianCalendar):boolean
removeAllManualAlerts():void
removeAllGeneratedAlerts():void
removeOldAlerts():void
removeGeneratedAlert(GregorianCalendar):boolean
removeManualAlert(GregorianCalendar):boolean
shiftAlerts(GregorianCalendar):void
getAlerts(GregorianCalendar,GregorianCalendar):List<Alert>
getAlert(GregorianCalendar):Alert
getManualAlerts(GregorianCalendar,GregorianCalendar):List<Alert>
getGeneratedAlerts(GregorianCalendar,GregorianCalendar):List<Alert>
update(Observable,Object):void
getString():String
toString():String
-manualAlerts
0..*
Alert
alert
time: GregorianCalendar
eventId: String
Alert(String,GregorianCalendar)
Alert(String,String)
getTime():GregorianCalendar
setTime(Date):void
getString():String
toString():String
getEventId():String
-saver
0..1
-dataSaver
0..1
DataSaver
user
basePath: String
DataSaver(String)
loadScannerFromFile(String):Scanner
loadStringsFromFile(String):List<String>
loadStringFromFile(String):String
saveToFile(String,List<String>):void
saveToFile(String,String):void
getFilesInDirectory(String):File[]
getFileNamesInDirectory(String):String[]
makeDirectory(String):void
deleteFile(String):void
deleteDirectory(String):void
loadCalendarGenerator(String):CalendarGenerator
loadCalendar(String):Calendar
loadSeries(String,ArrayList<EventCollection>):void
loadTags(String):ArrayList<Tag>
loadAlertCollections():List<AlertCollection>
loadMemos(String,ArrayList<Memo>):ArrayList<Memo>
loadEvents(String,ArrayList<EventCollection>):void
saveCalendar(Calendar):void
saveTags(TagManager):void
saveMemos(MemoManager):void
saveEvents(EventManager):void
saveSeries(EventManager):void
saveEventsToFile(String,List<Event>):void
loadEventsFromFile(String):List<Event>
saveRepeatingEvents(Series,List<RepeatingEvent>):void
loadRepeatingEvents(Scanner):List<RepeatingEvent>
stringsToEvent(String[]):Event
saveAlertCollections(AlertCollection):void
loadAlertCollection(String):AlertCollection
-calGen
0..1
CalendarGenerator
dates
ignoreList: List<GregorianCalendar>
startTime: GregorianCalendar
periods: List<Duration>
endTime: GregorianCalendar
CalendarGenerator(GregorianCalendar,List<Duration>,GregorianCalendar)
CalendarGenerator(GregorianCalendar,List<Duration>,GregorianCalendar,List<GregorianCalendar>)
getString():String
toString():String
addPeriod(Duration):void
addIgnore(GregorianCalendar):void
getIgnoreList():List<GregorianCalendar>
getPeriods():List<Duration>
getStartTime():GregorianCalendar
getEndTime():GregorianCalendar
setEndTime(GregorianCalendar):void
setStartTime(GregorianCalendar):void
Iterator():Iterator<GregorianCalendar>
CGI
dates
currentTime: GregorianCalendar
CGI()
nextSet():List<GregorianCalendar>
hasNext():boolean
next():GregorianCalendar
EventIterator
user
isValid: Predicate<Event>
eventCollectionEventIterators: List<Iterator<Event>>
EventIterator(Date,Predicate<Event>)
hasNext():boolean
next():Event
findNextIterator(int):void
-eventCollections
0..*
EventCollection
event
EventCollection(List<Event>)
setEvents(List<Event>):void
setPostponedEvents(List<Event>):void
getEvents():List<Event>
getEvent(String):Event
getEvents(GregorianCalendar,GregorianCalendar):List<Event>
getEvents(GregorianCalendar):List<Event>
getPostponedEvents():List<Event>
addEvent(Event):void
removeEvent(Event):boolean
editEvent(Event,Event):boolean
postponeEvent(Event):boolean
rescheduleEvent(Event,GregorianCalendar,GregorianCalendar):void
addTag(String,Tag):boolean
addMemo(String,Memo):boolean
toString():String
postponedToString():String
isOnTime(Event,GregorianCalendar,GregorianCalendar):boolean
roundUp(GregorianCalendar):GregorianCalendar
roundDown(GregorianCalendar):GregorianCalendar
addPostponedEvent(Event):void
update(Observable,Object):void
getEventIterator(Date):Iterator<Event>
iterator():Iterator<Event>
getEventOptions():String[]
EventCollectionIterator
event
current: int
EventCollectionIterator()
hasNext():boolean
next():Event
Series
event
name: String
Series(String,Event,CalendarGenerator)
setRepeatingEvents(List<RepeatingEvent>):void
getName():String
getRepeatingEvents():List<RepeatingEvent>
getEvents():List<Event>
getManualEvents():List<Event>
getEvent(String):Event
getEvents(GregorianCalendar,GregorianCalendar):List<Event>
getEvents(GregorianCalendar):List<Event>
removeEvent(Event):boolean
editEvent(Event,Event):boolean
postponeEvent(Event):boolean
addTag(String,Tag):boolean
addMemo(String,Memo):boolean
addRepeatingEvent(Event,GregorianCalendar,GregorianCalendar,Duration):void
toString():String
generateEvents(GregorianCalendar):List<Event>
generateEventsHelper(Event,CalendarGenerator,GregorianCalendar):List<Event>
addTime(GregorianCalendar,long):GregorianCalendar
getEventIterator(Date):Iterator<Event>
iterator():Iterator<Event>
-repeatingEvents
0..*
RepeatingEvent
event
startTime: GregorianCalendar
endTime: GregorianCalendar
inf: boolean
RepeatingEvent(Event,CalendarGenerator)
getBase():Event
getCalGen():CalendarGenerator
getString():String
toString():String
setStartDisplayTime(GregorianCalendar):void
setEndDisplayTime(GregorianCalendar):void
InfiniteSeriesIterator
event
current: int
InfiniteSeriesIterator()
hasNext():boolean
next():Event
-base
0..1
-possibleNext
0..*
Event
event
id: String
name: String
startDate: GregorianCalendar
endDate: GregorianCalendar
postponed: boolean
isPostponed():boolean
setPostponed(boolean):void
Event(String,GregorianCalendar,GregorianCalendar)
getName():String
getId():String
getStartDate():GregorianCalendar
getEndDate():GregorianCalendar
setName(String):void
setStartDate(GregorianCalendar):void
setEndDate(GregorianCalendar):void
shiftEvent(GregorianCalendar):void
getDuration():long
toString():String
getString():String
compareTo(Event):int
clone():Object