Time

- hour:int
- minute:int
- daynight:string

+ Time()
+ Time(int,int,string)
+ Time(const Time)
+ gethour( ) : int
+ getminutes() : int
+ getdaynight() : string
+ sethour(int):void
+ setminutes(int): void
+ setdaynight(string):void
+friendoperator<<(ostream,Time):ostream
+friendoperator>>(istream,Time):istream
+ operator<(Time):bool
+ operator>(Time):bool
+ operator==(Time):bool
+ operator!=(Time):bool

Appointment

-start: Time
- end: Time
- interval:int

+ Appointment()
+ Appointment(Time ,Time)
+ Appointment(const Appointment &)
+ calinterval():double
+ getstart():Time
+ getend():Time
+friend operator<<(ostream,appointment):
    ostream
+friend operator>> (istream,appointment):
    istream
+ operator==(Appointment):bool
+ operator!=(Appointment): bool
+ operator<(Appointment):bool
+ operator>(Appointment):bool
+ operator>=(Appointment):bool
+ operator<=(Appointment):bool

2   1

*

1

Calendar

- days:DayAppointments
- numOfDays:int

+Calendar()
+Calendar(int)
+InsertDaytoCalendar(DayAppointments ,int):void
+friend operator >>(istream , Calendar):istream
+friend operator<< (ostream, Calendar):ostream

*   1

DayAppointments

- appointments:Appointment
- numOfAppointments:int
- weekDay:string

+DayAppointments()
+DayAppointments(int ,string)
+DayAppointments(DayAppointments)
+ setweekday(string):void
+ setnumOfAppointments(int):void
+ getnumOfAppointments():int
+ getweekDay():string
+ checkoverlapped(int,Appointment):bool
+friendoperator<<(ostream,DayAppointments): ostream
+friendoperator>>(istream,DayAppointments):istream
+ operator== (DayAppointments):bool
+ operator!= (DayAppointments):bool
+operator[ ](int):Appointment