

带有待办事项功能的日历 设计文档

1. 核心类方法详解

CalendarInterval 类 :该类用于表征了一个日期时间区间对象,精确到分钟。提供了两个核心的属性 : GregorianCalendar startTime 与 GregorianCalendar endTime , 分别表示区间的始末端点。	
CalendarInterval()	默认构造方法, 一般不使用。
CalendarInterval(GregorianCalendar startTime, GregorianCalendar endTime)	startTime : 起始时间 endTime : 结束时间 初始化起始时间与结束时间, 同时在其中判断是否为合法时间区间。若不是, 则抛出 IntervalException
CalendarInterval(int startYear, int startMonth, int startDay, int startHr, int startMin, int endYear, int endMonth, int endDay, int endHr, int endMin)	startYear, startMonth, startDay, startHr, startMin : 起始时间参数 endYear, endMonth, endDay, endHr, endMin : 结束时间参数 初始化起始时间与结束时间, 同时在其中判断是否为合法时间且为合法时间区间。若不是, 则抛出 IntervalException
GregorianCalendar getStartTime()	返回值 : startTime 属性值 获得 startTime 属性值, 主要用于封装
void setStartTime(GregorianCalendar startTime)	startTime : 新的起始时间 返回值 : 无 设置 startTime 属性值, 在设置前会判断新的时间区间是否为合法时间区间。若不是, 抛出 IntervalException
GregorianCalendar getEndTime()	返回值 : endTime 属性值 获得 endTime 属性值, 主要用于封装
void setEndTime(GregorianCalendar endTime)	endTime : 新的结束时间 设置 endTime 属性值, 在设置前会判断新的时间区间是否为合法时间区间。若不是, 抛出 IntervalException
boolean isOverlapped(CalendarDate date)	date : 要判断是否重叠的日期 返回值 : 判断是否重叠的结果 判断该时间区间是否 date 所表示的日期有重叠
boolean isOverlapped(CalendarInterval interval)	interval : 要判断是否重叠的时间区间 返回值 : 判断是否重叠的结果 判断该时间区间是否 interval 所表示的时间区间相重叠

Todo 类：该类用于表征了一个待办事项对象。提供了四个私有属性： String content ：存储该待办事项的内容 boolean usingInterval ：表征该待办事项是某个时间区间的待办事项，还是一个某天的待办事项 CalendarDate date ：若 usingInterval 为 false， date 代表该待办事项所处的日期。否则 date 为 null CalendarInterval interval ：若 usingInterval 为 true， interval 代表该待办事项所处的时间区间。否则 interval 为 null	
Todo()	默认构造方法，一般不使用。
Todo(String content, CalendarDate date)	content ：待办事项内容 date ：待办事项日期 初始化一个某天的待办事项，日期为 date ，内容为 content
Todo(String content, CalendarInterval interval)	content ：待办事项内容 interval ：待办事项时间区间 初始化一个某时间区间的待办事项，时间区间为 interval ，内容为 content
String getContent()	返回值 ：待办事项内容 获取待办事项的内容
void setContent(String content)	content ：新的待办事项内容 设置新的待办事项内容
boolean isUsingInterval()	返回值 ：是否使用时间区间定位待办事项 获取是否用时间区间定位待办事项
CalendarDate getDate()	返回值 ：待办事项所在日期 若为某天的待办事项，则返回待办事项所对应的日期。否则为 null
CalendarInterval getInterval()	返回值 ：待办事项所在时间区间 若为用时间区间定位的待办事项，则返回待办事项所对应的的时间区间。否则为 null

TodoList 类：该类代表了一个待办事项列表，可供添加、查询与删除。	
TodoList()	默认构造方法，生成一个待办事项列表。
void add(Todo todo)	Todo ：将要添加的待办事项 将待办事项 todo 添加入待办事项列表
ArrayList<Todo> search(CalendarInterval interval)	interval ：要查询的时间区间 返回值 ： interval 时间区间中的所有待办事项组成的列表 查询 interval 这段时间区间中的待办事项，返回这些待办事项组成的 ArrayList
ArrayList<Todo> search(CalendarDate date)	date ：要查询的日期 返回值 ： date 日期中的所有待办事项组成

	的列表 查询 date 这一天中的待办事项，返回这些待办事项组成的 ArrayList
void delete(Todo todo)	todo : 需要删除的待办事项 删除该待办事项列表中的某个待办事项

IntervalException 类：处理时间区间不合法异常的异常处理类。	
@Override String getMessage()	返回值：异常信息 获取异常信息的方法

同时为 DateUtil 类添加了一个新的方法

boolean isValidInterval(GregorianCalendar start, GregorianCalendar end)	start : 起始时间 end : 结束时间 判断 start 与 end 组成的时间对是否能组成合法的时间区间
--	---

2. GUI 设计

在原程序下方增加多个 Panel，用于添加、查询和删除工作



3. 测试用例

源代码中，CalendarDateTest, DateUtilTest, CalendarIntervalTest, TodoTest, TodoListTest 四个类为测试类。本次 lab 修改了 DateUtilTest，增加了 CalendarIntervalTest, TodoTest, TodoListTest 三个测试类。