

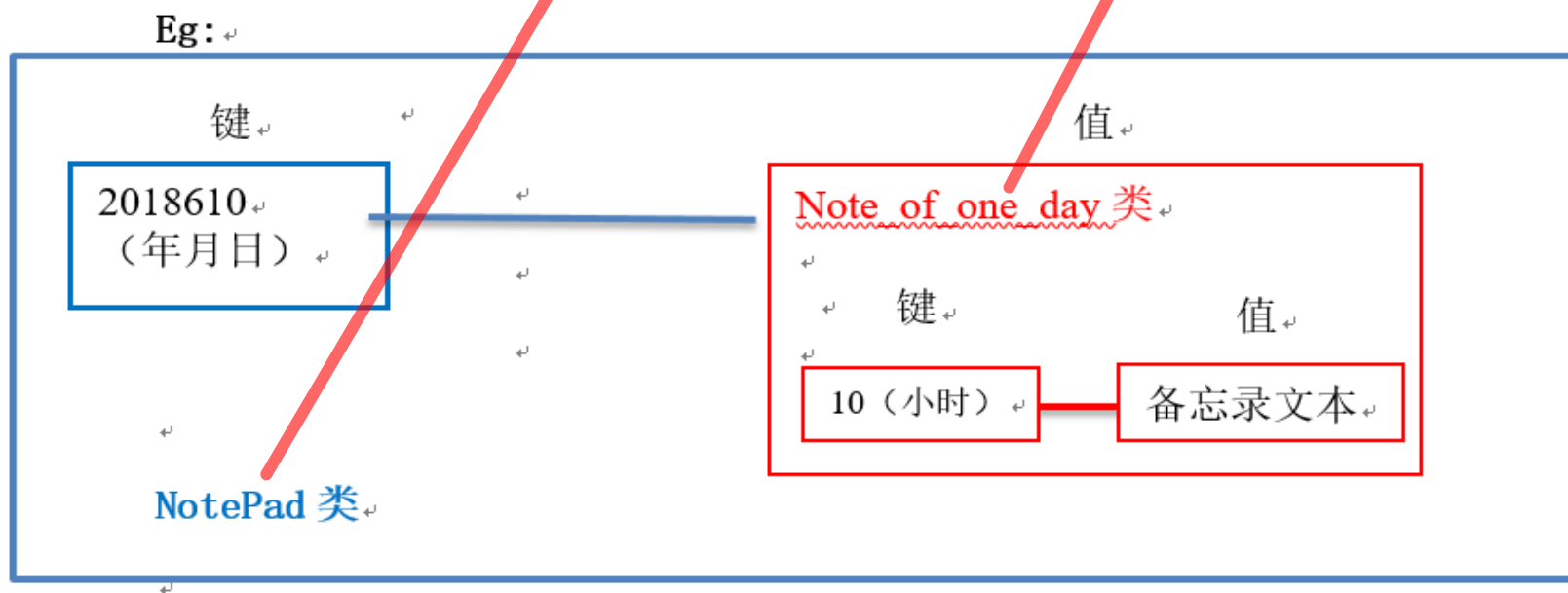
# 1.程序结构

三个主要文件为  
CalendarPad.java、  
NotePad.java、  
Note\_of\_one\_day.java



备忘录如何存储？

利用层次的键-值对  
集合结构

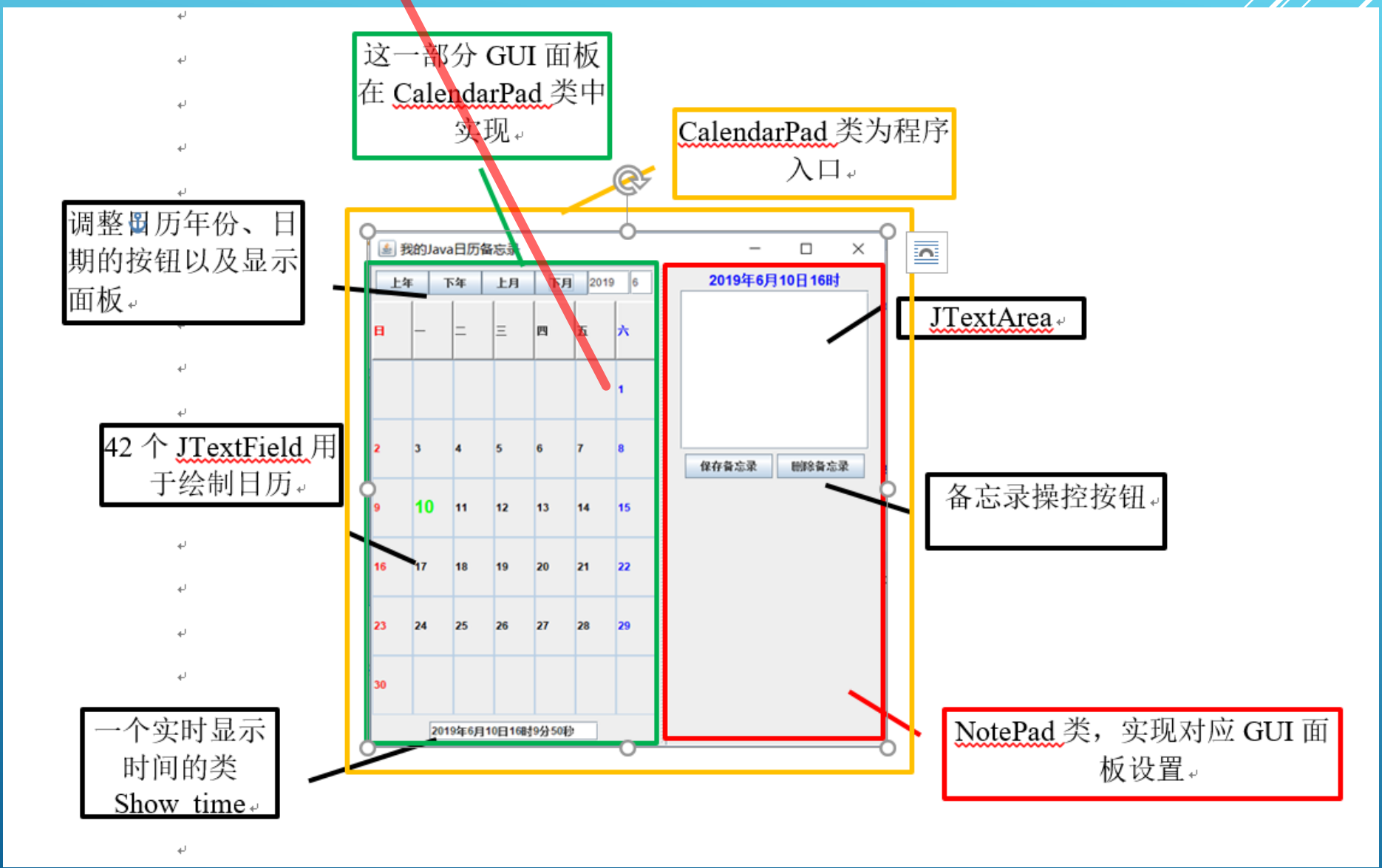


# 2.设计思路

## (1) GUI面板设计

利用Calendar类的方法可以获取某一天是一周的第几天，进而绘制日历

```
myCalendar=Calendar.getInstance();  
what_day_of_first_day=myCalendar.get(Calendar.DAY_OF_WEEK)-1;
```

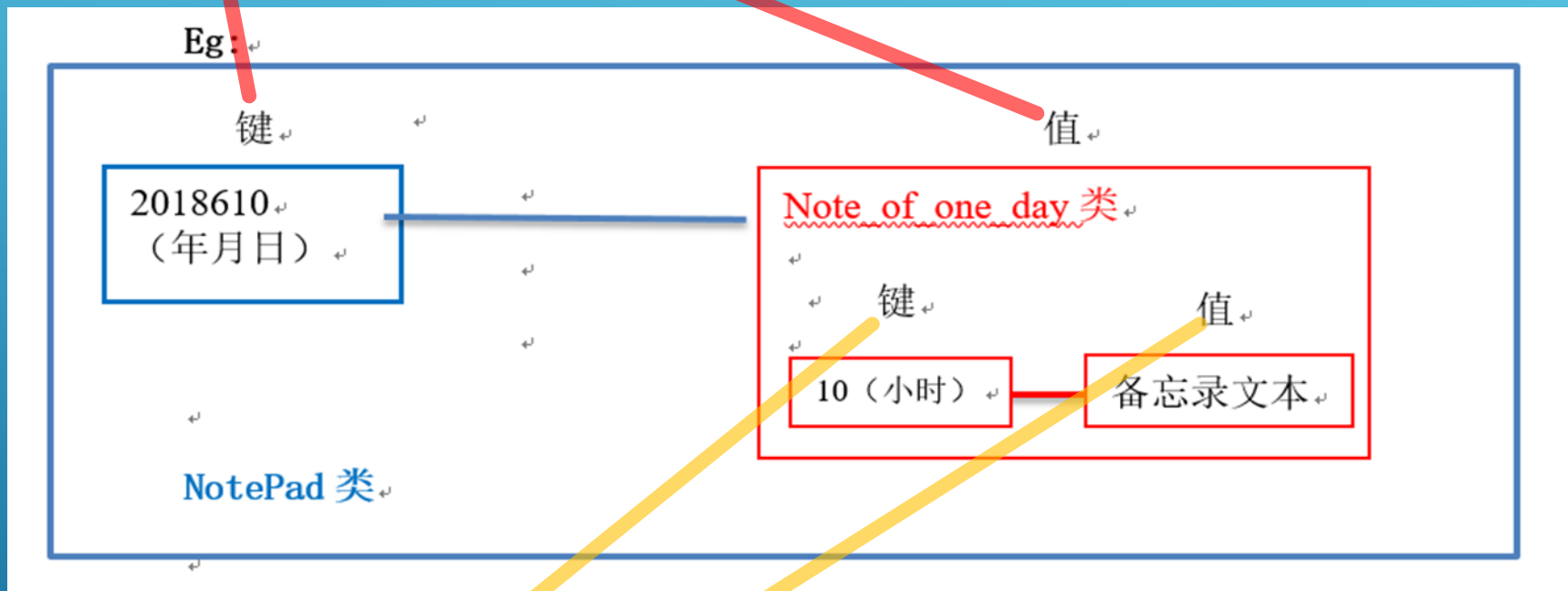


## (2) 备忘录存储

在NotePad类里  
有一个  
Hashtable变量

```
Hashtable<String, Note_of_one_day> if_contain_note=new Hashtable<String, Note_of_one_day>();  
//集合类型，用于存放某一天的备忘录，键是某一天的日期，值是Note_of_one_day对象
```

Note\_of\_one\_day  
类是存储备忘录具体  
内容的“最小”类：



在Note\_of\_one\_day类里  
有一个Hashtable变量

```
public Hashtable<Integer, String> specific_note=new Hashtable<Integer, String>();  
//具体存储24小时备忘录的Hash表
```

### (3) 如何提醒？

Note\_of\_one\_day类继承  
Runnable接口来实现线程功能

```
Hashtable<String,Note_of_one_day> if_contain_note=new Hashtable<String,Note_of_one_day>();  
//集合类型，用于存放某一天的备忘录，键是某一天的日期，值是Note_of_one_day对象
```

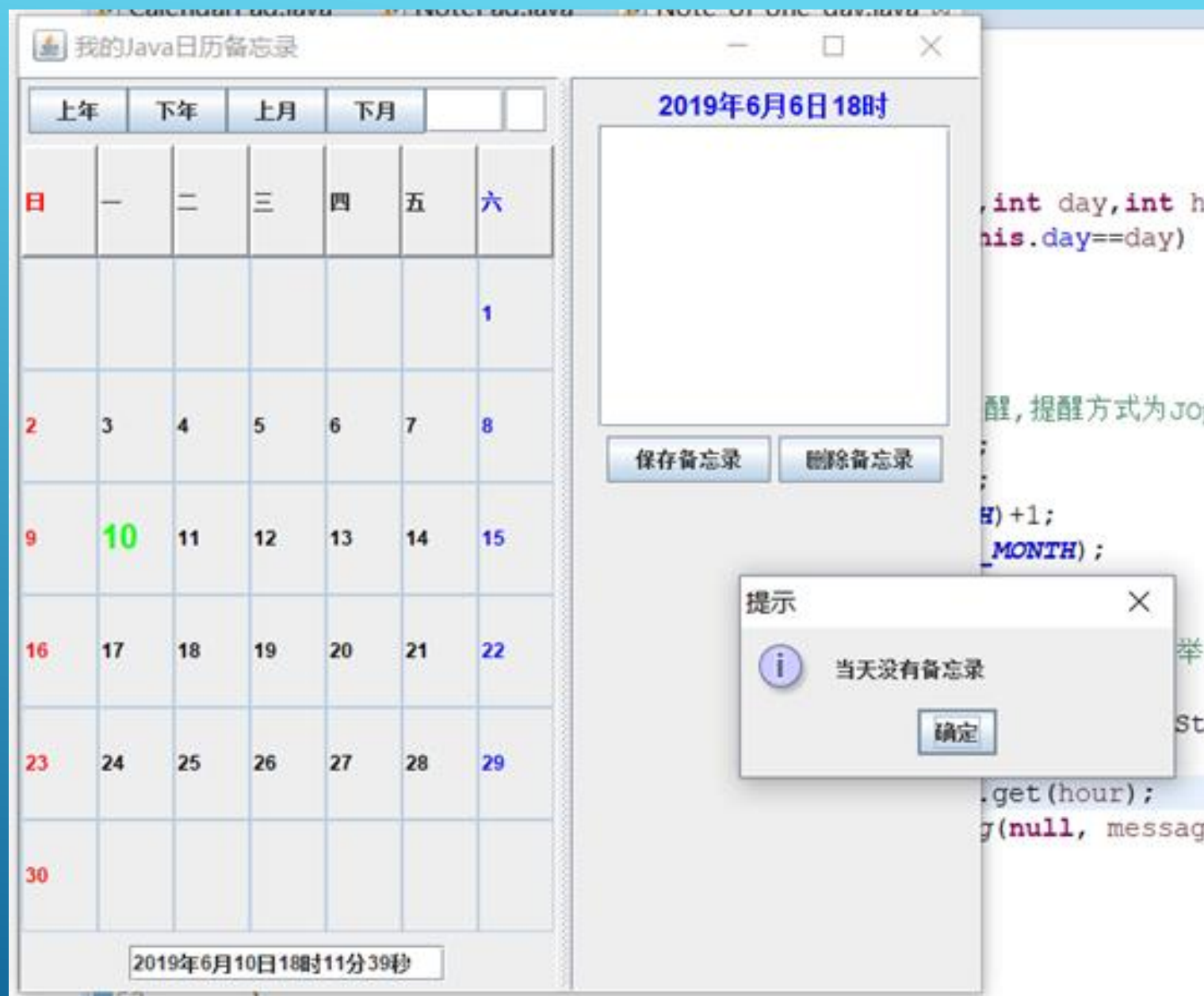
保存某一条备忘录 → 向if\_contain\_note中放入一个新的键-值对，也就是新建一个Note\_of\_one\_day类对象 → 启动Note\_of\_one\_day线程对象，每过一秒检查一下是否需要提醒

```
if_contain_note.put(key_of_day, new Note_of_one_day(year,month,day));
```

```
String op=JOptionPane.showInputDialog(this,"请选择要保存备忘录于哪个小时","提示",JOptionPane.QUESTION_MESSAGE);  
int selected_hour=Integer.parseInt(op);  
this.hour=selected_hour;  
if_contain_note.get(key_of_day).Save_a_note(year, month, day, selected_hour, content);  
new Thread(if_contain_note.get(key_of_day)).start();  
JOptionPane.showMessageDialog(null, "备忘录保存成功","提示",JOptionPane.INFORMATION_MESSAGE);
```

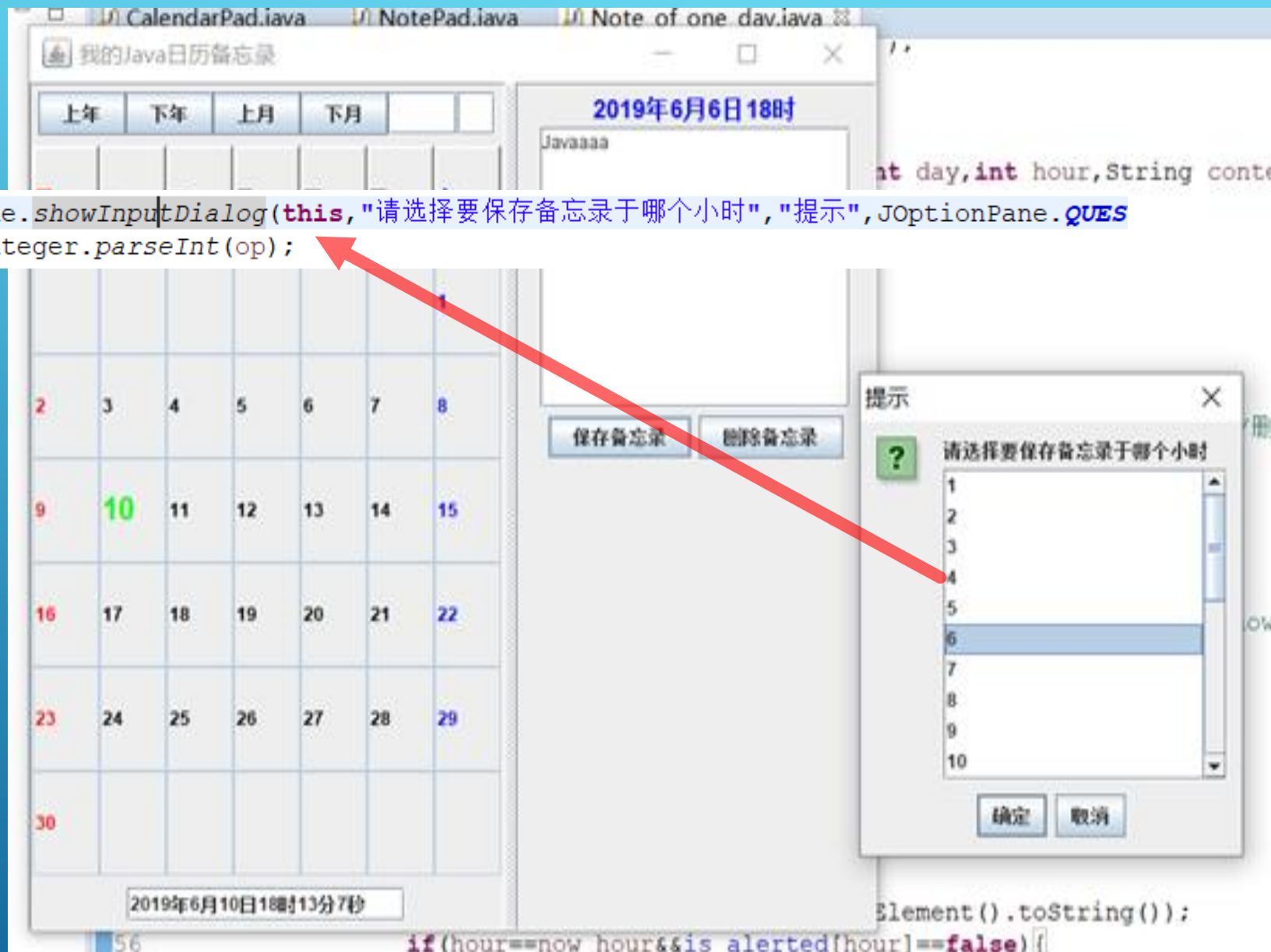
### 3.运行调试

点击日历的某一天

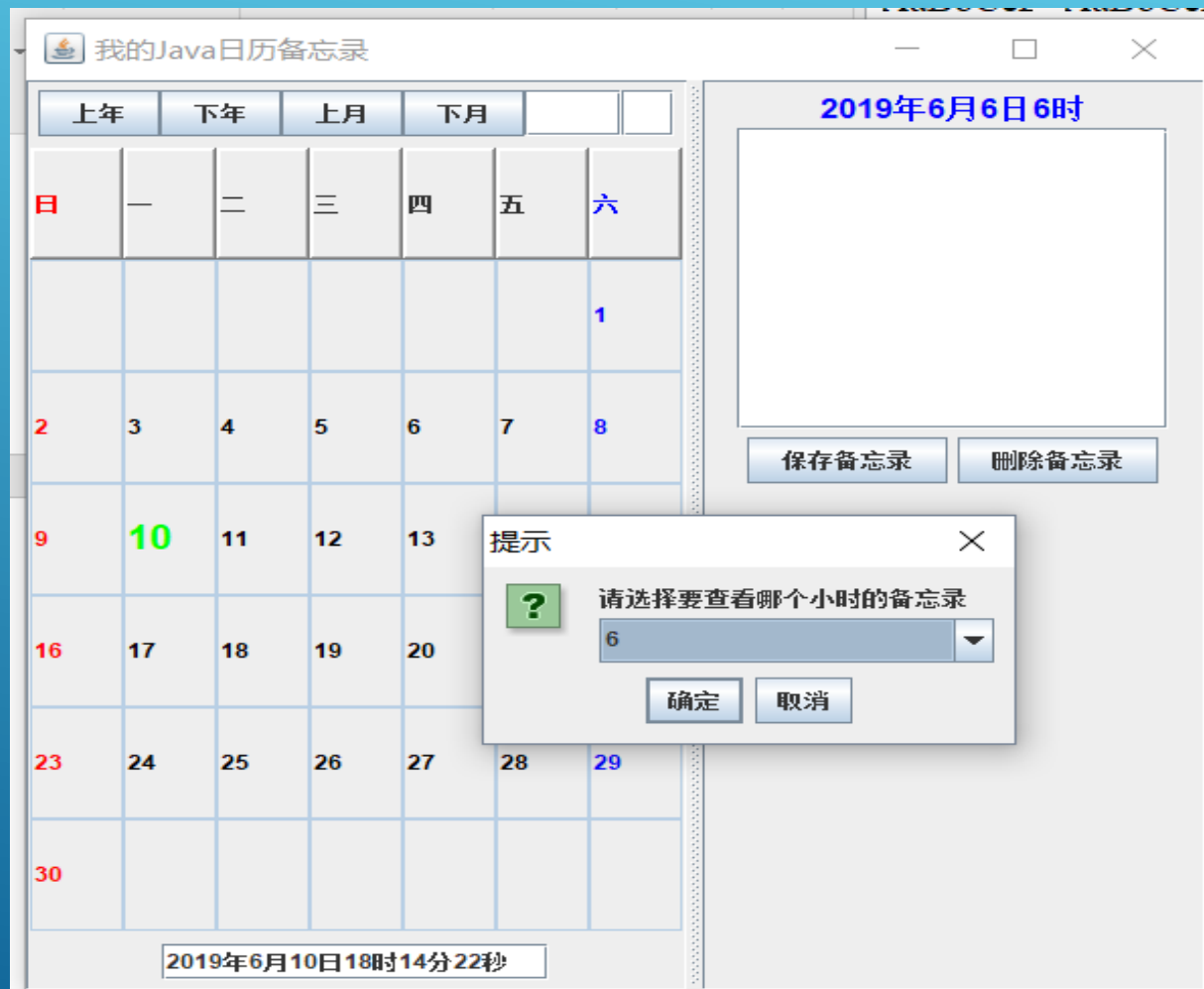


# 存储备忘录

```
String op=JOptionPane.showInputDialog(this,"请选择要保存备忘录于哪个小时","提示",JOptionPane.QUES  
int selected_hour=Integer.parseInt(op);
```

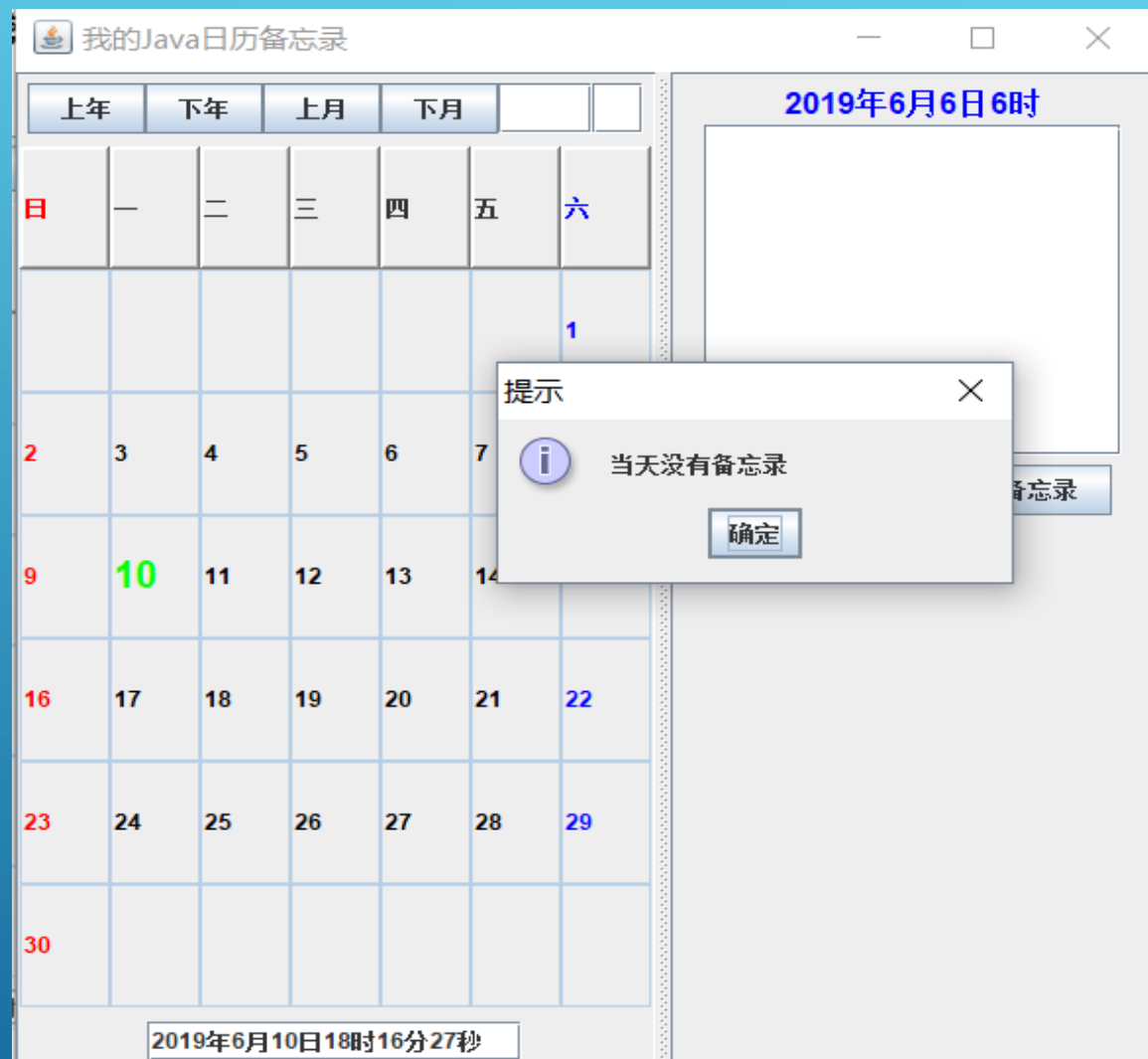
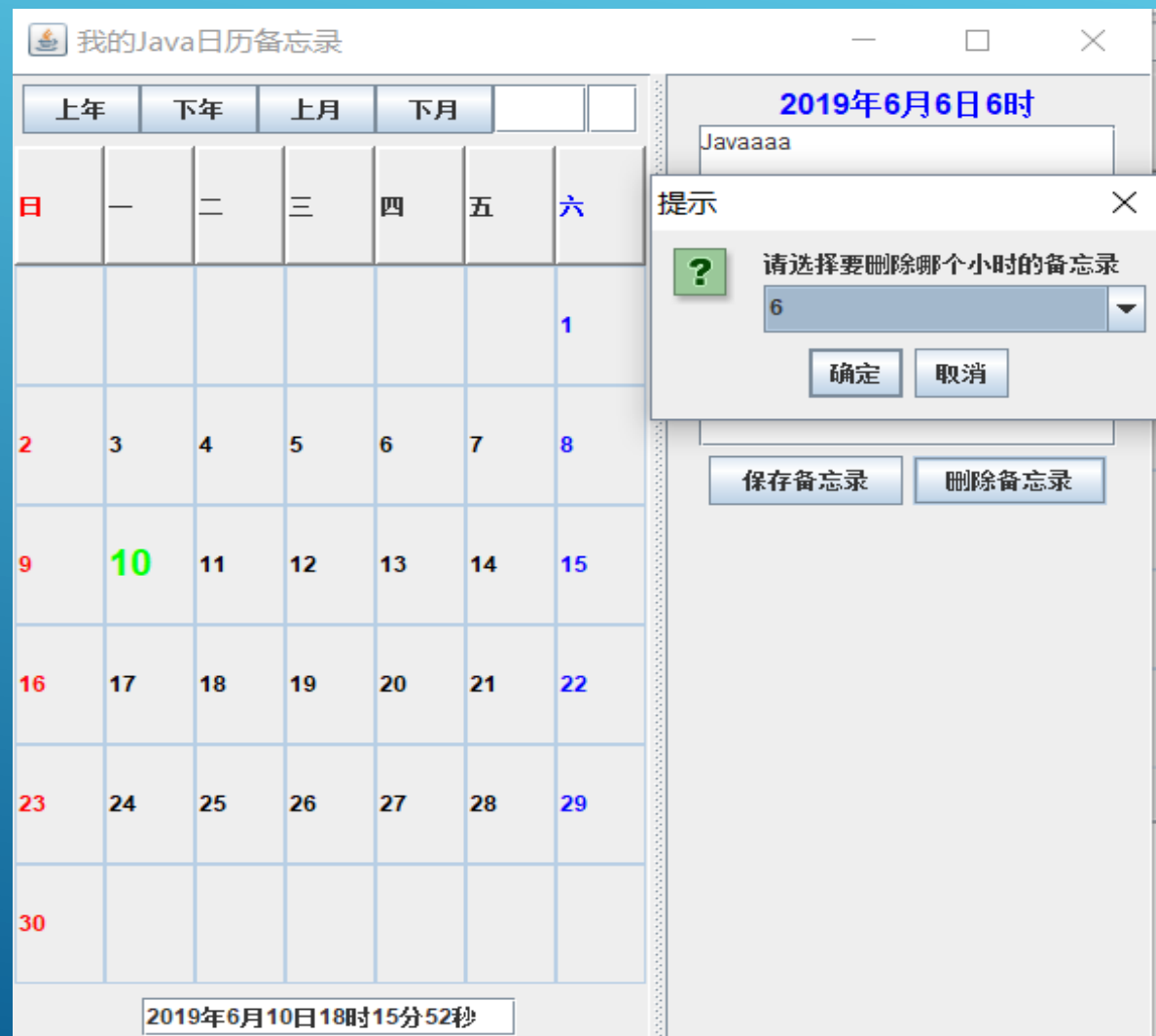


# 查看某一天某一小时的备忘录:





# 删除备忘录:





# 备忘录提醒:

