# 程序结构

## 目录结构

**AskMeWeather**

**┗--bin**

**┗--com**

**┗--janborn**

**┗--www**

**┗--GetNlp.class**

**┗--GetURL.class**

**┗--GetWeather.class**

**┗--Handler.class**

**┗--MainWindows.class**

**┗--Nlp.class**

**┗--Nlp$Items.class**

**┗--Nlp$aipNoConnectionException.class**

**┗--User.class**

**┗--Weather.class**

**┗--Weather$Daily.class**

**┗--Weather$DailyContent.class**

**┗--Weather$ErrorMessage.class**

**┗--Weather$ErrorMessageException.class**

**┗--Weather$LifeSuggestion.class**

**┗--Weather$Location.class**

**┗--Weather$Now.class**

**┗--WindowTool.class**

**┗--WriteTimer.class**

**┗--org**

**┗--json**

**┗--src**

**┗--com**

**┗--janborn**

**┗--www**

**┗--Communicate.java**

**┗--Demo.java**

**┗--GetNlp.java**

**┗--GetUrl.java**

**┗--GetWeather.java**

**┗--Handler.java**

**┗--Location.java**

**┗--MainWindows.java**

**┗--Nlp.java**

**┗--User.java**

**┗--Weather.java**

**┗--WindowTool.java**

**┗--WriteTimer.java**

**┗--org**

**┗--json**

## 类

**┗--Communicate.java**

**┗--Demo.class**

**┗--GetNlp.class**

**┗--GetURL.class**

**┗--GetWeather.class**

**┗--Handler.class**

**┗--MainWindows.class**

**┗--Nlp.class**

**┗--Nlp$Items.class**

**┗--Nlp$aipNoConnectionException.class**

**┗--User.class**

**┗--Weather.class**

**┗--Weather$Daily.class**

**┗--Weather$DailyContent.class**

**┗--Weather$ErrorMessage.class**

**┗--Weather$ErrorMessageException.class**

**┗--Weather$LifeSuggestion.class**

**┗--Weather$Location.class**

**┗--Weather$Now.class**

**┗--WindowTool.class**

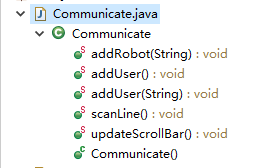
**┗--WriteTimer.class**

位于\bin\com\janborn\www\目录下。

### Communicate.class

用户和程序进行交互的类。获取用户的输入信息或选项信息，将其记录并显示在聊天面板上。

#### 成员列表



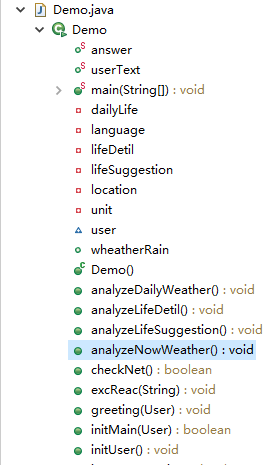
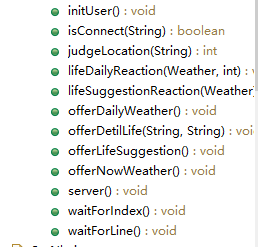
* 方法
  + Communicate()：无参构造函数
  + addRobot(String w)：接收机器人说的话，最后将机器人要说的话添加到聊天界面中
  + addUser(String w)：接收用户的选择，最后将用户的选择作为句子添加到聊天界面中
  + addUser()：将用户输入的话添加到聊天界面中
  + updateScrollBar()：将滚动条滑到最底端
  + scanLine()：读取输入框中用户的输入信息

### Demo.class

#### 功能

主调类。

#### 成员列表

* 属性
  + location ：地点
  + language：语言
  + unit：气温单位
  + dailyLife[]：用于逐日天气查询的时间
  + lifeSuggestion[]：用于存储生活指数需求的数组
  + lifeDetil[]：用于存储天气详细信息需求的数组
  + userText：用于记录输入信息
  + answer：回复信息
  + wheatherRain：记录是否已经下雨，以纠正降雨率
  + user：记录用户数据
* 方法
* Demo()：构造器，用于初始化用户
* initUser()：用户初始化模块
* initMain(User user)：应用初始化模块：1.打招呼、2.询问名字、3.判断网络连接是否正常：保证网络正常后再返回。
* checkNet()：检查网络。如果有网络，返回true；否则，询问用户是否还要继续查询。如果继续，那么机器人在有网络时自动唤醒；否则直接退出程序
* isConnect(String net)：检查网络连接。使用IO流打开网页连接。如果能打开说明网络连接正常，返回true；如果抛出异常，则网络连接失败，返回false
* greeting(User user)：和用户打招呼。首先送上一句时间问候；问好、自我介绍；登记用户信息。另外，用户信息将序列化后存在用户目录下。
* waitForLine()：等待读入用户输入信息。用于读入输入框中用户的输入信息。直到读到用户的输入信息或者触发事件才会退出该方法。
* waitForIndex()：等待读入用户按钮信息。用于读取按下的按钮信息。直到读到用户的按钮信息才会退出该方法。
* judgeLocation(String location)：判断目标位置是否支持查询。
  + - 1. 用于检查目标位置是否支持查询，目标位置通过参数传入。
      2. 通过在官方提供的地点Excel表中检索匹配，来判断目标地点是否支持。返回值：

1表示该地点支持，且原始字串以“省”字结尾；

2表示该地点支持，且已在原始字串后加上了“省”字；

-1表示该地点不支持

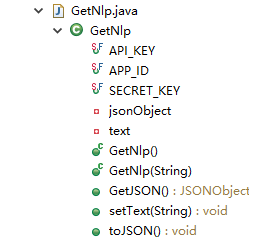
* analyzeNowWeather()：分析提取查询参数（实况天气）
* analyzeDailyWeather()：分析提取查询参数（逐日天气）
* analyzeLifeSuggestion()：分析提取查询参数（生活指数）
* analyzeLifeDetil()：分析提取查询参数（用户详情）；分析并设置用户想要查找的项目
* offerDetilLife(String start,String end)：提供服务（详细天气信息）；设置用户想要查找的项目
* offerNowWeather()：提供服务（实况天气）
* lifeDailyReaction(Weather askWeather,int day)：参数askWeather：天气信息查询结果；参数day：要查询的日子；将逐日天气的查询结果回应给用户
* offerDailyWeather()：提供服务（逐日天气）
* excReac(String e)：参数e：错误代码；用于处理心知天气错误代码的方法
* lifeSuggestionReaction(Weather askLifeSuggestion)：参数askLifeSuggestion：天气查询信息；将生活指数的查询结果回复给用户
* offerLifeSuggestion()：提供服务（生活指数）
* server()：主服务，进入服务分支的窗口。
* main(String[] args)：主调方法

### GetNlp.class

#### 功能

分析词法，获得JSONObject对象。

#### 成员列表



* 属性
  + APP\_ID：APP\_ID
  + API\_KEY：API\_KEY
  + SECRET\_KEY：SECRET\_KEY
  + text：用户输入的句子原文
  + jsonObject：得到的JSONObject对象
* 方法
* setText()：设置text属性
  + GetNlp()：无参构造函数
  + toJSON()：将得到的JSONObject对象存入jsonObject属性。无返回值。
  + GetNlp(String t)：有参构造函数，字符串t接收传进来的文本，同时给属性text赋值；然后把text文本，转成JSONObject格式，用jsonObject保存。无返回值。
  + GetJSON()：得到文本的JSONObject格式。返回JSONObject类型。

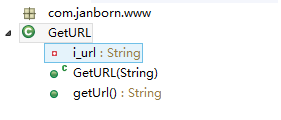
### GetURL.class

#### 功能

从URL获取满足JSON格式的String串。

**注：需要引入包org.json下的类。**

#### 成员列表



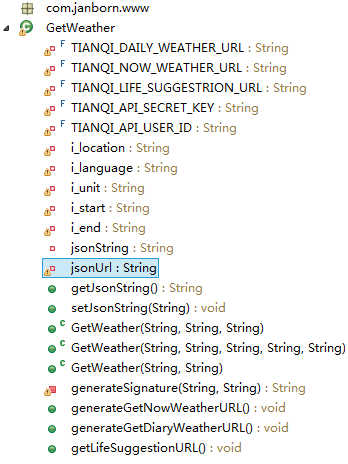
* 属性
  + i\_url：原始URL地址
* 方法
  + GetURL(String url)：构造器
  + getUrl()：从URL得到String数据存入临时变量JsonStr。返回值为JsonStr，类型为String。
* 异常抛出
  + IOException:网络异常,**请在主调函数进行处理**。

### GetWeather.class

#### 功能

向心知天气天气接口请求数据，获得URL地址。

#### 成员列表



* 属性
  + TIANQI\_DAILY\_WEATHER\_URL：逐日天气请求地址
  + TIANQI\_NOW\_WEATHER\_URL：实况天气请求地址
  + TIANQI\_LIFE\_SUGGESTRION\_URL：生活指数请求地址
  + TIANQI\_API\_SECRET\_KEY：secret key。
  + TIANQI\_API\_USER\_ID：user id。
  + i\_location：请求参数location。
  + i\_language：请求参数language。
  + i\_unit：请求参数unit。
  + i\_start：请求参数start。
  + i\_end：请求参数end。
  + jsonString：从jsonUrl获取的Json字符串。
  + jsonUrl：请求相应得到的URL。
* 方法
  + setJsonString()：数值属性jsonString。
  + GetWeather(String location, String language, String unit)：适用于获取实况天气的构造器。
  + GetWeather(String location, String start, String end, String language, String unit)：适用于逐日天气的构造器。
  + GetWeather(String location, String language)：适用于生活指数的构造器。
  + generateGetNowWeatherURL()：获取实况天气的URL。返回值类型为空。
  + generateGetDiaryWeatherURL()：获取逐日天气URL。返回值类型为空。
  + getLifeSuggestionURL()：获取生活指数URL。返回值类型为空。

**注**：对于三个GetWeather构造器，参数**language**和**unit**可省略。

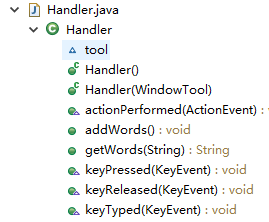
**注意**，如果缺少这两个参数，请传**空值（即“”）**

### Handler.class

#### 功能

用于监听键盘和鼠标事件的ActionListener,KeyListener接口实现类

#### 成员列表



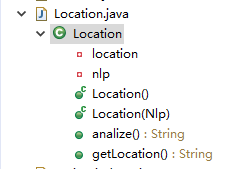
* 属性：
* tool : 用于访问窗口工具类的接口
* 方法：
* Handler()：构造器。
* Handler(WindowTool t)：带参的构造器，请使用这个。
* addWords()：将相关信息添加到聊天窗口。
* keyTyped(KeyEvent e)：监听键盘事件。
* keyPressed(KeyEvent e)：监听键盘按下事件。检查是否同时按下“ALT+ENTER”，若按下组合键，则表明用户输入完毕，将用户输入信息添加到聊天窗口。若按下“ENTER”键，则给出提示。
* keyReleased(KeyEvent e)：监听键盘事件。
* actionPerformed(ActionEvent e)：监听鼠标点击事件。当鼠标按下发送键，则读取用户输入信息，添加到聊天窗口中。
* componentResized(ComponentEvent e)：监听窗口尺寸变化事件。若窗口尺寸变化，则重新绘制输入框以适应窗口。
* getWords(String w)：获取信息。

### Location.class

#### 功能

获得地理位置

#### 成员列表



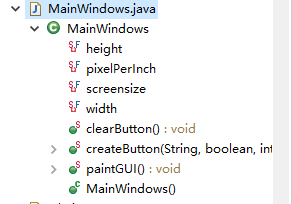
* 属性：
* Location：地理位置
* Nlp：Nlp的对象，用来获取地理位置
* 方法：
* Location()：无参构造函数
* getLocation()：获得词汇字符串location
* analize()：分析地理位置，赋值给字符串location；返回一个字符串类型的location值
* Location(Nlp nlp)：有参构造函数：设置文本内容text和把通过父类得到GetJSON格式为i\_item数组赋值；参数nlp是Nlp的对象,赋值给属性nlp

### MainWindows.class

#### 功能

用于绘制主窗口，并封装GUI相关的操作

#### 成员列表



* 属性
* Screensize: 获取分辨率
* Width: 分辨率的宽
* Height：分辨率的高
* 方法：
* MainWindows()：无参构造函数
* createButton(String w,boolean f,int index)：f：是否还要添加按钮 index:按钮编号
* clearButton()：清除按钮
* paintGUI()：绘制主窗口

### Nlp$Items$Loc\_details.class

Nlp类下的Items类下的嵌套类。

### Nlp$Items.class

Nlp类下的嵌套类。

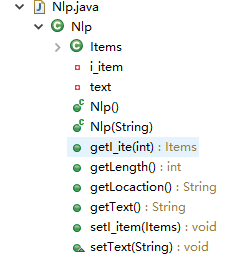
对应返回结果的“items”项，为对象数组。

### Nlp.class

#### 功能

通过继承父类GetNlp，分析jsonObject对象的格式，得到有效的地理位置信息。

#### 成员列表



**此类下的成员皆基于查询结果Json代码。**

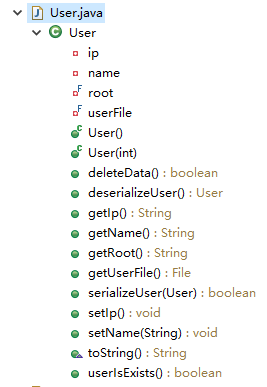
* 属性：如上图。
* 方法
  + getLength()：把文本转成JSONObject格式后，得到items数组的长度。返回一个int型的整数。
  + getLocaction()：得到地理位置信息。通过分析items，把词性是“LOC”的词保存到字符常量s中，直到遍历完整个数组，把字符串s返回出去。返回一个字符串。

### User.class

#### 功能

保存用户信息

#### 成员列表



* 属性
* name：默认用户名
* ip：用户IP
* root：序列化文件保存路径
* userFile：序列化文件
* 方法
* getUserFile()：返回序列化文件
* getRoot()：返回序列化文件保存路径
* toString()：用户信息的保存格式
* userIsExists()：判断该用户是否存在
* deserializeUser()：反序列化
* serializeUser(User u)：序列化
* User()：无参构造函数
* User(int i)：用于实例化临时User对象，都用一个会死循环
* deleteData()：删除用户信息
* getName()：返回用户名字
* setName(String name)：设置用户名字
* getIp()：返回用户IP
* setIp()：设置用户IP

### Weather$Daily.class

Weather类下的嵌套类。获取逐日天气所使用的类。

### Weather$DailyContent.class

Weather类下的嵌套类。

嵌套类Daily的模型。

**注：此类要和Daily类区别。Daily类与逐日天气返回数据第二项对应——返回数据第二项为JSON数组，应转为DailyContent类的List。**

### Weather$Location.class

Weather类下的嵌套类。获取地点信息所用的类。

对应所有天气请求返回结果的第一项。

### Weather$LifeSuggestion.class

Weather 类下的嵌套类。获取生活信息所用的类。

### Weather$ EerrorMessage.class

Weather 类下的嵌套类。获取错误信息。

### Weather$Now.class

Weather类下的嵌套类。获取实况天气所使用的类。

### Weather$ ErrorMessageException.class

Weather 类下的嵌套类。自定义异常类，用于抛出错误信息异常。

### Weather.class

#### 功能

继承自GetWeather类。

负责将获取到的JSON字符串转化为JSONObject对象，分析并返回属性值。

**注：同4.3.8。**

#### 成员列表

**此类下的成员皆基于查询结果Json代码。**

要求同4.3.8。

* 属性
  + Weather：存储获取的天气数据
  + location2：保存位置信息
  + lifeSuggestion：保存生活情况信息
  + daily：保存逐日天气信息
  + now：保存及时天气信息
  + errorManager：保存错误提示信息

**注：主调方法可以通过以上属性获取相应的信息。**

* + - 方法
      * public Weather(String location, String language, String unit)：获取及时天气信息
      * public Weather(String location, String start, String end, String language, String unit)：获取逐日天气信息
      * public Weather(String location, String language)：获取生活信息

**注：构造weather对象是根据输入的参数会出现不同的结果,其中language与unit如果没有输入,请输入” ”。**

**例weather(“石家庄”,” ” ,” ”),若需要通过IP确认归属地,请在构造时将location设置为IP。会抛出ErrorMessageException，抛出后说明发生了错误，请进行处理。**

* + - * private void setLocation()：用于设置location信息
      * SetErrorMessage：设置错误信息;

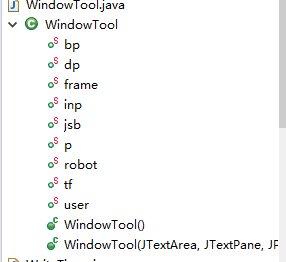
**注**：将所有内部类的**set()**方法**私有化**，禁止外部访问。

### WindowTool.class

#### 功能

窗口工具类。方便更新窗口内元件的类。

#### 成员列表



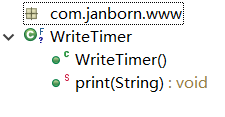
* + - 属性：
* tf：输入框
* p：上面板
* dp：下面板
* bp：按钮面板
* inp：输入面板
* jsb：滚动条
* frame：窗口框架
  + - 方法：
* WindowTool()：无参构造函数
* WindowTool(JTextArea tf,JTextPane p1,JPanel p2,JPanel p4,JPanel p3,JFrame f,JScrollBar jsb)：设置窗口工具类

### WriteTimer.class

#### 功能

**Final类**，不允许被继承。用于定时打印机器人的说话内容。

#### 成员列表



* + - 方法
* public static void print(String words)：**静态**方法，允许在实例化的情况下调用它进行打印。

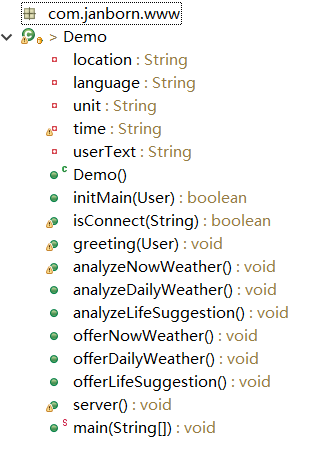
该函数模拟的是机器人思考并说话的过程。说话延时时长t∈[1,3]，单位为秒。

### Demo.class

#### 功能

主调类。

#### 成员列表



* 属性
* private String location：位置，装填用于查询天气的参数。默认值为“ip”。
* private String language：语言，装填用于查询天气的参数。默认值为“zh\_Han”。
* private String unit：单位，装填用于查询天气的参数。默认值为“c”
* private String time：时间长度，装填用于查询天气的参数。只用于查询逐日天气。
* private String userText：用户的输入文本。
* 方法
  + public Demo()：构造器。
  + public boolean initMain(User user)：初始化方法。应用初始化模块

1.打招呼2.询问名字3.判断网络连接是否正常

* + public boolean isConnect(String net)：检查网络连接。
  + public void greeting(User user)：用于和用户打招呼。
  + public void analyzeNowWeather()：用于提取查询实况天气需要的参数。
  + public void analyzeDailyWeather()：用于提取查询逐日天气需要的参数。
  + public void analyzeLifeSuggestion()：用于提取查询生活指数需要的参数。
  + public void offerNowWeather()：提供实况天气相关的服务。
  + public void offerDailyWeather()：提供逐日天气相关的服务。
  + public void offerLifeSuggestion()：提供生活指数相关的服务。
  + public void server() ：所有服务提供者。
  + public static void main(String[] args) throws IOException, ErrorMessageException：主调函数