**Python程序进阶课程报告**

[**李林锋**](mailto:2738289176@qq.com)

1. **需要安装的库**
2. pyqt5 2、sys 3、watchdog 4、matplotlib 5、pypinyin 6、request 7、schedule
3. **API接口**
4. 和风天气（[[和风天气开发服务 ~ 强大、丰富的天气数据服务 (qweather.com)](https://dev.qweather.com/)/](https://console.qweather.com/)）
5. **系统涉及架构**
6. **用户界面**
7. 主界面，主界面包含上方的预警按钮，说明按钮，点击图标可以看到相应的子窗口。下面会有介绍。紧接着从上往下依次是此时的温度，空气质量，湿度，风速，气压，体感温度，露点，以及当日运动建议。下面则是当前往后24h的具体数据情。接着是24h的温度趋势图，最下方是未来一周的天气情况，可直接进行预览。
8. 子窗口，子窗口含有三类。第一类，预警窗口，会显示当前本地是否存在警告信号，如果存在，点击后可看到详细情况；如果没有，则会显示对应文字。第二类，说明窗口，本窗口是用于说明本程序的要求，分别是输入城市准确，点击需要等待响应，不必多次点击，服务不可用时原因在于免费的API接口达到访问总量。第三类，未来七日具体情况，上面包含当日的具体情况，可以点击查看。
9. **构造情况**
10. 分为主界面程序，子界面程序，绘制趋势图程序，API接口数据获取程序，获得城市名程序。其中主界面程序通过引用API数据接口，API数据接口从城市名获取窗口获得需要查询的城市名，API界面将获得的数据传入主界面显示。

（2）代码核心讲解：api\_num

1° 整点更新软件

if \_\_name\_\_ == "\_\_main\_\_":  
 # 每小时的整点触发更新  
 schedule.every().hour.at(":00").do(search\_city\_code)  
 while True:  
 schedule.run\_pending()  
 time.sleep(1)

此处为较关键的一步，这里将会每个整点更新数据，实现程序的自动更新.

2° 当服务不可用时反馈出来的窗口来自于此。

try:  
 code = search\_city\_code(hanzi\_to\_pinyin(text))  
except:  
 app = QApplication([])  
 msg\_box = QMessageBox()  
 msg\_box.setIcon(QMessageBox.Warning)  
 msg\_box.setText("当前服务不可用")  
 msg\_box.setWindowTitle("警告")  
 msg\_box.setStandardButtons(QMessageBox.Ok)  
 msg\_box.exec\_()  
 app.quit()

3°代码核心讲解：curve\_draw 绘制图片并保存以供引用。

def showResult(xList, yList, xLabel):  
 plt.figure(figsize=(7.91, 1.91), dpi=100) # 设置图形大小和分辨率  
 plt.ylim( max(yList) + 1, min(yList) - 1)  
 plt.plot(xList, yList, '\*-') # 绘制曲线  
 plt.fill\_between(xList, yList, color='skyblue', alpha=1) # 填充曲线下方的颜色  
 plt.xlabel(xLabel) # 设置x轴标签  
 for i in range(len(xList)):  
 plt.text(xList[i], yList[i], str(yList[i]), ha='center', va='bottom', fontsize=10.5) # 在数据点上标注数值  
 plt.yticks([]) # 隐藏y轴刻度  
 plt.gca().invert\_yaxis()  
 plt.savefig('temperature.png') # 保存图片

4°代码核心讲解：weather\_forecast

def convert\_weather\_list(weather\_list):  
 new\_weather\_list = []  
 for weather in weather\_list:  
 if '晴' in weather:  
 new\_weather\_list.append('晴天')  
 elif '云' in weather:  
 new\_weather\_list.append('多云')  
 elif '阴' in weather:  
 new\_weather\_list.append('阴天')  
 elif '雨' in weather:  
 new\_weather\_list.append('暴雨')  
 elif '雾' in weather:  
 new\_weather\_list.append('雾')  
 elif '霾' in weather:  
 new\_weather\_list.append('天气-沙尘')  
 elif '扬沙' in weather:  
 new\_weather\_list.append('特大扬沙')  
 elif '浮尘' in weather:  
 new\_weather\_list.append('特大扬沙')  
 elif '暴' in weather:  
 new\_weather\_list.append('特大扬沙')  
 else:  
 new\_weather\_list.append(weather)  
 return new\_weather\_list

这个函数的作用是在于将获得的天气情况映射到图标名字。

5°核心代码讲解：weather\_forecast，主界面的绘制

class Ui\_MainWindow(QMainWindow):

6°核心代码讲解：weather\_forecast，主界面的数据更新，当api数据改变以后更新,使用watchdog监视。

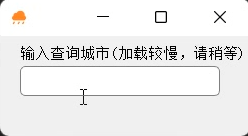
path = "api\_num.py" # 监视的文件  
event\_handler = FileModifiedHandler()  
observer = Observer()  
observer.schedule(event\_handler, path, recursive=True)  
observer.start()  
try:  
 while True:  
 time.sleep(1)  
except KeyboardInterrupt:  
 observer.stop()  
observer.join()

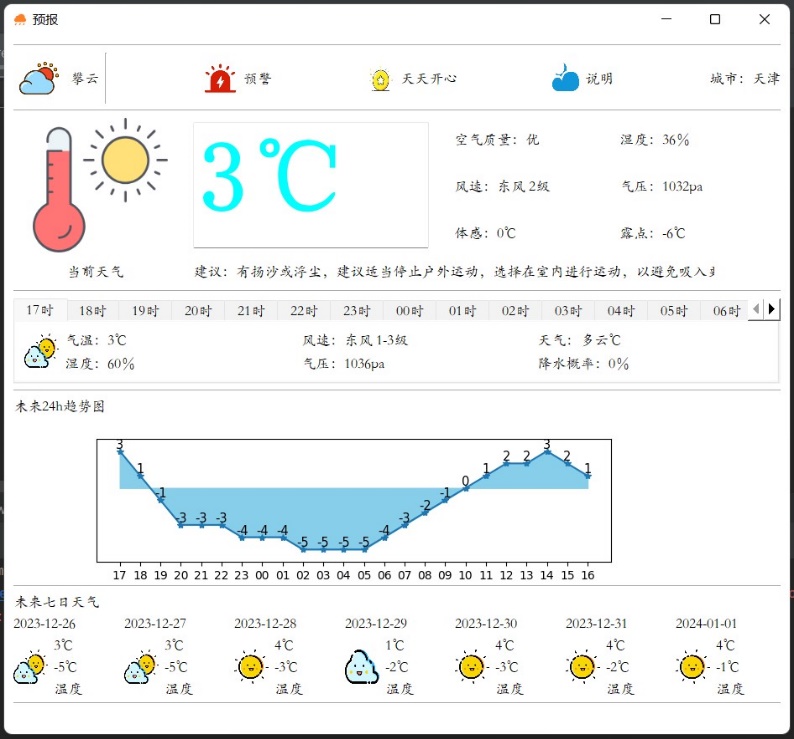
7°核心代码讲解：img\_rc.py 文件是图标图片的二进制文件

1. **交互情况**

**运行weather\_forecast.py文件**

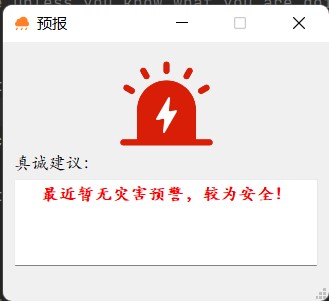
1° 初始界面



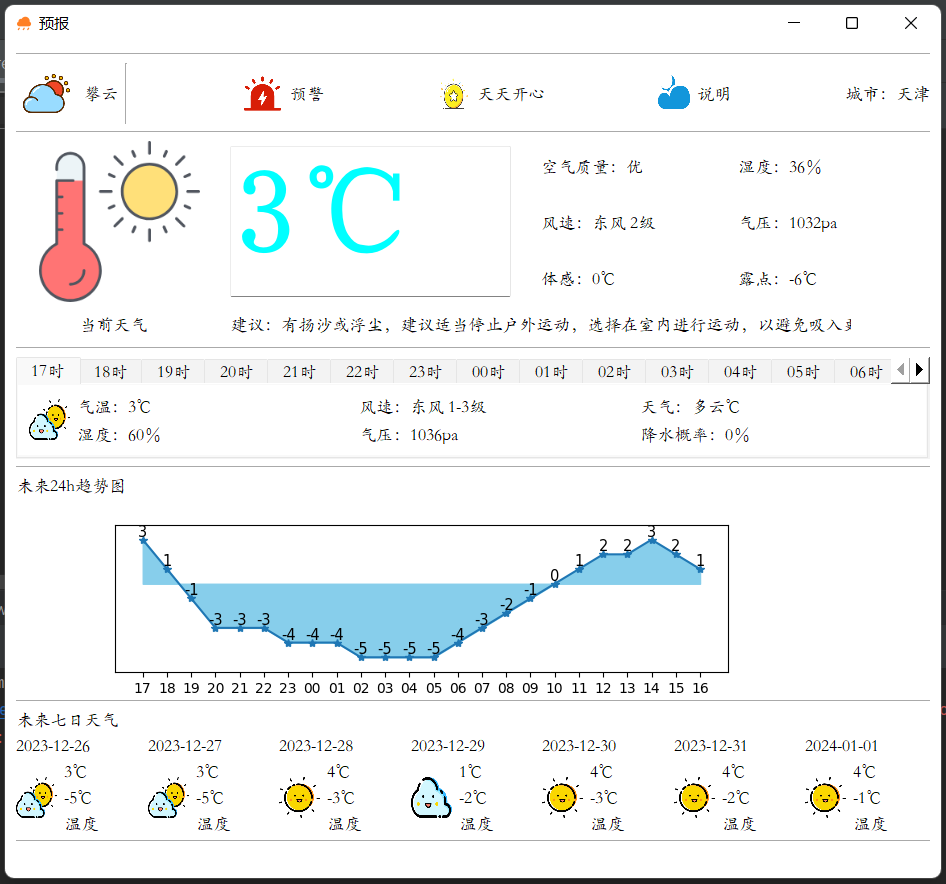
2° 在此输入城市名字后耐心等待，会出现主界面。

3° 如果服务暂不可用或者城市名输入有误会出现下列情况

4° 出现主界面以后，点击预警查看警告信息（选择的地方无预警通知）



5° 点击说明图标查看作者说明

6° 点击24h时间点查看情况

7° 点击未来七日情况查看情况（首次点击需要稍等片刻，加载较慢）。

8° 窗口中的控件大小会根据窗口大小改变，详情查看视频