

|  |  |
| --- | --- |
| 成 绩 |  |
| 批阅教师 |  |
| 日 期 |  |

**桂林电子科技大学**

**实训报告**

**2020-2021学年第1学期**

**学 院 计算机工程学院**

**课 程 JAVA程序设计实训**

**学 号 181603010819**

**姓 名 赵 开 强**

**指导老师 潘瑞远**

**日 期 2020.11.18**

# 实验目的

学会运用整个学期学到的所有JAVA编程知识，完成一个具有友好界面具和一定技术含量的JAVA应用程序的设计和实现的工作。

# 实验环境

第4教学楼4316机房。

# 实验同组成员：

无

# 实验内容

**1. 软件描述**

**这是一款拥有查看天气功能的天气预报软件可以根据输入的城市查询未来三天的天气情况添加了图片显示改进了界面布局使得更加简单易懂**

**2. 所有类的命名与功能表述：**

**比如：**

**Main类 ： 程序开始类**

**model包 ：根据天气预报接口返回生成的实体类**

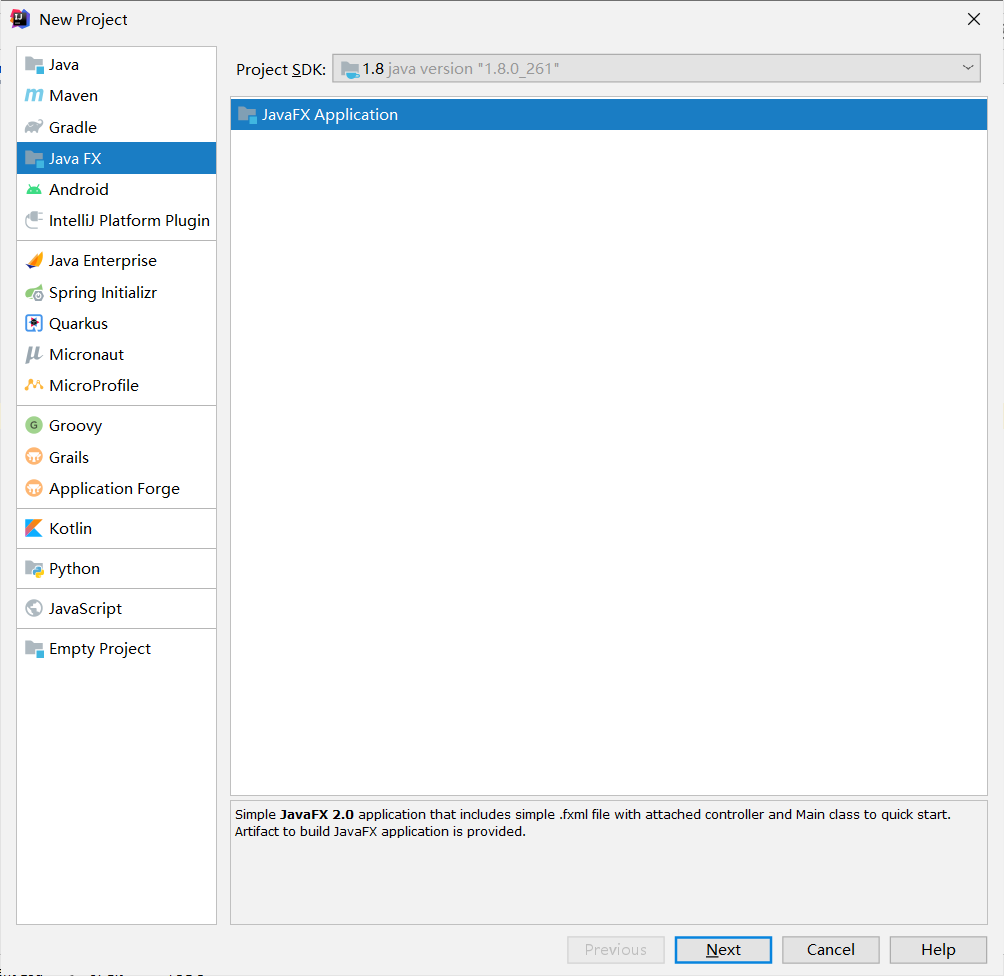
**HomeController类 ：程序实现的主要实现类**

# 实验步骤

（要求按实验内容写出实验步骤，图文并茂）

**1.实验步骤 和 源代码**

**（1）新建一个Javafx项目**

****

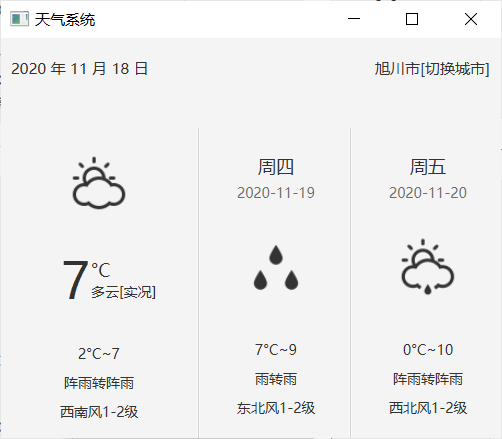
**（2）新建HomeController类及其对应的home.fxml布局文件**

**(3)不断修改控制类和界面最终得到代码**

**package** weather;  
  
**import** com.google.gson.Gson;  
**import** javafx.application.Platform;  
**import** javafx.beans.value.ChangeListener;  
**import** javafx.beans.value.ObservableValue;  
**import** javafx.collections.FXCollections;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.ComboBox;  
**import** javafx.scene.control.Label;  
**import** javafx.scene.image.Image;  
**import** javafx.scene.image.ImageView;  
**import** javafx.scene.input.MouseEvent;  
**import** javafx.scene.layout.HBox;  
**import** javafx.scene.layout.Pane;  
**import** model.NowWeatherModel;  
**import** model.SeachCityModel;  
**import** model.ThreeDayWeatherModel;  
**import** okhttp3.\*;  
  
**import** javax.swing.\*;  
**import** java.io.IOException;  
**import** java.time.LocalDate;  
**import** java.time.format.DateTimeFormatter;  
**import** java.util.ArrayList;  
**import** java.util.Calendar;  
**import** java.util.Date;  
**import** java.util.List;  
  
  
**public class** HomeController ***{* public** Label **top\_date**, **top\_city**; *//顶部的当前日期和城市* **public** ImageView **img\_one**, **img\_two**, **img\_three**; *//天气图* **public** Label **mon\_two**, **mon\_three**; *//周几* **public** Button **bt\_city**, **bt\_ok**; *//搜索按钮* **public** ComboBox **select\_city**;  
 **public** Label **one\_temp**, **two\_temp**, **three\_temp**; *//温度* **public** Label **one\_wea**, **two\_wea**, **three\_wea**; *//天气* **public** Label **one\_wind**, **two\_wind**, **three\_wind**; *//风速* **public** Label **date\_two**, **date\_three**; *// 日期* **public** Label **te\_current**, **te\_current\_text**; *// 实况温度 实况描述* **public** HBox **top\_info**;  
 **public** Pane **top\_input**;  
  
 SeachCityModel **cityData**; *// api获取的搜索城市数据* NowWeatherModel **nowWeatherList**; *// 实况数据* **final static** String ***KEY*** = **"5e47b2143db54d4d8279c7c853b5f0b0"**;  
  
 String **location** = **"北海"**;  
  
 *// 城市信息搜索* **public** String **citySeachInfoUrl** = **"https://geoapi.qweather.com/v2/city/lookup?key="** + ***KEY*** + **"&location="**;  
 *// 实况天气url* **public** String **getlocalWeatherUrl** = **"https://devapi.qweather.com/v7/weather/now?key=5e47b2143db54d4d8279c7c853b5f0b0&location="**;  
 *// 3天气预报* **public** String **getWeatherReportUrl** = **"https://devapi.qweather.com/v7/weather/3d?key=5e47b2143db54d4d8279c7c853b5f0b0&location="**;  
  
  
 */\*\*  
 \* 根据当前时间更新左上角日期  
 \*/* **public void** updateCurrentDate***() {*** DateTimeFormatter formatter = DateTimeFormatter.*ofPattern****(*"yyyy 年 MM 月 dd 日"*)***;  
 String format = LocalDate.*now****()***.format***(***formatter***)***;  
 System.***out***.println***(***format***)***;  
 **top\_date**.setText***(***format***)***;  
 ***}*** */\*\*  
 \* 初始化页面  
 \*/* **public void** initPage***() {*** initWeekDays***()***;  
 *// 下拉框点击事件监听* **select\_city**.getSelectionModel***()***.selectedIndexProperty***()***.addListener***(*new** ChangeListener***<***Number***>() {*** @Override  
 **public void** changed***(***ObservableValue***<***? **extends** Number***>*** observable, Number oldValue, Number newValue***) {  
  
  
 }  
 })***;  
 */\*OkHttpClient client = new OkHttpClient();  
 Request request = new Request.Builder()  
 .url(citySeachInfoUrl + "北海")  
 .build();  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 }  
  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 System.out.println(response.code());  
 System.out.println(response.body().string());  
 }  
 });\*/* ***}*** */\*\*  
 \* 初始化周数  
 \*/* **public void** initWeekDays***() {*** String***[]*** weekDays = ***{*"周日"**, **"周一"**, **"周二"**, **"周三"**, **"周四"**, **"周五"**, **"周六"*}***;  
 **int** two, three, four, five;  
 Calendar calendar = Calendar.*getInstance****()***;  
 calendar.setTime***(*new** Date***())***;  
 **int** w = calendar.get***(***Calendar.***DAY\_OF\_WEEK)*** - 1;  
 **if *(***w < 0***) {*** w = 0;  
 ***}*** two = w + 1;  
 three = w + 2;  
 four = w + 3;  
 five = w + 4;  
 **if *(***two > 6***)*** two -= 7;  
 **if *(***three > 6***)*** three -= 7;  
 **if *(***four > 6***)*** four -= 7;  
 **if *(***five > 6***)*** five -= 7;  
  
*// weekDays[w] 周几* **mon\_two**.setText***(***weekDays***[***two***])***;  
 **mon\_three**.setText***(***weekDays***[***three***])***;  
 ***}*** */\*\*  
 \* 点击切换城市响应事件  
 \*/* **public void** click\_city***(***MouseEvent mouseEvent***) {*** changeInput***()***;  
  
 ***}*** */\*\*  
 \* 点击搜索按钮响应事件  
 \*  
 \** ***@param mouseEvent*** *\*/* **public void** bt\_click\_city***(***MouseEvent mouseEvent***) {*** *//获取下拉输入框内容* String in\_text = **select\_city**.getEditor***()***.getText***()***;  
 getCityDateByName***(***in\_text***)***;  
 ***}*** */\*\*  
 \* 根据输入获取查询城市列表  
 \*/* **public void** getCityDateByName***(***String in\_text***) {*** *//如果输入框为空，弹出提示框* **if *(***in\_text.length***()*** < 1***) {*** JOptionPane.*showMessageDialog****(*null**, **"请输入城市再点击搜索"*)***;  
 **return**;  
 ***}*** OkHttpClient client = **new** OkHttpClient***()***;  
 Request request = **new** Request.Builder***()*** .url***(*citySeachInfoUrl** + in\_text***)*** .build***()***;  
 client.newCall***(***request***)***.enqueue***(*new** Callback***() {*** @Override  
 **public void** onFailure***(***Call call, IOException e***) {*** e.printStackTrace***()***;  
 ***}*** @Override  
 **public void** onResponse***(***Call call, Response response***)* throws** IOException ***{* cityData** = **new** Gson***()***.fromJson***(***response.body***()***.string***()***, SeachCityModel.**class*)***;  
  
 Platform.*runLater****(*new** Runnable***() {*** @Override  
 **public void** run***() {*** *//更新下拉框* updateCombobox***(*cityData*)***;  
 ***}  
 })***;  
  
 ***}  
 })***;  
 ***}*** */\*\*  
 \* 根据搜索返回的城市数据 更新下拉框  
 \*  
 \** ***@param cityData*** *\*/* **public void** updateCombobox***(***SeachCityModel cityData***) {*** ArrayList***<***String***>*** cityNames = **new** ArrayList***<>()***;  
 **if *(***cityData.getCode***()***.equals***(*"200"*)) {* for *(***SeachCityModel.LocationBean city : cityData.getLocation***()) {*** cityNames.add***(***city.getCountry***()*** + **"--"** + city.getAdm1***()*** + **"--"** + city.getAdm2***()*** + **"--"** + city.getName***())***;  
 ***}* select\_city**.setItems***(***FXCollections.*observableArrayList****(***cityNames***))***;  
 ***}* else *{* select\_city**.getEditor***()***.setText***(*"暂无搜索结果"*)***;  
  
*// select\_city.setItems(FXCollections.observableArrayList(""));* ***}  
 }*** */\*\*  
 \* 确定按钮点击事件  
 \*  
 \** ***@param mouseEvent*** *\*/* **public void** bt\_click\_ok***(***MouseEvent mouseEvent***) {*** *// 选择中的序号* **int** selectIndex = **select\_city**.getSelectionModel***()***.getSelectedIndex***()***;  
 *// 如果文本为空* **if *(*select\_city**.getEditor***()***.getText***()***.length***()*** == 0***) {*** JOptionPane.*showMessageDialog****(*null**, **"请输入内容后再进行此操作"*)***;  
 **return**;  
 ***}*** *// 如果没有选择，默认第一个* **if *(***selectIndex == -1***) {*** getCityDateByName***(*select\_city**.getEditor***()***.getText***())***;  
 selectIndex = 0;  
 ***}*** *// 根据下拉框选中的序号获取城市id* getNowWeatherDate***(*cityData**.getLocation***()***.get***(***selectIndex***)***.getId***())***;  
 getOtherWeather***(*cityData**.getLocation***()***.get***(***selectIndex***)***.getId***())***;  
  
 *// 根据选中的城市，获取其城市名 设置右上角名字标签* **top\_city**.setText***(*cityData**.getLocation***()***.get***(***selectIndex***)***.getName***()*** + **"[切换城市]"*)***; *//城市名* exitInput***()***;  
  
  
 ***}*** */\*\*  
 \* 更新界面  
 \*/* **public void** updateOtherPage***(***ThreeDayWeatherModel threeDayWeatherModel***) {*** List***<***ThreeDayWeatherModel.DailyBean***>*** list = threeDayWeatherModel.getDaily***()***;  
 **one\_temp**.setText***(***list.get***(***0***)***.getTempMin***()***+**"°C~"**+list.get***(***0***)***.getTempMax***())***;  
 **two\_temp**.setText***(***list.get***(***1***)***.getTempMin***()*** + **"°C~"** + list.get***(***1***)***.getTempMax***())***;  
 **three\_temp**.setText***(***list.get***(***2***)***.getTempMin***()*** + **"°C~"** + list.get***(***2***)***.getTempMax***())***;  
 **one\_wea**.setText***(***list.get***(***0***)***.getTextDay***()*** + **"转"** + list.get***(***0***)***.getTextNight***())***;  
 **two\_wea**.setText***(***list.get***(***1***)***.getTextDay***()*** + **"转"** + list.get***(***1***)***.getTextNight***())***;  
 **three\_wea**.setText***(***list.get***(***2***)***.getTextDay***()*** + **"转"** + list.get***(***2***)***.getTextNight***())***;  
 **one\_wind**.setText***(***list.get***(***0***)***.getWindDirDay***()*** + list.get***(***0***)***.getWindScaleDay***()*** + **"级"*)***;  
 **two\_wind**.setText***(***list.get***(***1***)***.getWindDirDay***()*** + list.get***(***1***)***.getWindScaleDay***()*** + **"级"*)***;  
 **three\_wind**.setText***(***list.get***(***2***)***.getWindDirDay***()*** + list.get***(***2***)***.getWindScaleDay***()*** + **"级"*)***;  
 **date\_two**.setText***(***list.get***(***1***)***.getFxDate***())***;  
 **date\_three**.setText***(***list.get***(***2***)***.getFxDate***())***;  
 **img\_two**.setImage***(*new** Image***(*"icon/"** + list.get***(***1***)***.getIconDay***()*** + **".png"*))***;  
 **img\_three**.setImage***(*new** Image***(*"icon/"** + list.get***(***2***)***.getIconDay***()*** + **".png"*))***;  
  
  
 ***}*** */\*\*  
 \* 根据id 获取实况天气数据 并调用实况天气更新方法  
 \*  
 \** ***@param locationId*** *\*/* **public void** getNowWeatherDate***(***String locationId***) {*** OkHttpClient client = **new** OkHttpClient***()***;  
 Request request = **new** Request.Builder***()*** .url***(*getlocalWeatherUrl** + locationId***)*** .build***()***;  
 client.newCall***(***request***)***.enqueue***(*new** Callback***() {*** @Override  
 **public void** onFailure***(***Call call, IOException e***) {*** e.printStackTrace***()***;  
 ***}*** @Override  
 **public void** onResponse***(***Call call, Response response***)* throws** IOException ***{*** String s = response.body***()***.string***()***;  
 **nowWeatherList** = **new** Gson***()***.fromJson***(***s, NowWeatherModel.**class*)***;  
 System.***out***.println***(***s***)***;  
  
 System.***out***.println***(*nowWeatherList**.getNow***()***.getIcon***())***;  
 System.***out***.println***(*"更新"*)***;  
  
 Platform.*runLater****(*new** Runnable***() {*** @Override  
 **public void** run***() {*** System.***out***.println***(*"更新ui"*)***;  
 updateNowPage***(*nowWeatherList*)***;  
  
 ***}  
 })***;  
  
 ***}  
 })***;  
 ***}*** */\*\*  
 \* 根据获取的现在天气的数据进行label赋值  
 \*  
 \** ***@param nowWeather*** *\*/* **public void** updateNowPage***(***NowWeatherModel nowWeather***) {*** System.***out***.println***(***nowWeather.toString***())***;  
 *//当前实况天气* **te\_current**.setText***(***nowWeather.getNow***()***.getTemp***())***;  
 *// 实况天气文本描述* **te\_current\_text**.setText***(***nowWeather.getNow***()***.getText***()*** + **"[实况]"*)***;  
 System.***out***.println***(***nowWeather.getNow***()***.getText***())***;  
  
 *//实况天气图片* String imgPath = **"icon/"** + nowWeather.getNow***()***.getIcon***()*** + **".png"**;  
 System.***out***.println***(***imgPath***)***;  
  
 **img\_one**.setImage***(*new** Image***(*"/icon/104.png"*))***;  
  
  
 ***}*** */\*\*  
 \* 获取三天天气预报  
 \*  
 \** ***@param locationId*** *传入locationID  
 \*/* **public void** getOtherWeather***(***String locationId***) {*** OkHttpClient client = **new** OkHttpClient***()***;  
 Request request = **new** Request.Builder***()*** .url***(*getWeatherReportUrl** + locationId***)*** .build***()***;  
 client.newCall***(***request***)***.enqueue***(*new** Callback***() {*** @Override  
 **public void** onFailure***(***Call call, IOException e***) {*** e.printStackTrace***()***;  
 ***}*** @Override  
 **public void** onResponse***(***Call call, Response response***)* throws** IOException ***{*** String string = response.body***()***.string***()***;  
 ThreeDayWeatherModel threeDayWeathers = **new** Gson***()***.fromJson***(***string, ThreeDayWeatherModel.**class*)***;  
 System.***out***.println***(*"-----------------"*)***;  
 System.***out***.println***(***string***)***;  
 Platform.*runLater****(*new** Runnable***() {*** @Override  
 **public void** run***() {*** updateOtherPage***(***threeDayWeathers***)***;  
  
  
 ***}  
 })***;  
 ***}  
 })***;  
 ***}*** */\*\*  
 \* 根据返回的天气描述匹配对应本地图片  
 \*  
 \** ***@param text*** *\** ***@return*** *\*/* **public** String matchIcon***(***String text***) {* switch *(***text***) {* case "晴"**:  
 **return "qing"**;  
 **case "霾"**:  
 **return "mai"**;  
 **case "小雨"**:  
 **return "xiaoyu"**;  
 **case "中雨"**:  
 **return "zhongyu"**;  
 **case "大雨"**:  
 **return "dayu"**;  
 **case "多云"**:  
 **return "duoyun"**;  
  
 ***}* return ""**;  
 ***}*** */\*\*  
 \* 切换到输入模式  
 \*/* **public void** changeInput***() {****// top\_city.setVisible(false);  
// bt\_ok.setVisible(true);  
// bt\_city.setVisible(true);  
// select\_city.setVisible(true);* **top\_info**.setVisible***(*false*)***;  
 **top\_input**.setVisible***(*true*)***;  
 ***}*** */\*\*  
 \* 退出输入模式  
 \*/* **public void** exitInput***() {****// top\_city.setVisible(true);  
// bt\_ok.setVisible(false);  
// bt\_city.setVisible(false);  
// select\_city.setVisible(false);* **top\_info**.setVisible***(*true*)***;  
 **top\_input**.setVisible***(*false*)***;  
  
 ***}  
  
  
}***

**（4）还有根据api生成NowWeatherModel、SeachCityModel、ThreeDatweatherModel类**

**2.实验运行效果截图**



# 实验总结（不少于100字）

通过本次java实验，让我进一步巩固了java的相关知识。Java不仅仅需要理论，编程能力也很重要。通过java实操能够很好的增强自己的编程能力。而通过独立自主完成项目，在不断设计程序，编写程序，发现问题，寻找解决方案、解决问题中能还能够获得很大的成就感和满足感。