**云南大学软件学院**

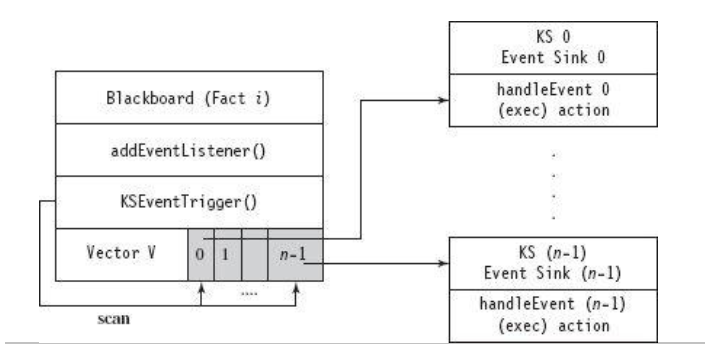
**软件设计与体系结构实验-实验报告**

课程： 软件设计与体系结构实验 任课教师： 包崇明 实验指导教师： 包崇明

序号\_05\_\_姓名： 孙启奥 学号： 20171120028 专业： 软件工程 日期： 成绩：

**实验四 黑板风格的气候预报系统设计**

1. **实验目的**
2. 理解黑板风格-组成构件（黑板、知识源、控制器）及实现组件间交互作用的连接件的实现方式。黑板存储数据，提供注册方法；知识源（多个）向黑板注册，当关注的数据发生变化时，进行处理，在黑板上更新数据；控制器：初始化黑板及知识源，确定问题解。
3. 理解发布/订阅方式实现黑板风格应用的基本方法和步骤。
4. 理解黑板风格的主要特点：多个知识源共同协作解决复杂问题，每个知识源具有特定的问题解决能力。
5. 掌握采用用面向对象程序设计语言（Java或者c++）设计黑板风格应用的基本方法及步骤：为黑板和知识源设计相应类，黑板类应提供注册方法，存放注册的知识源对象；知识源类能够接受黑板的通知，完成功能处理和更新黑板数据。
6. 进一步熟悉类图、时序图。
7. **实验步骤**
8. 采用Java或者C++设计一个黑板风格的气候预报系统。该系统的原理为：黑板存放天气预报基本要素和降雨，台风等气象预报信息；多个基本信息源（气候原始信息，如卫星云图，气压、气温、风速、风向、湿度等）；气象分析器（根据基本信息源信息通过某方法可得到降雨、台风等预报信息。至少两个：降雨预报器和台风预报器）；普通消费者，能够订阅需要的天气信息，如暴雨，台风等，并在订阅的预报发生后能获得通知，指向指定动作。
9. 根据上述气候预报系统的设计原理完成气候预报系统用例图表示的需求分析
10. 根据你的需求分析，画出该系统的设计类图（有能力的同学可实现系统）
11. 可参考如下图

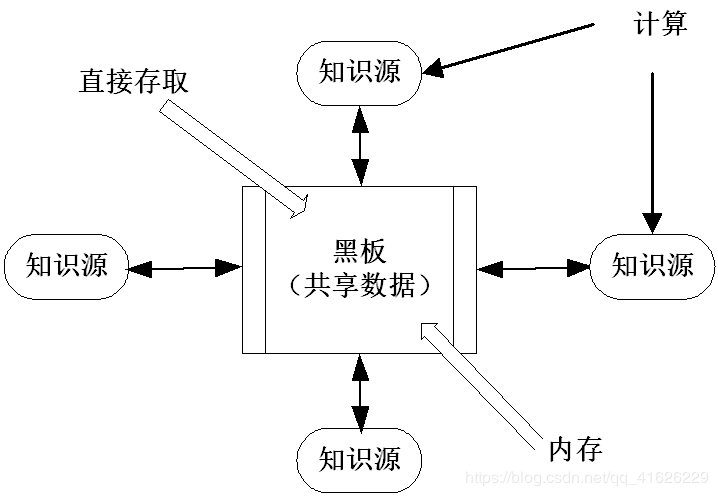


# 用例图及说明

1. **背景知识**

黑板模式中肯定会有一个黑板之类的对象，然后有一群类似老师的人会在黑板上记录下自己的想法或者对某件事情的看法，老师同时可以对黑板上的不正确的东西进行改进，从而不断更新黑板上的东西。从而概括一点就是：有一定数量的对象对统一中央数据结构进行操作，从而实现中央数据结构的持续更新。同时在这里面还会涉及到一个控制器。

比如A观察了B、C、D、E、F这么多个对象，按照观察者模式，当B、C、D、E、F中某个对象状态改变时，通过初始化一个A对象然后利用A对象去调用operation操作。但是在黑板模式中是这样，B、C、D、E、F一旦状态改变，它会将其记录在一个类似黑板的统一中央数据中，然后A对象只需从黑板上关注自己的观察对象状态是否发生改变，一旦有改变则调用operation()操作。这两种方法中发生了一个微妙的改变，似乎黑板模式更好的解决了观察者与被观察者之间的耦合、依赖性。



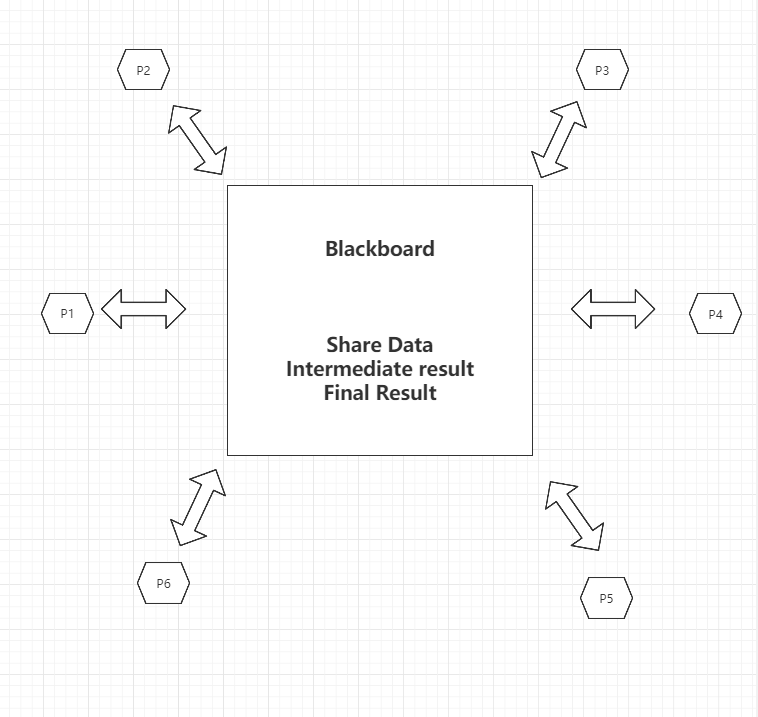
**黑板模式主要模块：**

1) 知识源：更新数据方法

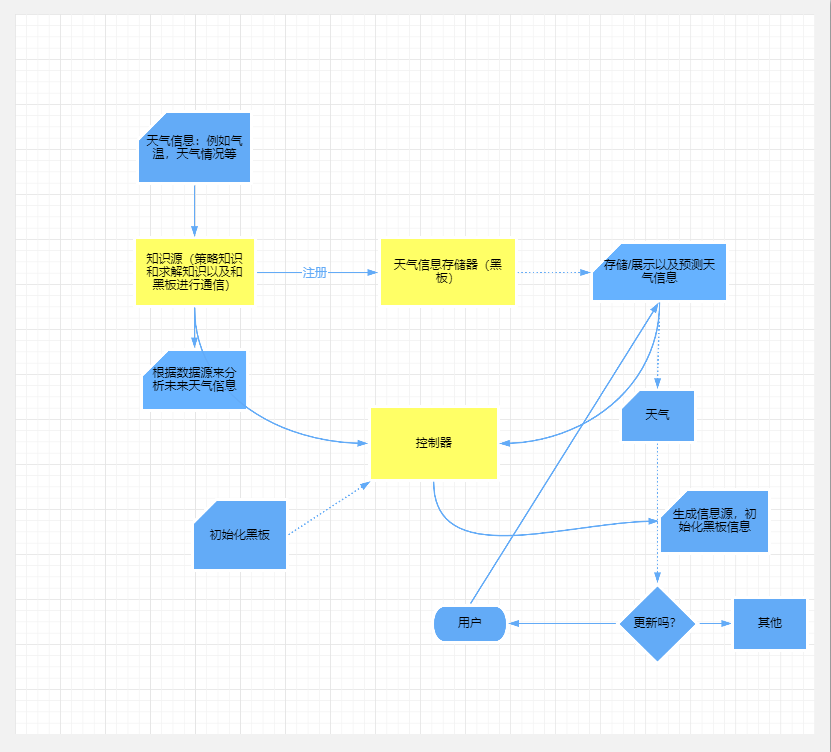
2) 黑板：提供注册方法，知识源用来更新知识。更新数据的方法。

3) 控制器：判断是否向用户推送信息。

**黑板模式的应用场景是要解决的任务可以分为多个子任务。解决方案如下图所示：**

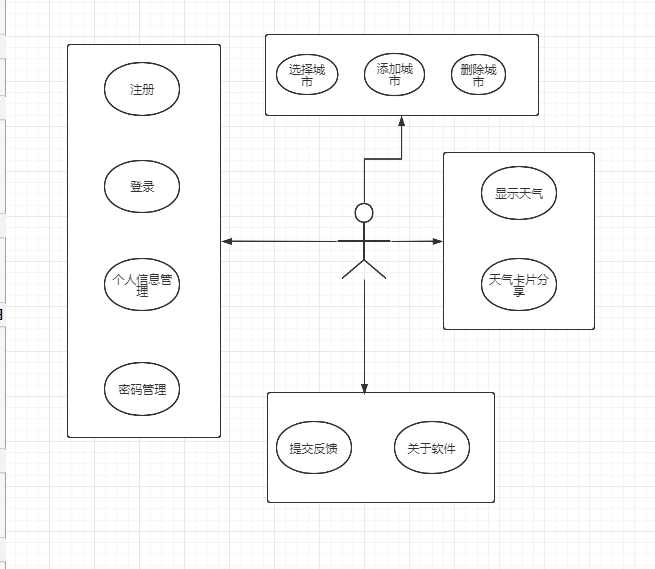


1. **天气预报系统简介**

天气信息和人们的日常生活息息相关，随着人们活动范围的扩大和出行需要，及时精确的获取天气信息显得越来越重要。为此，本次实验应用黑板风格的软件设计思想设计了天气预报系统，为用户提供天气相关的信息，用户可以通过该系统及时获取天气的状况和变化，及时做好各方面的计划和防御措施。该系统主要由基本信息源，气象分析器，消息存储器三部分组成如图x所示，用户通过订阅消息存储器之后，如果消息存储器中的天气信息更新，系统自动推送消息给用户。

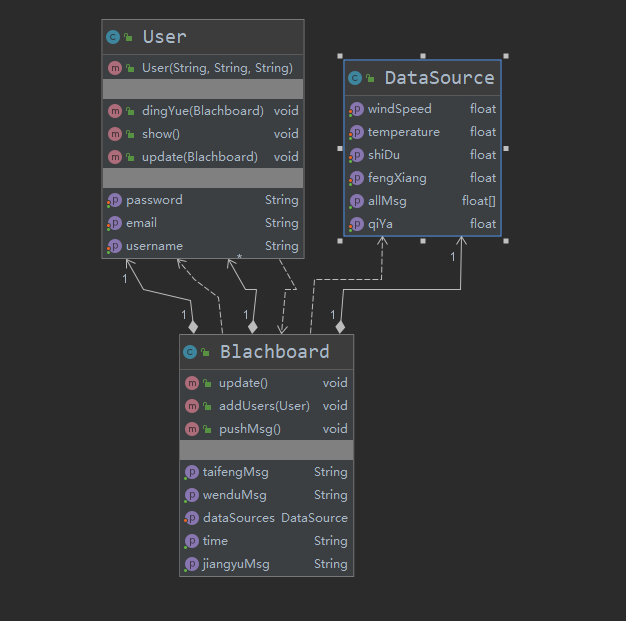
系统模型图

1. **模型图功能解释**
2. 知识源，用于存储各种信息更新。
3. 黑板，天气信息存储器。
4. 控制器，用于连接知识源和黑板。
5. 更新情况。
6. **用例图及说明**



|  |  |
| --- | --- |
| **用例名** | **用例介绍** |
| **注册** | **为用户提供注册功能** |
| **登录** | **账号登陆功能** |
| **个人信息** | **个人信息编辑** |
| **密码管理** | **用户可以对密码进行管理** |
| **选择城市** | **可以对城市进行选择** |
| **添加城市** | **可以添加所想要的城市** |
| **删除城市** | **可以删除已经添加的城市** |
| **显示天气** | **可以显示添加城市的天气** |
| **天气卡片分享** | **可以对天气进行分享** |
| **提交反馈** | **对软件提交意见** |
| **关于软件** | **查询软件信息** |

# 类图及说明



User类即为用户类

DataSource 类为数据源类

Blachboard类为天气信息储存器以及气象分析器

1. **实验总结**

黑板模式的实现方法

黑板模式一般不会对架构产生什么影响，但它通常会要求有一个清晰的消息结构。黑板模式一般都会提供一系列的过滤器，以便消息的消费者不再接触到与自己无关的消息。在实际开发中，黑板模式常见的有两种实现方式。

● 数据库作为黑板

利用数据库充当黑板，生产者更新数据信息，不同的消费者共享数据库中信息，这是最常见的实现方式。该方式在技术上容易实现，开发量较少，熟悉度较高。缺点是在大量消息和高频率访问的情况下，性能会受到一定影响。

在该模式下，消息的读取是通过消费者主动“拉取”，因此该模式也叫做“拉模式”。

● 消息队列作为黑板

以消息队列作为黑板，通过订阅-发布模型即可实现黑板模式。这也是黑板模式被淡忘的一个重要原因：消息队列（Message Queue）已经非常普及了，做Java开发的已经没有几个不知道消息队列的。

在该模式下，消费者接收到的消息是被主动推送过来的，因此该模式也称为“推模式”。

提示　黑板模式不做详细讲解，因为我们现在已经在大量使用消息队列，既可以做到消息的同步处理，也可以实现异步处理，相信大家已经在开发中广泛使用了，它已经成为跨系统交互的一个事实标准了。

1. **代码:**

**public void setFirst\_name(String first\_name) {**

**this.first\_name = first\_name;**

**}**

**public String getLast\_name() {**

**return last\_name;**

**}**

**public void setLast\_name(String last\_name) {**

**this.last\_name = last\_name;**

**}**

**public String getEmail() {**

**return email;**

**}**

**public void setEmail(String email) {**

**this.email = email;**

**}**

**public Boolean getIs\_active() {**

**return is\_active;**

**}**

**public void setIs\_active(Boolean is\_active) {**

**this.is\_active = is\_active;**

**}**

**public Date getData\_joined() {**

**return data\_joined;**

**}**

**public void setData\_joined(Date data\_joined) {**

**this.data\_joined = data\_joined;**

**}**

**}**

**package weatherBlackBoard;**

**public interface KnowledgeSource {**

**public void updateBlackBoard(BlackBoard blackBoard);**

**}**

**AirPressureKnowledge.java**

**package weatherBlackBoard;**

**public class AirPressureKnowledge implements KnowledgeSource {**

**private Double airPress;**

**public AirPressureKnowledge(Double airPress) {**

**this.airPress = airPress;**

**}**

**public Double getAirPress() {**

**return airPress;**

**}**

**public void setAirPress(Double airPress) {**

**this.airPress = airPress;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setAirPressure(airPress);**

**}**

**}**

**HumidityKnowledge.java**

**package weatherBlackBoard;**

**public class HumidityKnowledge implements KnowledgeSource {**

**private Double humidity;**

**public HumidityKnowledge(Double humidity) {**

**this.humidity = humidity;**

**}**

**public Double getHumidity() {**

**8**

**return humidity;**

**}**

**public void setHumidity(Double humidity) {**

**this.humidity = humidity;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setHumidity(humidity);**

**}**

**}**

**RainfallForecaster.java**

**package weatherBlackBoard;**

**public class RainfallForecaster implements KnowledgeSource {**

**private Boolean isRainfall;**

**public RainfallForecaster(Boolean isRainfall) {**

**this.isRainfall = isRainfall;**

**}**

**public Boolean getRainfall() {**

**return isRainfall;**

**}**

**public void setRainfall(Boolean rainfall) {**

**isRainfall = rainfall;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setRainfall(isRainfall);**

**}**

**}**

**TemperatureKnowledge.java**

**package weatherBlackBoard;**

**public class TemperatureKnowledge implements KnowledgeSource {**

**private Double temperature;**

**public TemperatureKnowledge(Double temperature) {**

**this.temperature = temperature;**

**}**

**9**

**public Double getTemperature() {**

**return temperature;**

**}**

**public void setTemperature(Double temperature) {**

**this.temperature = temperature;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setTemperature(temperature);**

**}**

**}**

**TyohoonForecaster.java**

**package weatherBlackBoard;**

**public class TyphoonForecaster implements KnowledgeSource {**

**private Boolean isTyphoon;**

**public TyphoonForecaster(Boolean isTyphoon) {**

**this.isTyphoon = isTyphoon;**

**}**

**public Boolean getTyphoon() {**

**return isTyphoon;**

**}**

**public void setTyphoon(Boolean typhoon) {**

**isTyphoon = typhoon;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setTyphoon(isTyphoon);**

**}**

**}**

**WindSpeedKnowledge.java**

**package weatherBlackBoard;**

**public class WindSpeedKnowledge implements KnowledgeSource {**

**private Double windSpeed;**

**public WindSpeedKnowledge(Double windSpeed) {**

**10**

**this.windSpeed = windSpeed;**

**}**

**public Double getWindSpeed() {**

**return windSpeed;**

**}**

**public void setWindSpeed(Double windSpeed) {**

**this.windSpeed = windSpeed;**

**}**

**@Override**

**public void updateBlackBoard(BlackBoard blackBoard) {**

**blackBoard.setWindSpeed(windSpeed);**

**}**

**}**

**BlackBoard.java**

**package weatherBlackBoard;**

**import java.util.List;**

**public class BlackBoard{**

**private Double airPressure;**

**private Double temperature;**

**private Double windSpeed;**

**private Double humidity;**

**private Boolean isRainfall;**

**private Boolean isTyphoon;**

**public BlackBoard(Double airPressure, Double temperature, Double windSpeed,**

**Double humidity, Boolean isRainfall, Boolean isTyphoon) {**

**this.airPressure = airPressure;**

**this.temperature = temperature;**

**this.windSpeed = windSpeed;**

**this.humidity = humidity;**

**this.isRainfall = isRainfall;**

**this.isTyphoon = isTyphoon;**

**}**

**public Double getAirPressure() {**

**return airPressure;**

**}**

**11**

**public void setAirPressure(Double airPressure) {**

**this.airPressure = airPressure;**

**}**

**public Double getTemperature() {**

**return temperature;**

**}**

**public void setTemperature(Double temperature) {**

**this.temperature = temperature;**

**}**

**public Double getWindSpeed() {**

**return windSpeed;**

**}**

**public void setWindSpeed(Double windSpeed) {**

**this.windSpeed = windSpeed;**

**}**

**public Double getHumidity() {**

**return humidity;**

**}**

**public void setHumidity(Double humidity) {**

**this.humidity = humidity;**

**}**

**public Boolean getRainfall() {**

**return isRainfall;**

**}**

**public void setRainfall(Boolean rainfall) {**

**isRainfall = rainfall;**

**}**

**public Boolean getTyphoon() {**

**return isTyphoon;**

**}**

**public void setTyphoon(Boolean typhoon) {**

**isTyphoon = typhoon;**

**}**

**}**

**12**

**Control.java**

**package weatherBlackBoard;**

**import java.util.List;**

**public class Control{**

**private BlackBoard blackBoard;**

**private List<KnowledgeSource> knowledgeSourceList;**

**public Control(BlackBoard blackBoard, List<KnowledgeSource>**

**knowledgeSourceList) {**

**this.blackBoard = blackBoard;**

**this.knowledgeSourceList = knowledgeSourceList;**

**}**

**public BlackBoard getBlackBoard() {**

**return blackBoard;**

**}**

**public void setBlackBoard(BlackBoard blackBoard) {**

**this.blackBoard = blackBoard;**

**}**

**public List<KnowledgeSource> getKnowledgeSourceList() {**

**return knowledgeSourceList;**

**}**

**public void setKnowledgeSourceList(List<KnowledgeSource> knowledgeSourceList) {**

**this.knowledgeSourceList = knowledgeSourceList;**

**}**

**public void monitorBlackBoard(){**

**}**

**public void useKnowledgeSource(){**

**}**

**}**

**User..java**

**package weatherBlackBoard;**

**import java.util.Date;**

**13**

**public class User {**

**private Long uid;**

**private String username;**

**private String first\_name;**

**private String last\_name;**

**private String email;**

**private Boolean is\_active;**

**private Date data\_joined;**

**private BlackBoard blackBoard;**

**public User(Long uid, String username, String first\_name, String last\_name,**

**String email, Boolean is\_active, Date data\_joined) {**

**this.uid = uid;**

**this.username = username;**

**this.first\_name = first\_name;**

**this.last\_name = last\_name;**

**this.email = email;**

**this.is\_active = is\_active;**

**this.data\_joined = data\_joined;**

**}**

**public void getData(){**

**}**

**public Long getUid() {**

**return uid;**

**}**

**public void setUid(Long uid) {**

**this.uid = uid;**

**}**

**public String getUsername() {**

**return username;**

**}**

**public void setUsername(String username) {**

**this.username = username;**

**}**

**public String getFirst\_name() {**

**return first\_name;**

**14**

**}**