**《 智能系统 》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学号** | | **姓名** | **承担任务** | | | **贡献度** | **得分** |
| **20204148** | | **方成林** | **知识库的设计和实现** | | | **A** | **90** |
| **20204155** | | **蓝兴烨** | **数据库的设计和实现，撰写报告** | | | **A** | **90** |
|  | |  |  | | |  |  |
|  | |  |  | | |  |  |
|  | |  |  | | |  |  |
|  | |  |  | | |  |  |
|  | |  |  | | |  |  |
| **实验题目** | **数据库与知识库设计** | | | | | | |
| **实验时间** | **2023/5/01** | | | **实验地点** | **DS3401** | | |
| **实验成绩** |  | | | **实验性质** | **□验证性 □设计性 ☑综合性** | | |
| 教师评价：  □算法/实验过程正确 □源程序/实验内容提交  □程序结构/实验步骤合理 □实验结果正确  □语法、语义正确 □报告规范  其他：  评价教师签名： | | | | | | | |
| 1. 实验目的   为实现十字路口红绿灯智能控制，本次实验的目的是：   1. 练习知识库的设计，测试与维护 2. 大型知识库的实现设计。 3. 可信度知识库的设计与实现（可选） | | | | | | | |
| 1. 实验项目内容 2. 基于pyknow，设计十字路口红绿灯智能控制的知识库 3. 基于pyknow语法，给出十字路口红绿灯智能控制的完整知识库。 4. 通过模拟运行，测试并修改知识库，保证十字路口红绿灯智能控制的正确性。 5. 基于关系数据库管理系统，（如mysql），设计并实现大型知识库的存储和管理。 6. 为什么需要用关系数据库来存储大型知识库？ 7. 对1中的知识库，设计出表结构（给出DDL程序） 8. 将1中的知识库存入关系数据库（给出DML程序） 9. 举例给出知识库的使用和维护过程（给出DML程序）   3、可信度知识库设计（可选） | | | | | | | |
| 三、实验过程或算法（代码）  1、基于pyknow，设计十字路口红绿灯智能控制的知识库  （1）基于pyknow语法，给出十字路口红绿灯智能控制的完整知识库。  **默认规则**   1. **红绿灯总时间60s** 2. **红绿灯每分钟改变一次**   **知识库**  因为南北车辆+东西车辆=总车辆，所以制定红绿灯的时间分配（switchTime）只需考虑一侧方向的车辆占总车辆的比值，如果东西方向的车辆占比多于一半，则增加东西方向的绿灯时间，假设一开始东西方向为绿灯，南北方向为红灯。   1. **如果东西车辆等于总车辆的一半==>东西方向的绿灯时间设定为30s** 2. **如果东西车辆大于总车辆的一半==>东西方向的绿灯时间增加到40s** 3. **如果东西车辆小于总车辆的一半==>东西方向的绿灯时间减少到20s**   代码见附件  （2）通过模拟运行，测试并修改知识库，保证十字路口红绿灯智能控制的正确性。  通过生成随机数得到每分钟东西方向和总的车辆数目，并且测试运行。      def generateCars(self):          global TotalCars          global WECars          TotalCars = random.randint(10, 100)          WECars= random.randint(0,TotalCars)          print("TotalCars=",TotalCars,"\n")          print("WECars=",WECars,"\n")  2、基于关系数据库管理系统，（如mysql），设计并实现大型知识库的存储和管理。    （1）为什么需要用关系数据库来存储大型知识库？  关系数据库更加稳定且易于维护。  （2）对1中的知识库，设计出表结构（给出DDL程序）  import pymysql  import pymysql  import pandas as pd  try:      # 打开数据库连接      db = pymysql.connect(host="localhost", user="root", password="123", database="mysql")      print('数据库连接成功！')      cursor = db.cursor()      sql = "SELECT \* FROM trafficLight"      cursor.execute(sql)      mydata = cursor.fetchall()  # 获取全部数据      # for data in mydata:      #     print(list(data))      # db.close()      pdate = pd.read\_sql(sql, db)  # 以DataFrame格式读取显示      print(pdate)      db.close()  except:      print("数据库连接失败:" + str(pymysql.Error))  （3）将1中的知识库存入关系数据库（给出DML程序）  #根据规则得出需要存储的数据值  switchtime=40 if(WECars)>(TotalCars/2) else 20;  #得到需存储值后插入数据  INSERT INTO trafficLight (NScar,EWcar,NSmorethanEW,AddNS,AddEW,reset) VALUES (NSCars,EWcars,NSmorethanEW\_p,AddNS\_p,AddEW\_p,reset\_p);  （4）举例给出知识库的使用和维护过程（给出DML程序）  SELECT \* from trafficLight where id = (SELECT max(id) FROM trafficLights); #获取最新一条的数据  把结果作为事实放进pyknow中运行 | | | | | | | |
| 1. 实验结果及分析   结果符合预期  （1）当东西方向车辆较少时，对应绿灯时间变为20s    （2）反之变为40s | | | | | | | |