.文 **mts**

°问题描述

°你的

。接口说明

°调用示例

°注意事项

文件处理

问题描述

南京市政府正在统计近日出入南京市区的车辆名单。你需要帮助政府实现T\*简单的数据库，记录这些车辆 信息，并支持增删爐功能。

你的任务

完成**CarRecord**类的实现，该类封装了车辆出入记录，包含以下信息：

1. 车牌号:由**5**位的大写字母或数字輙
2. 日期：不考虑年份和月份，取值为**1-31**的整数
3. 时间：包括小时和分钟。

完成**CarDatabase**类的实现。该类具有以下几个功能：

1. **CarRecord**对象的增删改查。你应该将**CarDatabase**的信息保存在f 文件(文件格式可以自由定 义)中，并且在每炒行操作时，将相应的变更同步更新到文件当中。
2. 记录文件的读取：可以从读取你保存的文件，将其中的所有信息恢复到内存当中。

接口说明

你需要在自己定义的**CarDatabase**类中至少实现以下接口，请队專阅读注释

//CarDatabase的构造函数，filename为你可以用于记录数据的文件名 CarDatabase::CarDatabase(string filename);

〃读取你保存的文件，读入的数据应覆盖当前CarDatabase实例的数据 void CarDatabase::read();

〃添加一条CarRecord信息，对于数据完全相同的重复记录只添加一条

void CarDatabase::add(CarRecord cr);

〃删除车牌号为car的所有记录

void CarDatabase::del(string car);

〃删除日期为date的所有记录

void CarDatabase::del(int date);

〃删除日期为date且车牌号为car的所有记录

void CarDatabase::del(string car,int date);

//time的格式为"hh:mm",当小时或分钟不足两位时，有一位前导0

//以下所有时间的输入和输出格式都与此相同

〃删除日期为date、车牌号为car且时间为time的所有记录

void CarDatabase::del(string car,int date,string time);

〃删除日期为date,时间介于time\_start和time\_end之间(包®time\_start和time\_end)的记录 void CarDatabase::del(int date,string time\_start,string time\_end);

//查找车牌号为car的所有记录，结果按时间顺序(日期 小时分钟)从早到晚排列 vector<CarRecord> CarDatabase::IndexRecordsByCar(string car);

〃查找日期为date的所有记录，多次出入的车辆只保留最后一个(时间最晚的)记录，按时间顺序从早 到晚排列

〃若两车出入的时间相同，贝!I按车牌号的字典序由小到大排列

vector<CarRecord> CarDatabase::IndexRecordsByDate(int date);

对**CarRecord**类，你需要实现

//CarRecord的默认构造函数

CarRecord::CarRecord();

//set函数

void CarRecord::set(string car^int date,string time);

〃各个get函数

string CarRecord::getCar();

int CarRecord::getDate();

〃你可以用任何形式记录时间，但返回时一定是"hhzmm"格式的字符串 string CarRecord::getTime();

调用示例

CarDatabase类的调用示例

CarRecord cr;

cr.set(nA9999",*1,*"16:33”)；

CarDatabase cd("database11),cd2("database");

cd.add(cr);

cd2.read();

vector<CarRecord> cars=cd2.IndexRecordsByCar("A9999"); //...

注意事项

1. 善加利用**STL**
2. 你磯完成一个类，并创建以下四个文件，将其打包为**zip**压缩包上传;

xxxx.zip

I

|--CarDatabase.h

| --CarDatabase.cpp

I--CarRecord.h

I --CarRecord.cpp

1. 你可以在类中增加新的函数，但不要改变类的命名及接口，否则无法通过测试;
2. 注意内存安全，避免内存泄漏；
3. 注戒件编码格式为utf-8；
4. 注意不要在提交的源代码中包含**main**函数；