  
“数据结构”

课程设计报告

**设计题目**  汽车租赁管理系统

**姓 名**  张渊

**学 号**  2022210000

**专 业**  计算机科学与技术

**班 级**  22-0

**完成日期**

课程设计成绩评定表

|  |  |  |  |
| --- | --- | --- | --- |
| **设计题目** | 汽车租赁管理系统 | **成绩** |  |
| **内容简介** | **要求**：  ① 车辆基本信息管理（车辆的添加、删除、修改）；  ② 租车管理功能（租车、换车、计费等）；  ③ 统计功能（统计出租率、出租费用）。  本系统包含的功能  1.添加新的车辆  2.删除已有车辆  3.修改车辆信息  4.租用车辆  5.归还车辆  6.展示现有的车辆信息  7.统计目前的出租率和出租费用  0.退出程序  本系统可以添加新的车辆、删除已有车辆、修改现有的车辆信息、租用车辆、归还车辆、展示现有的车辆信息、统计目前（正在被租用车辆）的出租率和出租费用  本系统包含了文件的创建、读取功能。启动后可以在程序所在的文件夹内查看目前车辆的基本信息，实现长期储存。 | | |
| **评语** | **教师签名：** | | |

1. **需求和规格说明**

设计实现一个简单的汽车租赁管理系统。

**要求**：

① 车辆基本信息管理（车辆的添加、删除、修改）；

② 租车管理功能（租车、换车、计费等）；

③ 统计功能（统计出租率、出租费用）。

1. **设计**

车辆信息通过定义一个Car类来封装，其中包含车牌号码、车品牌、租车价格、 是否出租、租车时间等基本属性。系统管理功能通过定义一个RentCarManager类来实现，其中包含添加车辆、删除车辆、修改车辆信息、租车、还车、展示车辆信息、统计出租率和出租费用等方法。  
 程序使用了菜单式交互方式，让用户选择要执行的具体操作。  
 在执行每个具体操作之前，程序会自动加载车辆信息，执行完毕后再保存车辆信息到文件，以保证汽车租赁系统的数据一致性。  
 统计功能中，出租率通过已出租车辆数除以总车辆数计算得出，出租费用总额则将所有已出租车辆的租车费用相加所得。  
 程序通过读取和保存车辆信息的文本文件来实现持久化数据存储。

**系统类图**

Car

string plate; //车牌号码

string brand; //车品牌

double price; //租车价格

bool is\_rented; //是否出租

int time; //租车时间

CarManager

vector<Car> car\_list; //车辆列表

loadCarsFromFile() //从文件读取车辆信息

saveCarsToFile()//将车辆信息保存到文件

addCar()//添加车辆

removeCar()//删除车辆

modifyCar()//修改车辆信息

rentCar()//租车

returnCar()//归还车辆

showCars()//展示所有车辆信息

showStats()//展示出租率和出租费用

**属性和方法定义**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类名** | **成员类别** | **类型** | **成员名** | **描述** |
| Car | 属性 | string | plate | 车牌号码 |
| string | brand | 车品牌 |
| double | price | 租车价格 |
| bool | Is\_rented | 是否出租 |
| int | time | 租车时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类名** | **成员类别** | **类型** | **成员名** | **描述** |
| CarManager | 属性 | Car | car\_list | 车辆列表 |
| 方法 | void | loadCarsFromFile()) | 从文件读取车辆信息 |
| void | saveCarsToFile() | 将车辆信息保存到文件 |
| void | addCar() | 添加车辆 |
| void | removeCar() | 删除车辆 |
| void | modifyCar() | 修改车辆信息 |
| void | rentCar() | 租车 |
| void | returnCar() | 归还车辆 |
| void | showCars() | 展示所有车辆信息 |
| void | showStats() | 展示出租率和出租费用 |

1. **用户手册**

程序运行时，首先会提示输入您想要执行的操作

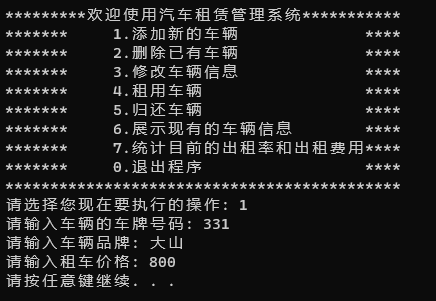
1. 添加新的车辆 2.删除已有车辆3.修改车辆信息 4.租用车辆 5.归还车辆 6.展示现有的车辆信息 7.统计目前的出租率和出租费用0.退出程序

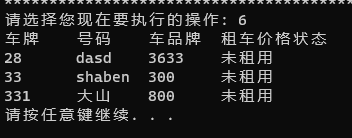
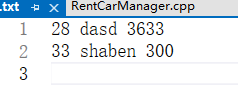
在输入的过程中按照程序对应的提示执行即可

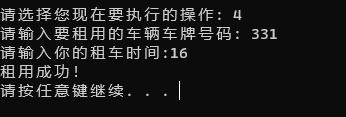
1. **调试及测试**

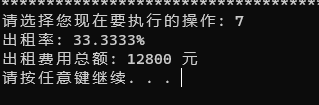
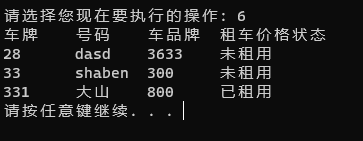
在调试过程中发现了一些问题。计算当前的出租费用时，由于起初设计缺陷，导致只算了一个单位的租车费用，无法按照时间收费。在一些地方的判断出现了失误，导致出现错误程序也没有相应的回应。

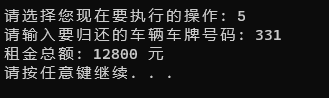
1. **运行实例**

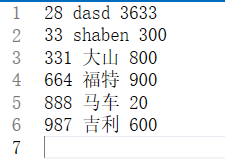
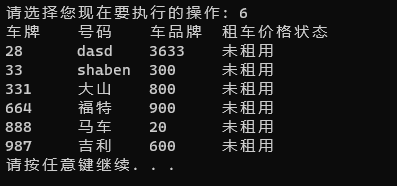


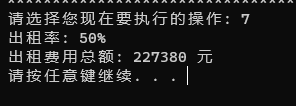


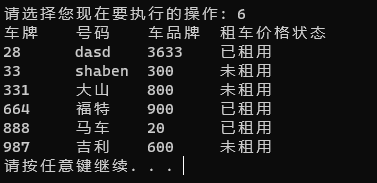


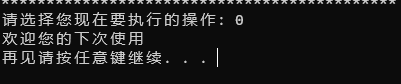
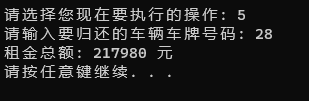












1. **进一步改善**
2. 为了解决计算收费的问题，我在car类里又添加了一个计算时间的属性，而不是在对应的函数中添加，从而使得时间成为car的属性，而不是局部变量。

（2）为了解决出现错误无法回应的问题，我在对应的程序中添加了适当的判断，提高了代码的健壮性，使得出现错误可以作出相应的回应，很大程度上避免了程序的崩溃。

1. **心得体会**

在完成这个课程设计的过程中，我深深地感受到了软件开发的复杂性和挑战性。在设计和实现一个简单的汽车租赁管理系统时，需要考虑很多方面，比如如何设计合理的数据结构、如何实现各种复杂的业务逻辑、如何优化程序性能等等。

在我的设计中，租车和换车操作都需要对车辆的状态进行更新，这会导致频繁的数据操作和文件读写操作，从而降低程序的性能。为了解决这个问题，我通过数据操作以减少数据和文件系统的访问次数，从而提高程序的响应速度和性能表现。

我在统计功能的实现上也遇到了一些问题。在我的设计中，我选择了使用一些简单的算法和数据结构来实现租赁率和出租费用的统计功能，但是相比较复杂的实际情况，统计算法和数据结构显然还存在很大的改进空间。因此，在实际应用中，需要根据具体的业务需求来选择更加高效和精准的统计算法和数据结构。

总体而言，这个课程设计帮助我深入了解了软件开发的复杂性和挑战性，更加深入地理解了数据结构和算法的重要性，同时也让我在实践中不断提升自己的编程技能和解决问题的能力。我相信，在今后的实际工作中，这些经验和教训一定会对我的工作有所帮助。

1. **对课程设计的建议**

第一，要注重实践，把理论运用到实际中去。数据结构是一门理论与实践相结合的课程，在课程设计中，可以引入一些实际的数据结构问题，并给学生实际的编程任务，引导学生深入思考和理解数据结构的实际应用。

第二，要注重项目设计，注重组织能力和创新思维。在现代社会，软件开发不仅需要掌握技术能力，更需要组织能力和创新思维。因此，在课程设计中可以引入项目式教学，让学生通过团队合作，完成一个完整的软件开发项目。这样既可以锻炼学生的技术能力，也可以培养学生的组织能力和创新思维。

第三，要注重细节。在课程设计中，要特别注重细节，如代码的可读性与可维护性，注释的规范与完整性，错误处理与界面交互等。

最后，我认为课程设计还是要“量力而行”，根据学生的掌握程度和实际情况来确定适合的难度和设计方案，防止学生过度压力和迷失方向。

以上是我的建议，感谢老师。

1. **附录⎯⎯源程序**

****（在Word内启动无法创建文件 应当在文件夹里启动程序，便会在文件夹里创建程序对应的文件）

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <ctime>

#include <iomanip>

using namespace std;

// 定义汽车类

class Car {

public:

string plate; //车牌号码

string brand; //车品牌

double price; //租车价格

bool is\_rented; //是否出租

int time; //租车时间

Car(string p, string b, double pr) : plate(p), brand(b), price(pr), is\_rented(false) {}

};

// 定义汽车租赁管理系统类

class CarManager {

private:

vector<Car> car\_list; //车辆列表

public:

CarManager() {} //构造函数

void loadCarsFromFile() { //从文件读取车辆信息

ifstream input\_file("cars.txt");

string plate, brand;

double price;

while (input\_file >> plate >> brand >> price) {

car\_list.push\_back(Car(plate, brand, price));

}

input\_file.close();

}

void saveCarsToFile() { //将车辆信息保存到文件

ofstream output\_file("cars.txt");

for (int i = 0; i < car\_list.size(); i++) {

output\_file << car\_list[i].plate << " " << car\_list[i].brand << " " << car\_list[i].price << endl;

}

output\_file.close();

}

void addCar() { //添加车辆

string plate, brand;

double price;

cout << "请输入车辆的车牌号码: ";

cin >> plate;

cout << "请输入车辆品牌: ";

cin >> brand;

cout << "请输入租车价格: ";

cin >> price;

car\_list.push\_back(Car(plate, brand, price));

saveCarsToFile();

}

void removeCar() { //删除车辆

string plate;

cout << "请输入要删除车辆的车牌号码: ";

cin >> plate;

for (int i = 0; i < car\_list.size(); i++) {

if (car\_list[i].plate == plate) {

car\_list.erase(car\_list.begin() + i);

saveCarsToFile();

cout << "删除成功！" << endl;

return;

}

}

cout << "未找到该车辆！" << endl;

}

void modifyCar() { //修改车辆信息

string plate;

cout << "请输入要修改车辆信息的车牌号码: ";

cin >> plate;

for (int i = 0; i < car\_list.size(); i++) {

if (car\_list[i].plate == plate) {

cout << "请输入要修改的车辆信息(车牌号码、车品牌、租车价格，用空格分隔): ";

cin >> car\_list[i].plate >> car\_list[i].brand >> car\_list[i].price;

saveCarsToFile();

return;

}

}

cout << "未找到该车辆！" << endl;

}

void rentCar() { //租车

string plate;

cout << "请输入要租用的车辆车牌号码: ";

cin >> plate;

for (int i = 0; i < car\_list.size(); i++) {

if (car\_list[i].plate == plate) {

if (car\_list[i].is\_rented) { //已经被租用

cout << "该车已被租用！" << endl;

}

else { //未被租用

car\_list[i].is\_rented = true;

cout << "请输入你的租车时间:";

int m\_time; cin >> m\_time;

car\_list[i].time = m\_time;

cout << "租用成功！" << endl;

}

saveCarsToFile();

return;

}

}

cout << "未找到该车辆！" << endl;

}

void returnCar() { //归还车辆

string plate;

double total\_price;

cout << "请输入要归还的车辆车牌号码: ";

cin >> plate;

for (int i = 0; i < car\_list.size(); i++) {

if (car\_list[i].plate == plate) {

if (car\_list[i].is\_rented) { // 已经被租用

car\_list[i].is\_rented = false;

total\_price = car\_list[i].time \* car\_list[i].price;

cout << "租金总额: " << total\_price << " 元" << endl;

saveCarsToFile();

}

else { // 未被租用

cout << "该车未被租用，无需归还！" << endl;

}

return;

}

}

cout << "未找到该车辆！" << endl;

}

void showCars() { //展示所有车辆信息

cout << "车牌\t" << "号码\t" << "车品牌\t" << "租车价格状态\t" << endl;

for (int i = 0; i < car\_list.size(); i++) {

cout << car\_list[i].plate << "\t" << car\_list[i].brand << "\t" << car\_list[i].price << "\t" << (car\_list[i].is\_rented ? "已租用" : "未租用") << endl;

}

}

void showStats() { //展示出租率和出租费用

int total\_num = car\_list.size(); //总车辆数

int rented\_num = 0; //已出租车辆数

double total\_fee = 0; //出租费用总额

for (int i = 0; i < car\_list.size(); i++) {

if (car\_list[i].is\_rented) {

rented\_num++;

total\_fee += car\_list[i].price \* car\_list[i].time;

}

}

double rent\_rate = (double)rented\_num / total\_num \* 100;

cout << "出租率: " << rent\_rate << "%" << endl;

cout << "出租费用总额: " << total\_fee << " 元" << endl;

}

};

int main() {

CarManager manager;

manager.loadCarsFromFile();

while (true) {

cout << "\*\*\*\*\*\*\*\*\*欢迎使用汽车租赁管理系统\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 1.添加新的车辆 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 2.删除已有车辆 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 3.修改车辆信息 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 4.租用车辆 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 5.归还车辆 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 6.展示现有的车辆信息 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 7.统计目前的出租率和出租费用\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\* 0.退出程序 \*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "请选择您现在要执行的操作: ";

int choice;

cin >> choice;

switch (choice) {

case 1:

manager.addCar();//添加新的车辆

system("pause");

system("cls");

break;

case 2:

manager.removeCar();//删除已有车辆

system("pause");

system("cls");

break;

case 3:

manager.modifyCar();//修改车辆信息

system("pause");

system("cls");

break;

case 4:

manager.rentCar();//租用车辆

system("pause");

system("cls");

break;

case 5:

manager.returnCar();//归还车辆

system("pause");

system("cls");

break;

case 6:

manager.showCars();//展示现有的车辆信息

system("pause");

system("cls");

break;

case 7:

manager.showStats();//统计目前的出租率和出租费用

system("pause");

system("cls");

break;

case 0:

{

manager.saveCarsToFile();//退出程序

cout << "欢迎您的下次使用" << endl << "再见";

system("pause");

system("cls");

return 0;

}

default:

{

cout << " 输入错误，请重新输入！" << endl;

system("pause");

system("cls");

}

}

}

return 0;

}