

车载软件设计基础

课后实践2



**2022至2023学年第 1 学期**

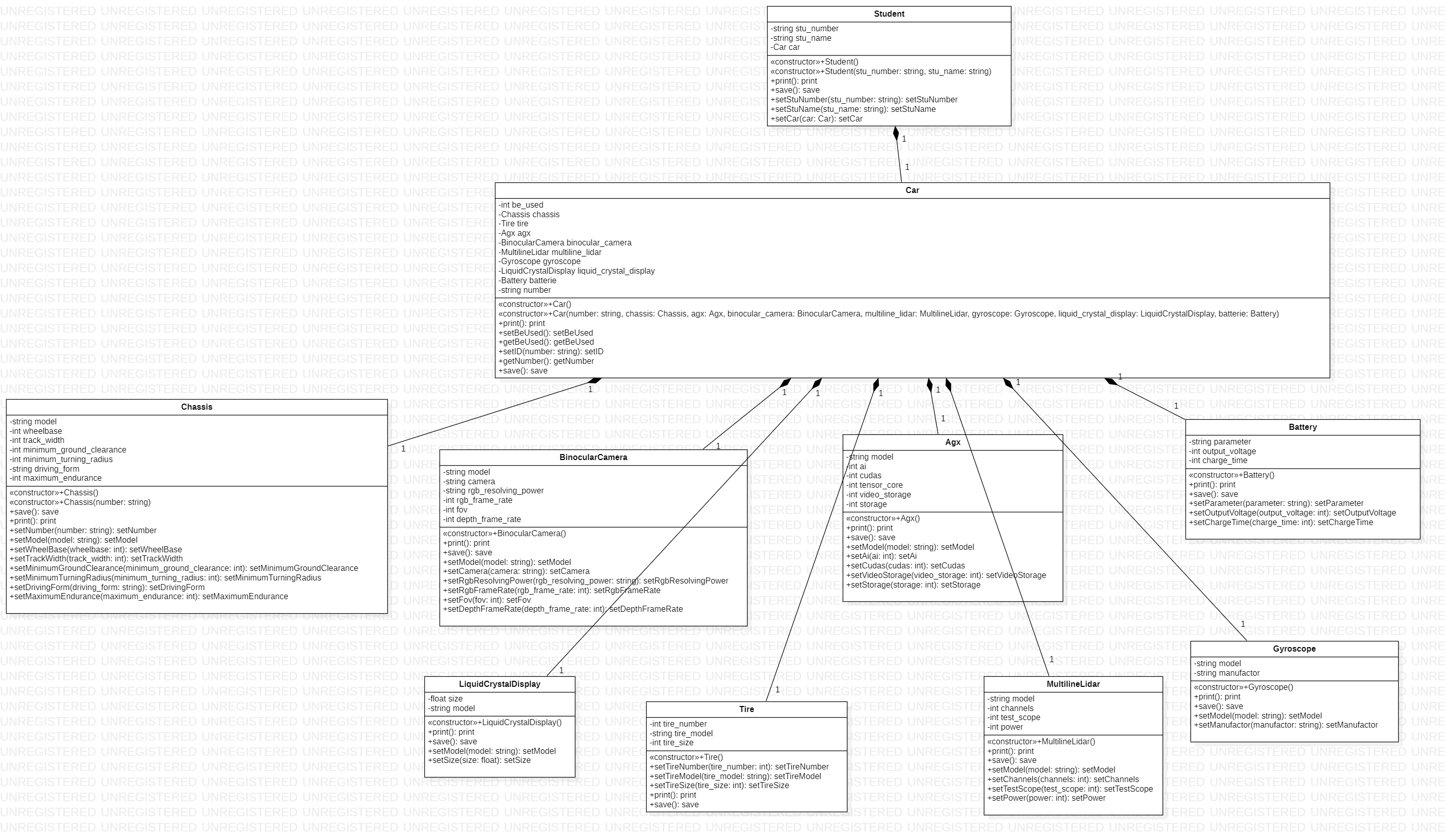
|  |  |
| --- | --- |
| 学号 | 姓名 |
| E2019051 | 曹代伟 |
| 任课教师 | 陈恒鑫 |
| 成 绩 |  |

|  |  |
| --- | --- |
| 任务书 | |
| 任务内容 | 请针对课后实践1的内容，将各模块用class改写。各模块的属性保持不变，增加相关操作行为。补充信息如下：   1. 智能小车   增加操作：   1. setID 实现编号录入 2. print 实现智能小车信息的屏幕输出 3. save 实现智能小车信息的本地文件保存 4. 底盘   增加操作：   1. 各属性的set方法 2. print 实现底盘信息的屏幕输出 3. save 实现底盘信息的本地文件保存 4. 轮胎   增加操作：   1. 各属性的set方法 2. print 实现轮胎信息的屏幕输出 3. save 实现轮胎信息的本地文件保存 4. AGX套件   增加操作：   1. 各属性的set方法 2. print 实现AGX套件的屏幕输出 3. save 实现AGX套件的本地文件保存 4. 双目摄像头   增加操作：   1. 各属性的set方法 2. print 实现双目摄像头信息的屏幕输出 3. save 实现双目摄像头信息的本地文件保存 4. 9轴陀螺仪   增加操作：   1. 各属性的set方法 2. print 实现9轴陀螺仪信息的屏幕输出 3. save 实现9轴陀螺仪信息的本地文件保存 4. 多线激光雷达   增加操作：   1. 各属性的set方法 2. print 实现多线激光雷达信息的屏幕输出 3. save 实现多线激光雷达信息的本地文件保存 4. 液晶显示屏   增加操作：   1. 各属性的set方法 2. print 实现液晶显示屏信息的屏幕输出 3. save 实现液晶显示屏信息的本地文件保存 4. 电池模块   增加操作：   1. 各属性的set方法 2. print 实现电池模块信息的屏幕输出 3. save 实现电池模块信息的本地文件保存 4. 学生信息包括：   增加操作：   1. 各属性的set方法 2. print 实现学生信息的屏幕输出 3. save 实现学生信息的本地文件保存   要求：   1. 上述方法的参数自定义。构造函数自定义 2. Main函数中实现小车信息录入、分配、保存和屏幕显示。 3. 小车各模块的保存和显示分别通过相应对象的操作完成 4. 用程序打开保存的文件，然后将相关信息显示在屏幕上（含分配的学生信息）。一次显示1辆小车信息，按n键显示下一辆小车信息，按p显示上一辆小车信息。请注意：显示第1台显示按p无效，显示第10台小车，按n无效。 |
| 程序规范 | （1）所有程序代码采用C++编写，使用git进行源代码管理；  （2）类名、变量名、函数名应符合C++的命名规范，并在代码中前后保持一致；  （3）涉及面向对象的程序，例如自定义的类，应符合面向对象的设计原则；  （4）正确使用头文件和源文件，自定义的头文件应符合头文件的编写原则，例如用条件宏定义确保头文件不被多次引用、不在头文件中进行类和函数的实现（模板除外）； |
| 报告要求 | （1）报告至少应该包括程序设计、程序效果展示、总结分析3个部分；  （2）程序设计描述组成程序的模块、类、函数以及他们之间的相互关系，若有算法，可以描述算法流程；  （3）程序效果展示除了程序运行效果截图之外，应该有必要的文字说明；  （4）总结分析可以分析实现的效果与理想情况的差异，分析导致这些差异的原因，切忌不要写成心得体会；  （5）报告应该格式规范、排版整洁、少语病和错误。 |
| 作业提交 | （1）含有git仓库（有.git目录）的完整源代码；  （2）任务报告。 |
| 评分标准 | 按照五级制打分，分为优秀、良好、中等、及格、不及格，各评分项占总成绩的比例为：  （1）任务完成情况占评分的60%；  （2）报告占评分的40%。  评分老师根据各部分的完成情况，直接给出总成绩。 |

**一、程序设计**

根据题意分析，一辆智能小车有唯一的编号，并且拥有底盘、轮胎、多线激光雷达、双目摄像头等多个部件，这些部件都是小车不可或缺的一部分，这是典型的整体与部分的关系，所以在设计类时优先考虑****组合(Composition)，**即Car类的数据成员包含Chassis(底盘)类、Tire(轮胎)类、Agx类等其它车辆部件对象。最后，每位学生拥有唯一一辆智能小车，所以考虑在Student的数据成员中加入Car类或者小车的编号来标识该学生拥有该智能小车，此处为了在print()和save()等输出操作时方便，选择直接以Car对象作为数据成员。**

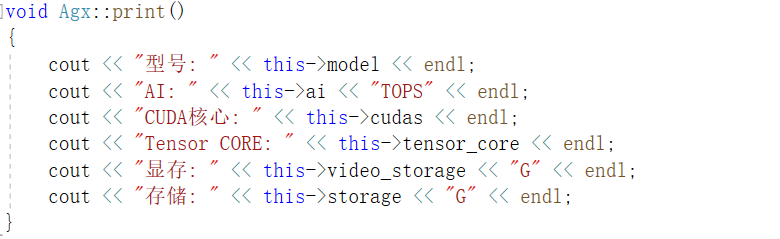




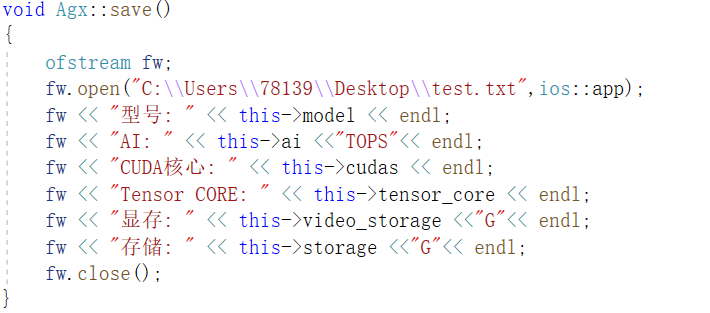
(类图)

由上述类图可知，程序设计了class Student、class Car、class Chassis、class Agx、class Tire、class Battery、class BinocularCamera、class Gyroscope、class LiquidCrystalDisplay、class MultilineLidar共10个class，每个class都具有print()方法和save()方法来进行对象数据的打印和保存。同时也具有各属性的set()方法。

其中以class Agx 举例说明：



print()方法示例



save()方法示例

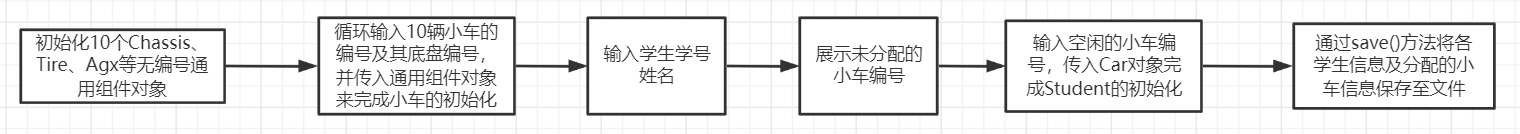


set()方法示例

为了从文本文件中读取数据并加以展示，设计了readData()函数。该函数首先通过ifstream打开文件并按行读取到某字符串中，直至读取到最后一行，即包括“充电时长”四字的一行。将读取的字符串数据保存到string message[10]中，然后通过while循环不断监听用户输入，当用户输“N”时，读取message中的下一组数据，当用户输入“P”时，读取上一组数据。当然还要考虑到数组越界的问题，在最后一组数据时，输入”N”无效。处于第一组数据时，输入”P”无效。



程序总体流程如下图：



完整main.cpp文件如下：

#include "Chassis.h"

#include "Car.h"

#include "Student.h"

#include "Agx.h"

#include "Tire.h"

#include "Battery.h"

#include "BinocularCamera.h"

#include "Gyroscope.h"

#include "LiquidCrystalDisplay.h"

#include "MultilineLidar.h"

#include "conio.h"

#include <iostream>

#include<string>

using namespace std;

void readData()

{

ifstream fr;

fr.open("C:\\Users\\78139\\Desktop\\test.txt", ios::in);

string buf;

string message[10];

string::size\_type idx;

int i = 0;

while (getline(fr, buf))

{

if (i == 10)

break;

message[i]=message[i]+buf+'\n';

idx = buf.find("充电时长");

if (idx != string::npos)

{

getline(fr, buf);

message[i] = message[i] + buf;

i++;

}

}

int current\_page = 0;

char c;

cout << message[0];

while(true)

{

c = \_getch();

if (c == 'n'&&current\_page<9)

{

system("cls");

cout << message[++current\_page];

}

if (c == 'p'&&current\_page>0)

{

system("cls");

cout << message[--current\_page];

}

}

}

int main()

{

ofstream fw;

fw.open("C:\\Users\\78139\\Desktop\\test.txt", ios::trunc);

fw.clear();

fw.close();

string chassis\_number;//底盘编号

string car\_number; //小车编号

//10个小车

Car cars[10];

//10个AGX套件

Agx agxs[10];

//10个双目摄像头

BinocularCamera binocular\_cameras[10];

//10个多线激光雷达

MultilineLidar multiline\_lidar[10];

//10个陀螺仪

Gyroscope gyroscope[10];

//10个显示屏

LiquidCrystalDisplay liquid\_crystal\_display[10];

//10个电池

Battery batteries[10];

//10套轮胎

Tire tires[10];

//创建10个电池、双目摄像头、轮胎、等无编号、通用的组件

for (int i = 0; i < 10; i++)

{

agxs[i] = Agx();

binocular\_cameras[i] = BinocularCamera();

multiline\_lidar[i] = MultilineLidar();

gyroscope[i] = Gyroscope();

liquid\_crystal\_display[i] = LiquidCrystalDisplay();

batteries[i] = Battery();

tires[i] = Tire();

}

for (int i = 0; i < 10; i++)

{

cout << "开始构建第: " << i + 1 << " 辆小车...." << endl;

cout << "请输入智能小车编号(cqusn打头的16位数字+字母)" << endl;

cin >> car\_number;

cout << "请输入底盘编号(dp打头的8位数字+字母)" << endl;

cin >> chassis\_number;

Chassis chassis(chassis\_number);

Car car(car\_number, chassis, agxs[i], binocular\_cameras[i],

multiline\_lidar[i],gyroscope[i], liquid\_crystal\_display[i], batteries[i],tires[i]);

cars[i] = car;

}

cout << "10台小车信息录入完毕，开始分配小车....." << endl;

string stu\_number, stu\_name;

Student student;

for (int i = 0; i < 10; i++)

{

cout << "请输入第: " << i + 1 << "位学生的学号及姓名!" << endl;

cin >> stu\_number >> stu\_name;

cout << "请为该学生分配小车，目前空闲的小车编号如下:" << endl;

for (int i = 0; i < 10; i++)

{

if (!cars[i].getBeUsed())

cout << cars[i].getNumber() << " ";

}

cout << endl;

cin >> car\_number;

for (int i = 0; i < 10; i++)

{

if (cars[i].getNumber() == car\_number)

{

cars[i].setBeUsed();

student = Student(stu\_number, stu\_name);

student.setCar(cars[i]);

}

}

student.save();

}

//从文件中读取信息

readData();

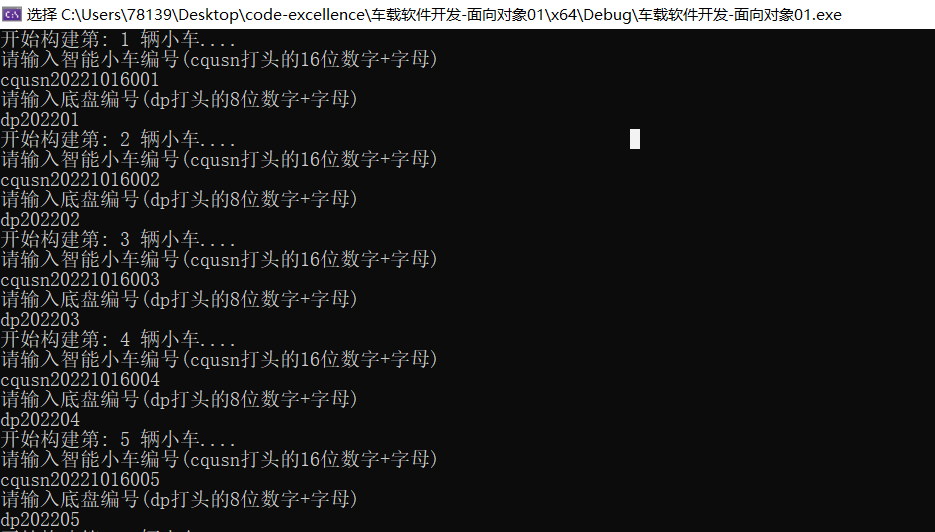
system("pause");

return 0;

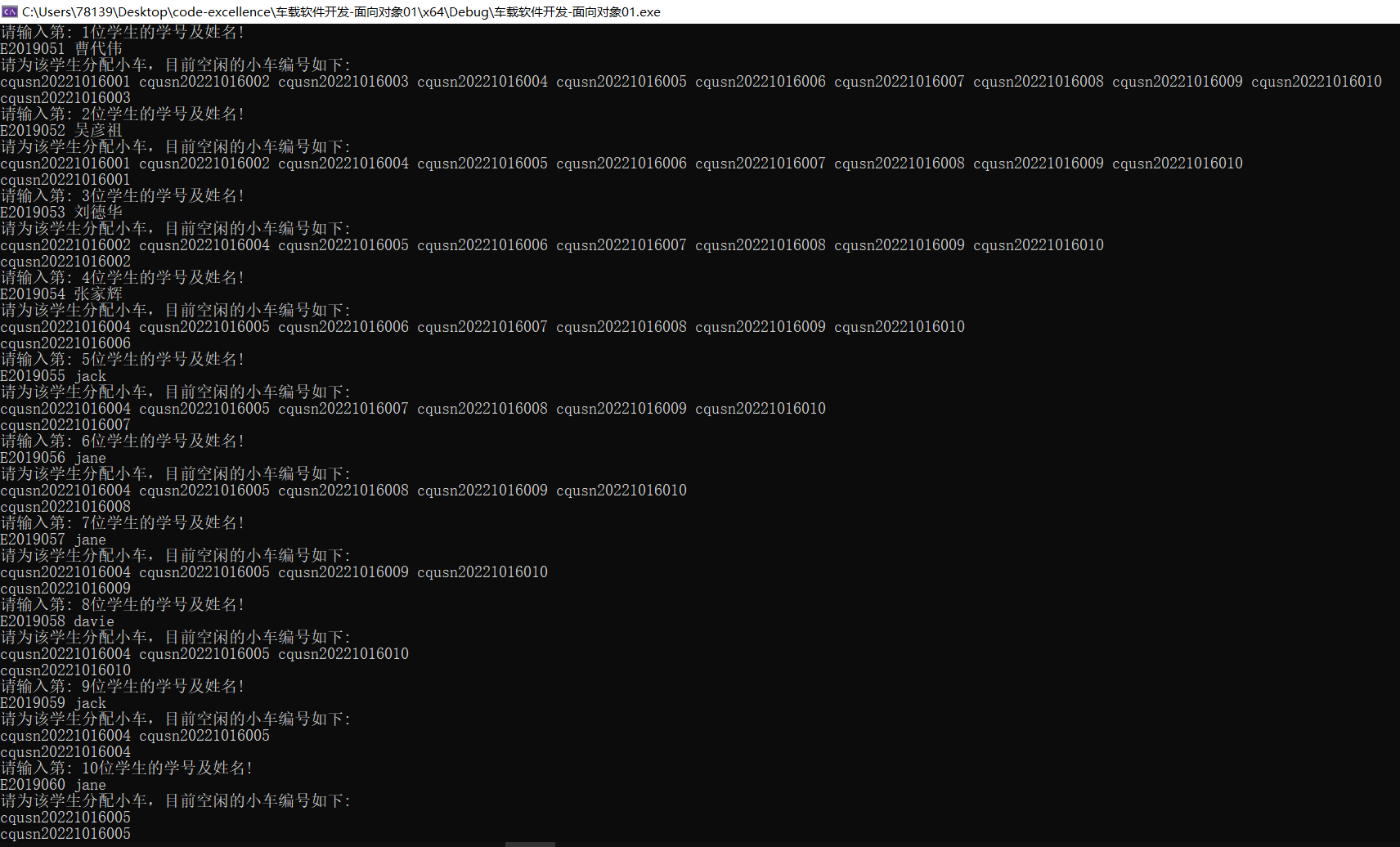
}

1. **程序效果展示**

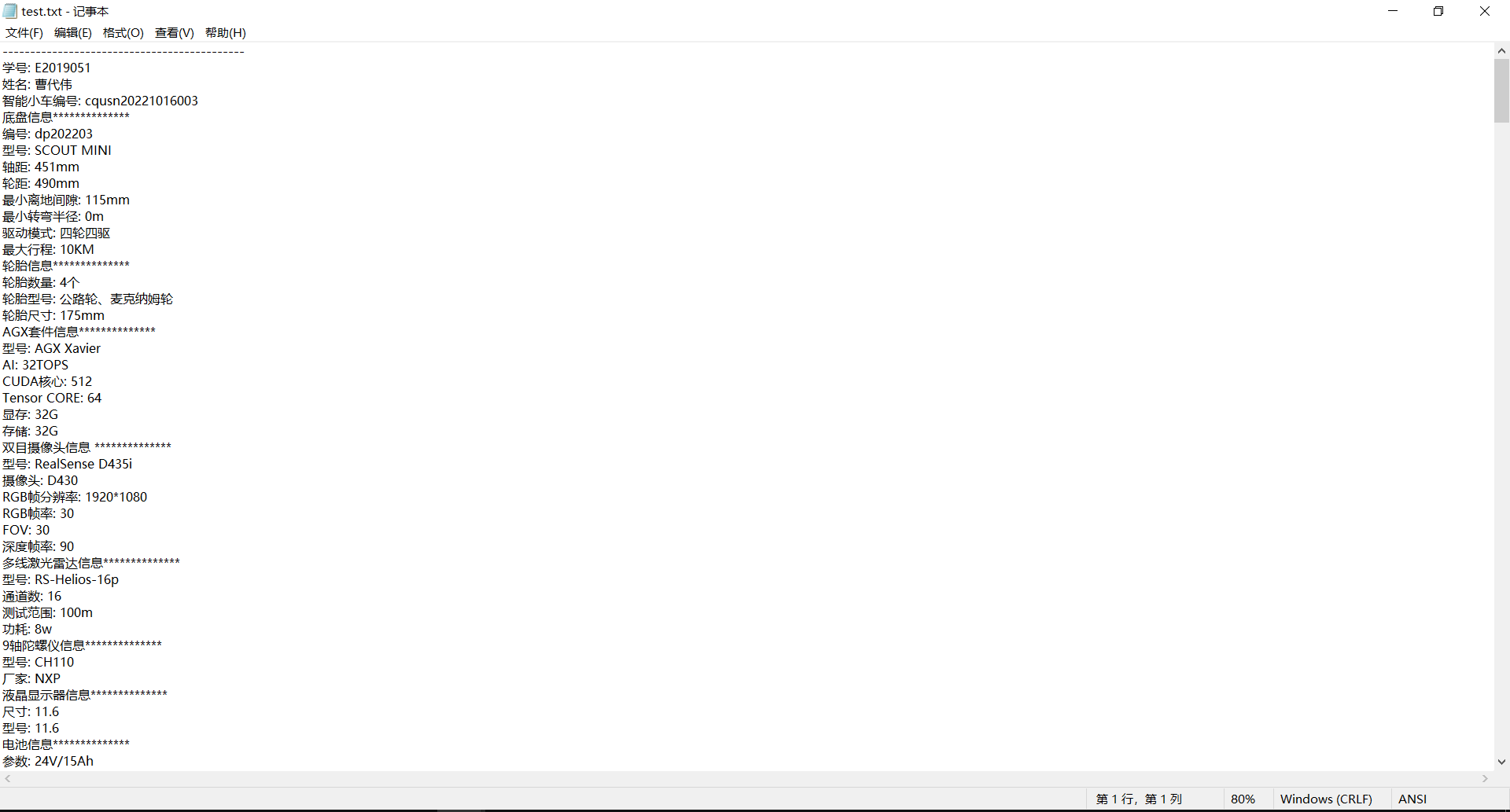
首先循环输入10辆智能小车编号及其底盘编号来完成小车对象的创建。



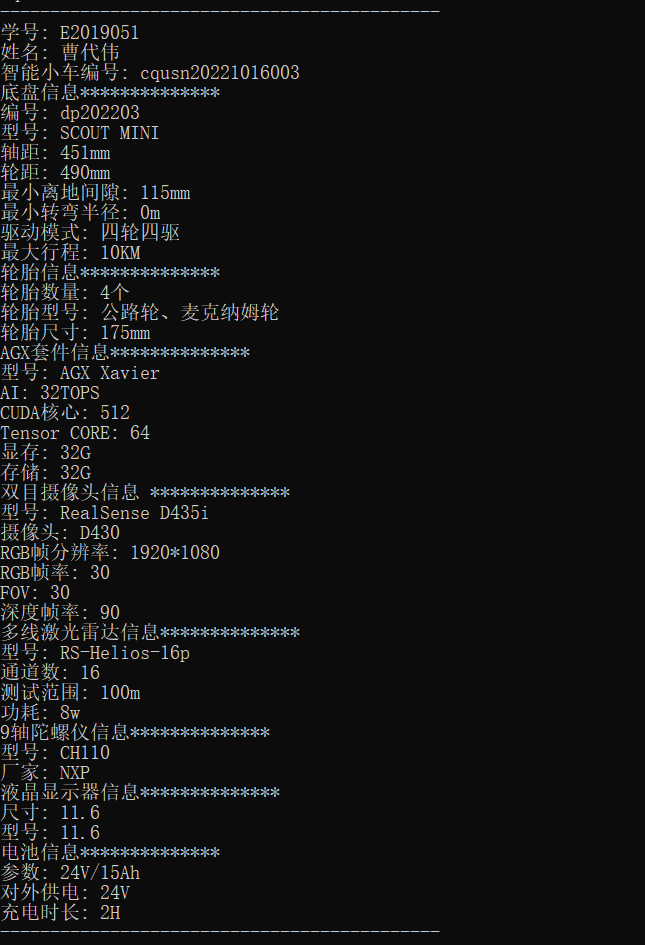
然后循环输入10位学生的学号及姓名，之后会展示目前未分配的小车编号，输入其中某一编号即可完成小车分配。



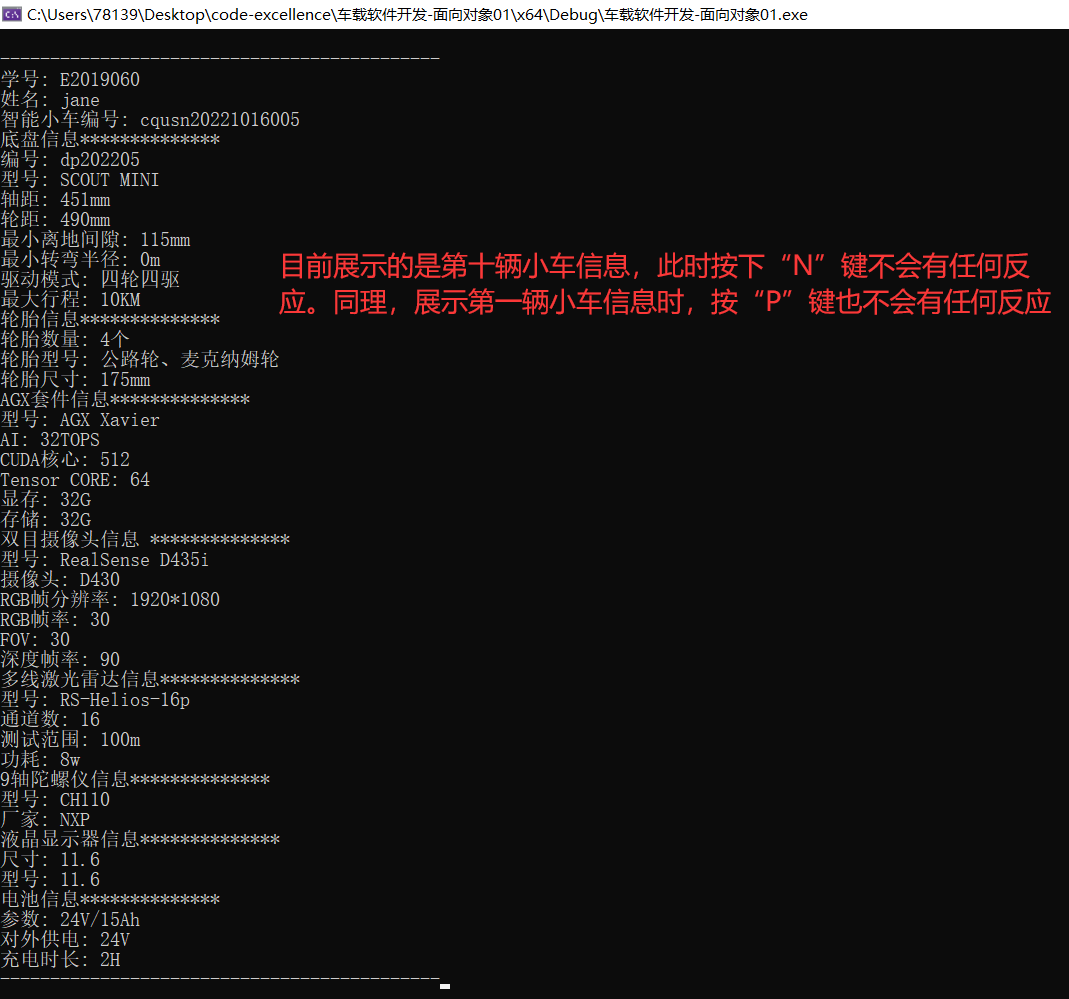
当10俩小车全部分配完毕后，所有学生信息及其分配的小车信息均会保存到test.txt文件中。



通过readData()函数读取之前保存至文件下信息，并输出为以下格式。展示的信息中，前三行显示最重要的学生学号和姓名，以及该学生拥有的智能小车编号。后面依次展示小车的具体详情，如底盘信息、Agx套件信息、轮胎信息等。



读取文件时默认展示第一辆小车的分配情况，此时程序持续监听用户的键盘输入，当用户按下“N”键时就会展示下一辆小车的分配情况，按下“P”键时则会展示上一辆小车的分配情况。当然，在展示第一辆小车时P键无效，展示最后一辆小车时N键无效。



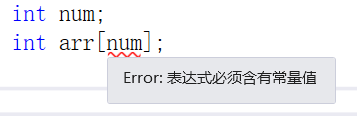
1. **总结和分析**

此次实验效果基本达到预期，能够通过程序录入小车信息、学生信息和分配信息并保存至硬盘中进行读写，较好地实现了报告所要求的各个功能。但是在编码过程中也遇到许多问题：

1.类设计阶段采用何种设计模式存在困惑，在学习面向对象编程时最重要三个特征就是封装、继承、多态。那么在类关系设计时第一反应便是采用继承，但经过分析与初步编码测试后，发现继承并不适合该题目要求，编写出的程序也很难达到复用的目的。因为车辆很明显是由底盘、轮胎、摄像头等部件组合而成，是部分与整体的关系，所以最后选择采用组合进行类关系设计，并顺利完成编码。

2.编码规范不达标，变量名、函数名等易混淆。根据C++编码谷歌标准，在进行C++编码时，类名应采用驼峰命名，普通变量应采用下划线命名，常量名应全大写，成员方法应见名知意，不能使用中文拼音。

3.不同开发环境、编译器版本可能在运行同一代码时会产生截然不同的结果。例如在visual studio2013开发环境中，由于编译器版本不支持动态数组。所以在定义数组时，其大小必须事先确定，即必须以常量进行初始化，而不能为变量。



1. **参考文献**

[1]C++ 组合 (Composition)模式详解.http://t.csdn.cn/XUqVc,2021-05-14.

[2]C/C++编写命名规范.http://t.csdn.cn/VzFcz,2022-05-21.

[3]C++ 保存数据的四种方式.http://t.csdn.cn/6g3xK,2021-03-20.

[4]C++面向对象设计原则详解.http://t.csdn.cn/KkfqZ,2020-03-29.

[5][美]Stephen Prata.C++ Primer plus[M].张海龙 袁国忠.北京:人民邮电出版社,2015:96-183.