

重庆大学

车载软件开发基础

课后实践 1



2022 至 2023 学年第 1 学期

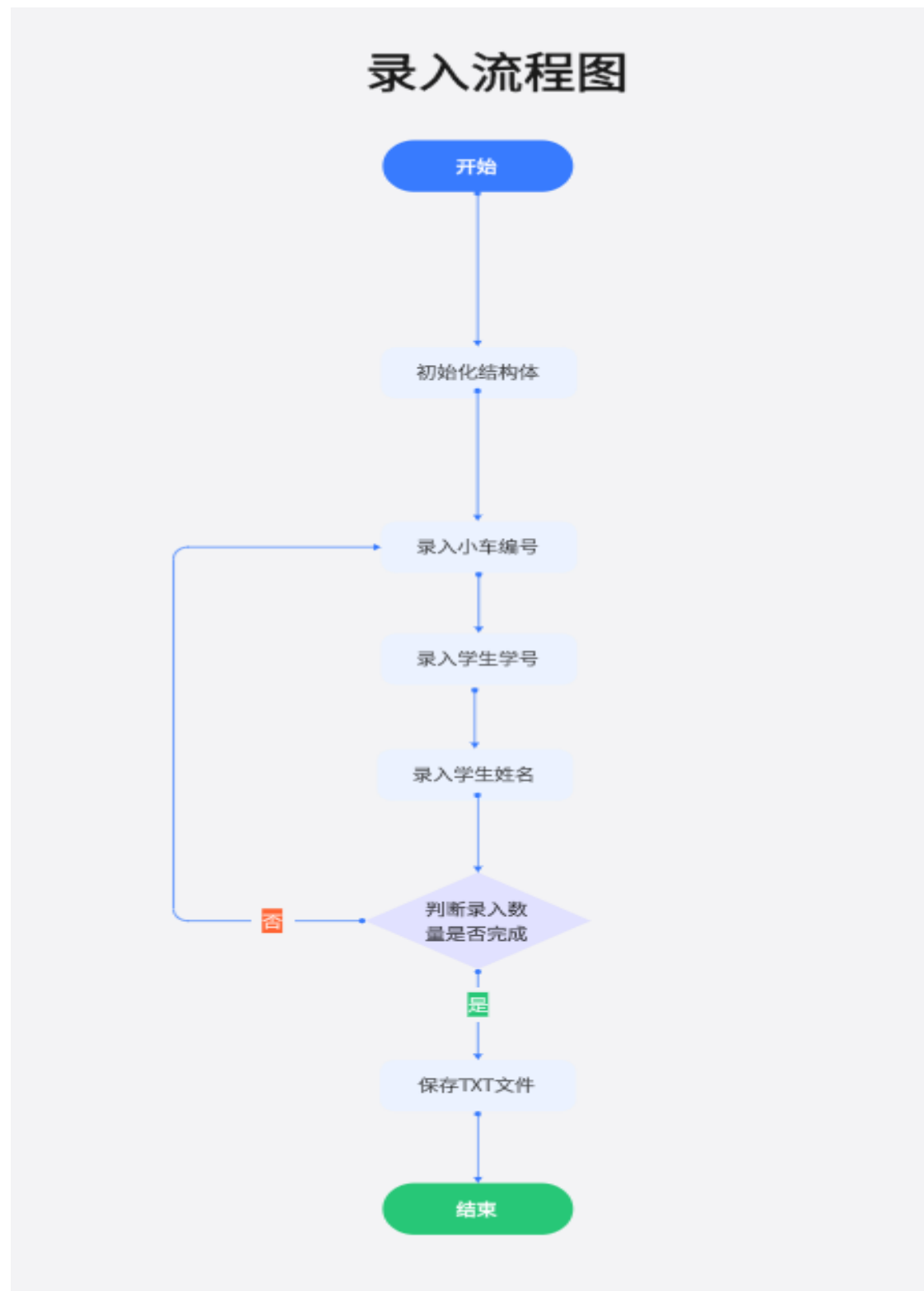
学号	姓名
E2019151	向洋
任课教师	刘骥
成 绩	

任务书	
任务内容	<p>某校购买了多台智能小车，并将其分配给同学们实验使用。请使用 C++语言编写一个程序，实现对这批智能小车的信息录入及分配，并以文件方式保存相关资料。具体资料如下：</p> <p>1、智能小车信息包括：</p> <p> (1) 编号：自定义，cqusn 打头的 16 位数字+字母</p> <p> (2) 底盘</p> <p> a) 编号：dp 打头的 8 位数字+字母</p> <p> b) 型号：SCOUT MINI</p> <p> c) 轴距：451mm</p> <p> d) 轮距：490mm</p> <p> e) 最小离地间隙：115mm</p> <p> f) 最小转弯半径：0m</p> <p> g) 驱动形式：四轮四驱</p> <p> h) 最大行程：10KM</p> <p> i) 轮胎（4 个）</p> <p> i. 型号：公路轮、麦克纳姆轮</p> <p> ii. 尺寸：175mm</p> <p> (3) AGX 套件（1 个）</p> <p> a) 型号：AGX Xavier</p> <p> b) AI：32 TOPS</p> <p> c) CUDA 核心：512</p> <p> d) Tensor CORE：64</p> <p> e) 显存：32G</p> <p> f) 存储：32G</p> <p> (4) 双目摄像头（1 个）</p> <p> a) 型号：RealSense D435i</p> <p> b) 摄像头：D430</p> <p> c) RGB 帧分辨率：1920*1080</p> <p> d) RGB 帧率：30</p> <p> e) FOV：87*58</p> <p> f) 深度帧率：90</p> <p> (5) 多线激光雷达（1 个）</p> <p> a) 型号：RS-Helios-16p</p> <p> b) 通道数：16</p> <p> c) 测试范围：100m</p> <p> d) 功耗：8W</p> <p> (6) 9 轴陀螺仪（1 个）</p> <p> a) 型号：CH110</p> <p> b) 厂家：NXP</p> <p> (7) 液晶显示屏（1 个）</p> <p> a) 尺寸：11.6</p>

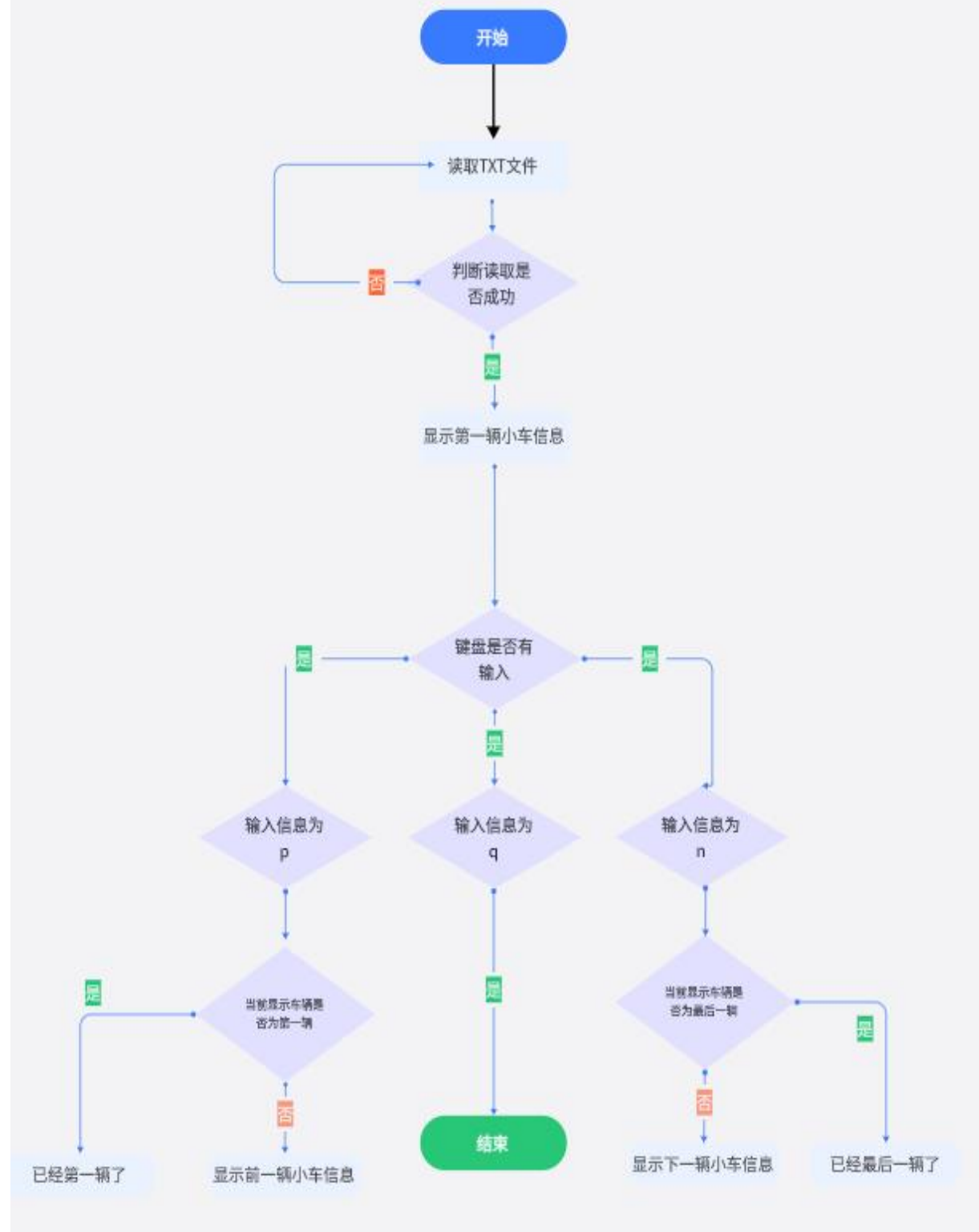
	<p>b) 型号: super</p> <p>(8) 电池模块 (1 个)</p> <p>a) 参数: 24V/15Ah</p> <p>b) 对外供电: 24V</p> <p>c) 充电时长: 2H</p> <p>2、学生信息包括:</p> <p>(1) 学号</p> <p>(2) 姓名</p> <p>要求:</p> <p>1、共采购了 10 台, 请先分别完成 10 台小车的信息录入, 并完成编号。</p> <p>2、根据编号, 将每台小车分配给每名同学 (含学号和姓名)</p> <p>3、将相关信息以文件方式保存到本地硬盘中</p> <p>4、用程序打开 3 中保存的文件, 然后将相关信息显示在屏幕上 (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。</p>
程序规范	<p>(1) 所有程序代码采用 C++ 编写;</p> <p>(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;</p> <p>(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;</p> <p>(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);</p>
报告要求	<p>(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;</p> <p>(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;</p> <p>(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;</p> <p>(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;</p> <p>(5) 报告应该格式规范、排版整洁、少语病和错误。</p>
作业提交	<p>(1) 完整源代码;</p> <p>(2) 任务报告。</p>
评分标准	<p>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:</p> <p>(1) 任务完成情况占评分的 60%;</p> <p>(2) 报告占评分的 40%。</p> <p>评分老师根据各部分的完成情况, 直接给出总成绩。</p>

一、程序设计

流程图



车辆信息显示



1. 定义 CAR 头文件。

```
#ifndef CAR_H
#define CAR_H
struct Car_list{
    string TrolleyNo;//小车编号
    string chassisnumber;//底盘编号
    string ChassisModel;//底盘型号
    string wheelbase;//底盘轴距
    string trackwidth;//底盘轮距
    string clearance;//最小离地间隙
    string turningradius;//最小转弯半径
    string Drivingform;//驱动形式
    string Maximumstroke;//最大行程
    string Tiretype;//轮胎型号
    string Tyresize;//轮胎尺寸
    string AGXmodel;//AGX 型号
    string AI;//AGX AI
    string Cudacore;//CUDA 核心
    string TensorCORE;//Tensor CORE
    string Videostorage;//显存
    string storage;//存储
    string cameramodel;//双目摄像头型号
    string camera;//摄像头
    string framerate;//RGB 帧分辨率
    string framerate;//RGB 帧率
    string FOV;//FOV
    string DepthFrameRate;//深度帧率
    string Lidarmodel;//多线激光雷达型号
    string channelNumber;//通道数
    string Testscope;//测试范围
    string powerwaste;//功耗
    string Gyroscopemodel;//陀螺仪型号
    string manufacturer;//厂家
    string Screensize;//液晶显示屏尺寸
    string LCDmodel;//液晶显示屏型号
    string Batteryparameters;//电池参数
    string Powersupply;//对外供电
    string Chargingduration;//充电时长
    string stu_id;//学生学号
    string stu_name;//学生姓名
};
#endif
```

2. 小车信息录入及保存为文本代码。

```
#include<string>
```

```
#include<iostream>
#include<fstream>
using namespace std;
#include "CAR.h"

#define car_total_number 10//定义本批次需要录入车辆的数量

int main(){
    int i=1;
    ofstream fout("Car_list.txt");//利用 ofstream 类的构造函数创建一个文件输出流对象来打开文件
    while(!cin.eof())//当 cin 没有处于流尾状态时，循环继续。ctrl+z 再按一次回车就能退出循环了。
    {

        string TrolleyNo;//小车编号
        string chassisnumber="dp888888";//底盘编号
        string ChassisModel="SCOUT_MINI";//底盘型号
        string wheelbase="451mm";//轴距
        string trackwidth="490mm";//轮距
        string clearance="115mm";//最小离地间隙
        string turningradius="0m";//最小转弯半径
        string Drivingform="四轮驱动";//驱动形式
        string Maximumstroke="10KM";//最大行程
        string Tiretype="公路轮、麦克纳姆轮";//轮胎型号
        string Tyresize="175mm";//轮胎尺寸
        string AGXmodel="AGX_Xavier";//AGX 型号
        string AI="32_TOPS";//AGX AI
        string Cudacore="512";//CUDA 核心
        string TensorCORE="64";//Tensor CORE
        string Videostorage="32G";//显存
        string storage="32G";//存储
        string cameramodel="RealSense_D435i";//双目摄像头型号
        string camera="D340";//摄像头
        string framerate="1920*1080";//RGB 帧分辨率
        string framerate="30";//RGB 帧率
        string FOV="87*58";//FOV
        string DepthFrameRate="90";//深度帧率
        string Lidarmodel="RS-Helios-16p";//多线激光雷达型号
        string channelNumber="16";//通道数
        string Testscope="100m";//测试范围
        string powerwaste="8W";//功耗
        string Gyroscopemodel="CH110";//陀螺仪型号
        string manufacturer="NXP";//厂家
```

```
string Screensize="11.6";//液晶显示屏尺寸
string LCDmodel="super";//液晶显示屏型号
string Batteryparameters="24V/15Ah";//电池参数
string Powersupply="24V";//对外供电
string Chargingduration="2H";//充电时长
```

```
string stu_id;//学生学号
string stu_name;//学生姓名
```

```
cout << "第" << i<< "次输入" << endl;
cout<<"(1)小车编号: "<<endl;
cin>>TrolleyNo;
cout<<"(3)学生学号: "<<endl;
cin>>stu_id;
cout<<"(2)学生姓名: "<<endl;
cin>>stu_name;
```

```
Car_list
```

```
s={stu_name,stu_id,TrolleyNo,chassisnumber,ChassisModel,wheelbase,track
width,clearance,turningradius,Drivingform,Maximumstroke,
    Tiretype,Tyresize,AGXmodel,AI,Cudacore,TensorCORE,Videostorage,s
torage,cameramodel,camera,framesresolution,framerate,
    FOV,DepthFrameRate,Lidarmodel,channelNumber,Testscope,powerwaste
,Gyroscopemodel,manufacturer,Screensize,LCDmodel,
    Batteryparameters,Powersupply,Chargingduration};
fout<<s.TrolleyNo<<" ";
fout<<s.chassisnumber<<" ";
fout<<s.ChassisModel<<" ";
fout<<s.wheelbase<<" ";
fout<<s.trackwidth<<" ";
fout<<s.clearance<<" ";
fout<<s.turningradius<<" ";
fout<<s.Drivingform<<" ";
fout<<s.Maximumstroke<<" ";
fout<<s.Tiretype<<" ";
fout<<s.Tyresize<<" ";
fout<<s.AGXmodel<<" ";
fout<<s.AI<<" ";
fout<<s.Cudacore<<" ";
fout<<s.TensorCORE<<" ";
fout<<s.Videostorage<<" ";
fout<<s.storage<<" ";
```



```
fout<<s.cameramodel<<" ";
fout<<s.camera<<" ";
fout<<s.framerresolution<<" ";
fout<<s.framerate<<" ";
fout<<s.FOV<<" ";
fout<<s.DepthFrameRate<<" ";
fout<<s.Lidarmodel<<" ";
fout<<s.channelNumber<<" ";
fout<<s.Testscope<<" ";
fout<<s.powerwaste<<" ";
fout<<s.Gyroscopemodel<<" ";
fout<<s.manufacturer<<" ";
fout<<s.Screensize<<" ";
fout<<s.LCDmodel<<" ";
fout<<s.Batteryparameters<<" ";
fout<<s.Powersupply<<" ";
fout<<s.Chargingduration<<" ";
fout<<s.stu_name<<" ";
fout<<s.stu_id<<endl;

i++;
if(i==(car_total_number+1))
{
    break;
}

}
fout.close();
}
```

3. 小车信息显示源代码。

```
#include<string>
#include<iostream>
#include<fstream>
#include<vector>
using namespace std;
#include "CAR.h"

vector<Car_list> v;
void printf_car_information(int i);

int main(){
    string TrolleyNo;
    int i;
    char z;
    ifstream fin("Car_list.txt");
    while (fin.peek()!=EOF)//peek()方法是读取文件指针下一个位置的值，但并不移动文件指针
    {
        Car_list s;
        fin>>s.stu_name>>s.stu_id>>s.TrolleyNo>>s.chassisnumber>>s.ChassisModel>>s.wheelbase>>s.trackwidth>>s.clearance>>s.turningradius;
        fin>>s.Drivingform>>s.Maximumstroke>>s.Tiretype>>s.Tyresize>>s.AGXmodel>>s.AI>>s.Cudacore>>s.TensorCORE;
        fin>>s.Videostorage>>s.storage>>s.cameramodel>>s.camera>>s.frame resolution>>s.framerate>>s.FOV>>s.DepthFrameRate>>s.Lidarmodel;
        fin>>s.channelNumber>>s.Testscope>>s.powerwaste>>s.Gyroscopemode l>>s.manufacturer>>s.Screensize>>s.LCDmodel;
        fin>>s.Batteryparameters>>s.Powersupply>>s.Chargingduration;
        v.push_back(s);
    }

    i=0;//初始化进入程序默认显示第一辆车辆信息
    printf_car_information(i);//输出第 i 辆小车信息

    while(0<=i<=9)
    {
        cout<<"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)"<<endl;
        cout<<"按 q 退出"<<endl;
        cin>>z;//键盘输入
        if(z=='p')
            if(i>=1&&i<10)
            {
```

```

        i--;
        printf_car_information(i); //输出第 i 辆小车信息

    }
    else
    {
        cout<<"已经是第一辆小车! "<<endl;
    }
    if(z=='n')
        if(i >=0&& i<9){
            i++;
            printf_car_information(i); //输出第 i 辆小车信息

        }
    else{
        cout<<"已经是最后一辆小车! "<<endl;
    }

    if(z=='q')
    {
        break;
    }
}

fin.close();
}

void printf_car_information(int i)
{
    cout<<"(1) 编号: "<<v[i].TrolleyNo<<endl;
    cout<<"(2) 底盘"<<endl;
    cout<<"    a) 编号:"<<v[i].chassisnumber<<endl;
    cout<<"    b) 型号:"<<v[i].ChassisModel<<endl;
    cout<<"    c) 轴距:"<<v[i].wheelbase<<endl;
    cout<<"    d) 轮距:"<<v[i].trackwidth<<endl;
    cout<<"    e) 最小离地间隙:"<<v[i].clearance<<endl;
    cout<<"    f) 最小转弯半径:"<<v[i].turningradius<<endl;
    cout<<"    g) 驱动形式:"<<v[i].Drivingform<<endl;
    cout<<"    f) 最大行程:"<<v[i].Maximumstroke<<endl;
    cout<<"    i) 轮胎(4 个)"<<endl;
    cout<<"        i. 型号:"<<v[i].Tiretype<<endl;
    cout<<"        ii. 尺寸:"<<v[i].Tyresize<<endl;
    cout<<"(3) AGX 套件 (1 个)"<<endl;
}

```

```

cout<<"    a)  型号:"<<v[i].AGXmodel<<endl;
cout<<"    b)  AI:"<<v[i].AI<<endl;
cout<<"    c)  CUDA 核心:"<<v[i].Cudacore<<endl;
cout<<"    d)  Tensor CORE:"<<v[i].TensorCORE<<endl;
cout<<"    e)  显存:"<<v[i].Videostorage<<endl;
cout<<"    f)  存储:"<<v[i].storage<<endl;
cout<<"(4)  双目摄像头 (1 个)"<<endl;
cout<<"    a)  型号:"<<v[i].cameramodel<<endl;
cout<<"    b)  摄像头:"<<v[i].camera<<endl;
cout<<"    c)  RGB 帧分辨率:"<<v[i].frameresolution<<endl;
cout<<"    d)  RGB 帧率:"<<v[i].framerate<<endl;
cout<<"    e)  FOV:"<<v[i].FOV<<endl;
cout<<"    f)  深度帧率:"<<v[i].DepthFrameRate<<endl;
cout<<"(5)  多线激光雷达 (1 个)"<<endl;
cout<<"    a)  型号:"<<v[i].Lidarmodel<<endl;
cout<<"    b)  通道数:"<<v[i].channelNumber<<endl;
cout<<"    c)  测试范围:"<<v[i].Testscope<<endl;
cout<<"    d)  功耗:"<<v[i].powerwaste<<endl;
cout<<"(6)  9 轴陀螺仪 (1 个)"<<endl;
cout<<"    a)  型号:"<<v[i].Gyroscopemodel<<endl;
cout<<"    b)  厂家:"<<v[i].manufacturer<<endl;
cout<<"(7)  液晶显示屏 (1 个)"<<endl;
cout<<"    a)  尺寸:"<<v[i].Screensize<<endl;
cout<<"    b)  型号:"<<v[i].LCDmodel<<endl;
cout<<"(8)  电池模块 (1 个)"<<endl;
cout<<"    a)  参数:"<<v[i].Batteryparameters<<endl;
cout<<"    b)  对外供电:"<<v[i].Powersupply<<endl;
cout<<"    c)  充电时间:"<<v[i].Chargingduration<<endl;
cout<<"学生姓名: "<<v[i].stu_name<<" ,学生学号:
"<<v[i].stu_id<<endl;

}

```

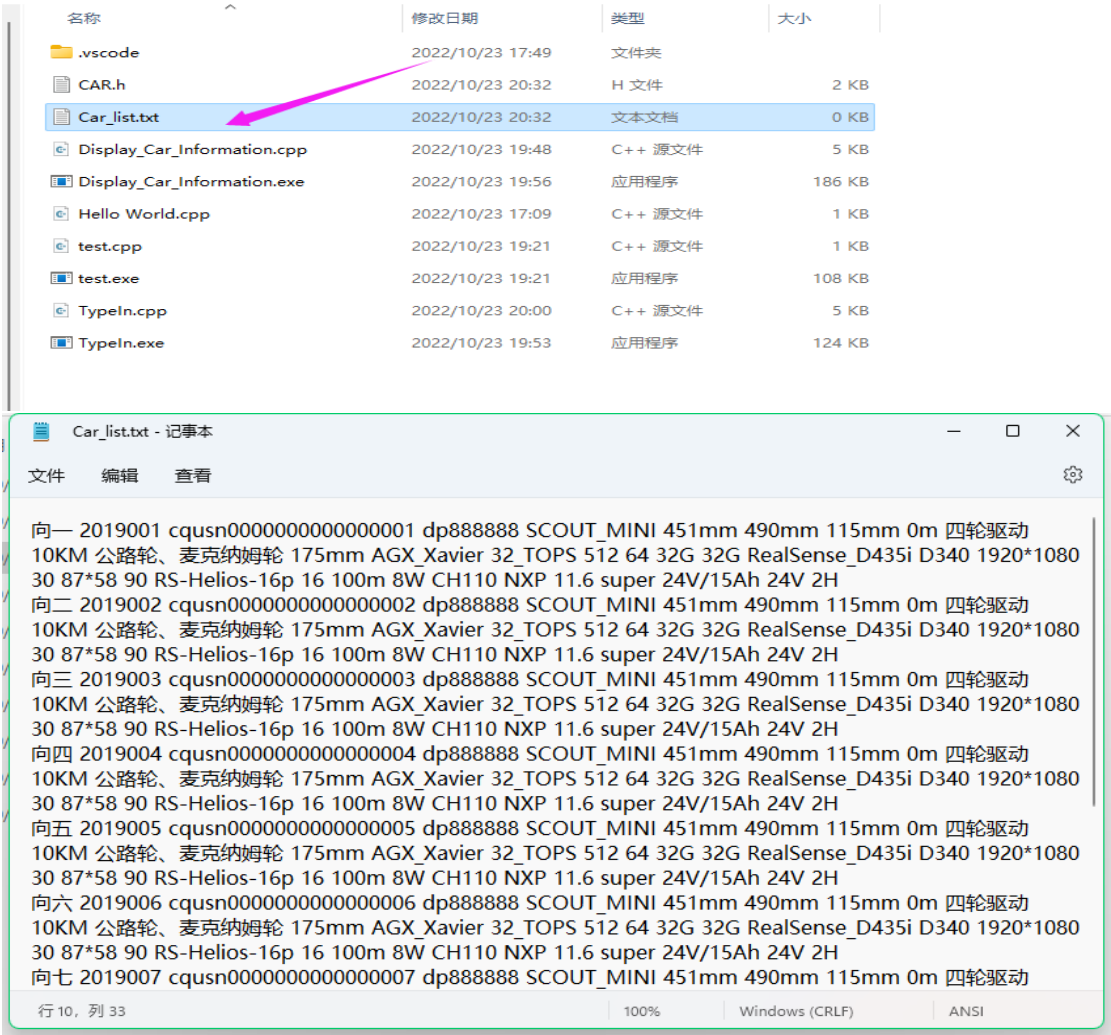
二、程序效果展示

1. 执行程序 TypeIn. cpp。依次录入小车的编号，以及被分派到的学生的学号和姓名。



```
e:\CAR\TypeIn.exe
第1次输入
(1)小车编号:
cqusn0000000000000001
(3)学生学号:
2019001
(2)学生姓名:
向一
第2次输入
(1)小车编号:
```

2. 录入到达设定本批次需要录入车辆的数量，即第十次录入后，自动退出。生成“Car_list.txt”文本，并将输入录入信息保存。



名称	修改日期	类型	大小
.vscode	2022/10/23 17:49	文件夹	
CAR.h	2022/10/23 20:32	H 文件	2 KB
Car_list.txt	2022/10/23 20:32	文本文档	0 KB
Display_Car_Information.cpp	2022/10/23 19:48	C++ 源文件	5 KB
Display_Car_Information.exe	2022/10/23 19:56	应用程序	186 KB
Hello World.cpp	2022/10/23 17:09	C++ 源文件	1 KB
test.cpp	2022/10/23 19:21	C++ 源文件	1 KB
test.exe	2022/10/23 19:21	应用程序	108 KB
TypeIn.cpp	2022/10/23 20:00	C++ 源文件	5 KB
TypeIn.exe	2022/10/23 19:53	应用程序	124 KB

Car_list.txt - 记事本

文件 编辑 查看

向一 2019001 cqusn0000000000000001 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向二 2019002 cqusn0000000000000002 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向三 2019003 cqusn0000000000000003 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向四 2019004 cqusn0000000000000004 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向五 2019005 cqusn0000000000000005 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向六 2019006 cqusn0000000000000006 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动
10KM 公路轮、麦克纳姆轮 175mm AGX_Xavier 32_TOPS 512 64 32G 32G RealSense_D435i D340 1920*1080
30 87*58 90 RS-Helios-16p 16 100m 8W CH110 NXP 11.6 super 24V/15Ah 24V 2H
向七 2019007 cqusn0000000000000007 dp888888 SCOUT_MINI 451mm 490mm 115mm 0m 四轮驱动

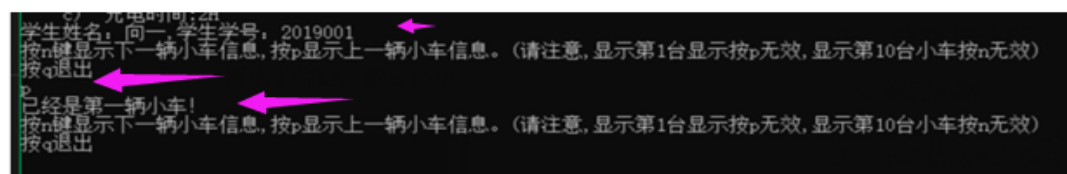
行 10, 列 33 | 100% | Windows (CRLF) | ANSI

3. 执行程序 “Display_Car_Information.cpp”，默认从保存的第一辆小车开始显示，按 n 可查看下一辆小车信息，按 p 可查看上一辆小车信息，按 q 结束。

The screenshot shows a Windows application window titled "e:\CAR\Display_Car_Information.exe". The application displays a list of car specifications for the first car in the database. The specifications are numbered (1) through (8) and include details like chassis, AGX components, cameras, radar, lidar, and battery. At the bottom, it prompts the user to press 'n' for the next car, 'p' for the previous car, or 'q' to exit. The first car's ID is cqsun000000000000000001.

```
(1) 编号: cqsun000000000000000001
(2) 底盘
a) 编号: dp888888
b) 型号: SCOUT_MINI
c) 轴距: 451mm
d) 轮距: 490mm
e) 最小离地间隙: 115mm
f) 最小转弯半径: 0m
g) 驱动形式: 四轮驱动
h) 最大行程: 10KM
i) 轮胎 (4个)
    i. 型号: 公路轮、麦克纳姆轮
    ii. 尺寸: 175mm
(3) AGX零件 (1个)
a) 型号: AGX_Xavier
b) AI: 32_TOPS
c) CUDA核心: 512
d) Tensor CORE: 64
e) 显存: 32G
f) 存储: 32G
(4) 双目摄像头 (1个)
a) 型号: RealSense_D435i
b) 摄像头: D340
c) RGB帧分辨率: 1920*1080
d) RGB帧率: 30
e) FOV: 87*58
f) 深度帧率: 90
(5) 多线激光雷达 (1个)
a) 型号: RS-Helios-16p
b) 通道数: 16
c) 测试范围: 100m
d) 功耗: 8W
(6) 9轴陀螺仪 (1个)
a) 型号: CH110
b) 厂家: NXP
(7) 液晶显示屏 (1个)
a) 尺寸: 11.6
b) 型号: super
(8) 电池模块 (1个)
a) 参数: 24V/15Ah
b) 对外供电: 24V
c) 充电时间: 2H
学生姓名: 向一, 学生学号: 2019001
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)
按q退出
p
已经是第一辆小车!
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)
按q退出
```

在第一辆小车时，按 p 查看上一辆小车信息，则提示无效



按 n 查看下一辆小车信息

The screenshot shows the application window after pressing 'n'. It now displays the specifications for the second car in the database, with ID cqsun000000000000000002. The specifications are identical in format to the first car's. The prompt at the bottom now indicates that pressing 'p' will show the previous car (the first car).

```
(1) 编号: cqsun000000000000000002
(2) 底盘
a) 编号: dp888888
b) 型号: SCOUT_MINI
c) 轴距: 451mm
d) 轮距: 490mm
e) 最小离地间隙: 115mm
f) 最小转弯半径: 0m
g) 驱动形式: 四轮驱动
h) 最大行程: 10KM
i) 轮胎 (4个)
    i. 型号: 公路轮、麦克纳姆轮
    ii. 尺寸: 175mm
(3) AGX零件 (1个)
a) 型号: AGX_Xavier
b) AI: 32_TOPS
c) CUDA核心: 512
d) Tensor CORE: 64
e) 显存: 32G
f) 存储: 32G
(4) 双目摄像头 (1个)
a) 型号: RealSense_D435i
b) 摄像头: D340
c) RGB帧分辨率: 1920*1080
d) RGB帧率: 30
e) FOV: 87*58
f) 深度帧率: 90
(5) 多线激光雷达 (1个)
a) 型号: RS-Helios-16p
b) 通道数: 16
c) 测试范围: 100m
d) 功耗: 8W
(6) 9轴陀螺仪 (1个)
a) 型号: CH110
b) 厂家: NXP
(7) 液晶显示屏 (1个)
a) 尺寸: 11.6
b) 型号: super
(8) 电池模块 (1个)
a) 参数: 24V/15Ah
b) 对外供电: 24V
c) 充电时间: 2H
学生姓名: 向二, 学生学号: 2019002
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)
按q退出
```

按 p 查看上一辆小车信息

```
e:\CAR\Display_Car_Information.exe
(8) 电池模块 (1个)
  a) 参数:24V/15Ah
  b) 对外供电:24V
  c) 充电时间:2H
学生姓名: 向六, 学生学号: 2019006
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)
按q退出
p
(1) 编号: cqusr0000000000000005
(2) 底盘
  a) 编号: dp888888
  b) 型号: SCOUT_MINI
  c) 轴距: 451mm
  d) 轮距: 490mm
  e) 最小离地间隙: 115mm
  f) 最小转弯半径: 0m
  g) 驱动形式: 四轮驱动
  h) 最大行程: 10KM
  i) 轮胎 (4个)
    i. 型号: 公路轮、麦克纳姆轮
    ii. 尺寸: 175mm
(3) AGX套件 (1个)
  a) 型号: AGX_Xavier
  b) AI: 32_TOPS
  c) CUDA核心: 512
  d) Tensor CORE: 64
  e) 显存: 32G
  f) 存储: 32G
(4) 双目摄像头 (1个)
  a) 型号: RealSense_D435i
```

在第十辆小车时，按 n 查看下一辆小车信息，则提示无效

```
e:\CAR\Display_Car_Information.exe
(4) 双目摄像头 (1个)
  a) 型号: RealSense_D435i
  b) 摄像头: D340
  c) RGB帧分辨率: 1920*1080
  d) RGB帧率: 30
  e) FOV: 87*58
  f) 深度帧率: 90
(5) 多线激光雷达 (1个)
  a) 型号: RS-Helios-16p
  b) 通道数: 16
  c) 测试范围: 100m
  d) 功耗: 3W
(6) 陀螺仪 (1个)
  a) 型号: CH110
  b) 厂家: NXP
(7) 液晶显示屏 (1个)
  a) 尺寸: 11.6
  b) 型号: super
(8) 电池模块 (1个)
  a) 参数: 24V/15Ah
  b) 对外供电: 24V
  c) 充电时间: 2H
学生姓名: 向十, 学生学号: 2019010
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)
按q退出
n
已经是最后一辆小车!
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)
按q退出
```

三、总结与分析

一、编程思路

1. 头文件中定义小车结构体，将小车参数进行定义；
2. 小车信息录入和学生分配，车辆信息显示，分为两个独立的部分。降低程序复杂度；
3. 同批次小车物理参数相同，固定的参数直接定义并赋值，然后写入到 txt 文件中，并保存。
4. 车辆信息显示程序读取保存的 txt 文件，对应赋值，然后按照给定的格式输出显示小车及学生的信息，然后使用 while 循环，检测键盘输入信息，实现“上一辆、下一辆”小车信息显示。

二、一些问题与改进

1. 题目要求将采购的 10 台小车进行信息录入，并将其分配给同学。在定义结构体后，发现同批次小车，大部分物理参数都是相同，无须重复录入。相同信息我们在初始化程序时直接写好，我们只需要录入每辆小车不同的信息即可。

```
string TensorCORE="64";//Tensor CORE
string Videostorage="32G";//显存
string storage="32G";//存储
string cameramodel="RealSense_D435i";//双目摄像头型号
string camera="D340";//摄像头
string framerate="1920*1080";//RGB帧分辨率
string framerate="30";//RGB帧率
string FOV="87*58";//FOV
string DepthFrameRate="90";//深度帧率
string Lidarmodel="RS-Helios-16p";//多线激光雷达型号
string channelNumber="16";//通道数
string Testscope="100m";//测试范围
string powerwaste="8W";//功耗
string Gyroscopemodel="CH110";//陀螺仪型号
string manufacturer="NXP";//厂家
string Screensize="11.6";//液晶显示屏尺寸
string LCDmodel="super";//液晶显示屏型号
string Batteryparameters="24V/15Ah";//电池参数
string Powersupply="24V";//对外供电
string Chargingduration="2H";//充电时长

string stu_id;//学生学号
string stu_name;//学生姓名
```

相同信息

```
cout << "第" << i << "次输入" << endl;
cout<<"(1)小车编号: "<<endl;
cin>>TrolleyNo;
cout<<"(3)学生学号: "<<endl;
cin>>stu_id;
cout<<"(2)学生姓名: "<<endl;
cin>>stu_name;
```

车辆不同信息

2. 在读取 txt 文件中的内容时，出现输出一行空行。
读文件时，用 while (!fin.eof()) 结束会导致输出一行空行，

```
9 void printf_car_information(int i);
10
11 int main(){
12     string TrolleyNo;
13     int i;
14     char z;
15     ifstream fin("Car_list.txt");
16     while (!fin.eof())
17     {
18         Car_list s;
19         fin>>s.stu_name>>s.stu_id>>s.TrolleyNo>>s.chassisnumber>>s.ChassisModel>>s.
20         fin>>s.Drivingform>>s.Maximumstroke>>s.Tiretype>>s.Tyresize>>s.AGXmodel>>s.
21         fin>>s.Videostorage>>s.storage>>s.cameramodel>>s.camera>>s.framesolution>
22         fin>>s.channelNumber>>s.Testscope>>s.powerwaste>>s.Gyroscopemodel>>s.manufa
23         fin>>s.Batteryparameters>>s.Powersupply>>s.Chargingduration;
24         v.push_back(s);
25     }
26
27     i=0;//初始化进入程序默认显示第一辆车辆信息
28     printf_car_information(i);//输出第1辆小车信息
29
30     while(0<=i<=9)
31     {
32         cout<<"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
33         cout<<"按q退出"<<endl;
34         cin>>z;//键盘输入
35         if(z=='p')
36             if(i>=1&&i<10)
37             {
38                 i--;
39             }
40         if(z=='n')
41             if(i<9)
42             {
43                 i++;
44             }
45         printf_car_information(i);
46     }
47 }
```

选择 e:\CAR\Display_Car_Information.exe

```
(1) 编号: <quasi000000000000000001
(2) 底盘
a) 编号: dp888888
b) 型号: SCOUT_MINI
c) 轴距: 451mm
d) 轮距: 490mm
e) 最小离地间隙: 115mm
f) 最小转弯半径: 0m
g) 驱动形式: 四轮驱动
h) 最大行程: 10KM
i) 轮胎 (4个)
    i. 型号: 公路轮、麦克纳姆轮
    ii. 尺寸: 175mm
(3) AGX属性 (1个)
a) 型号: AGX_Xavier
b) AI: 32_TOFS
c) CUDA核心: 512
d) Tensor CORE: 64
e) 显存: 32G
f) 存储: 32G
(4) 双目摄像头 (1个)
a) 型号: RealSense_D435i
b) 摄像头: D340
c) RGB帧分辨率: 1920*1080
d) RGB帧率: 30
e) FOV: 87*58
f) 深度帧率: 90
(5) 多线程雷达 (1个)
a) 型号: RS-Helios-16p
b) 通道数: 16
```

因为文件指针到最后一个字符时并不会触发 eof, 再读一次读不到数据才触发 eof, 这样 c 还保留着上一次的数据, 又被输出一次, peek() 方法是读取文件指针下一个位置的值, 但并不移动文件指针:

改为:

while (fin.peek() != EOF)

```
9 void printf_car_information(int i);
10
11 int main(){
12     string TrolleyNo;
13     int i;
14     char z;
15     ifstream fin("Car_list.txt");
16     while (fin.peek() != EOF) // peek()方法是读取文件指针下一个位置的值, 但并不移动文件指针
17     {
18         Car_list s;
19         fin>>s.stu_name>>s.stu_id>>s.TrolleyNo>>s.chassisnumber>>s.ChassisModel>>s.
20         fin>>s.Drivingform>>s.Maximumstroke>>s.Tiretype>>s.Tyresize>>s.AGXmodel>>s.
21         fin>>s.Videostorage>>s.storage>>s.cameramodel>>s.camera>>s.framesolution>
22         fin>>s.channelNumber>>s.Testscope>>s.powerwaste>>s.Gyroscopemodel>>s.manufa
23         fin>>s.Batteryparameters>>s.Powersupply>>s.Chargingduration;
24         v.push_back(s);
25     }
26
27     i=0;//初始化进入程序默认显示第一辆车辆信息
28     printf_car_information(i);//输出第1辆小车信息
29
30     while(0<=i<=9)
31     {
32         cout<<"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
33         cout<<"按q退出"<<endl;
34         cin>>z;//键盘输入
35         if(z=='p')
36             if(i>=1&&i<10)
37             {
38                 i--;
39             }
40         if(z=='n')
41             if(i<9)
42             {
43                 i++;
44             }
45         printf_car_information(i);
46     }
47 }
```

选择 e:\CAR\Display_Car_Information.exe

```
(1) 编号: <quasi000000000000000001
(2) 底盘
a) 编号: dp888888
b) 型号: SCOUT_MINI
c) 轴距: 451mm
d) 轮距: 490mm
e) 最小离地间隙: 115mm
f) 最小转弯半径: 0m
g) 驱动形式: 四轮驱动
h) 最大行程: 10KM
i) 轮胎 (4个)
    i. 型号: 公路轮、麦克纳姆轮
    ii. 尺寸: 175mm
(3) AGX属性 (1个)
a) 型号: AGX_Xavier
b) AI: 32_TOFS
c) CUDA核心: 512
d) Tensor CORE: 64
e) 显存: 32G
f) 存储: 32G
(4) 双目摄像头 (1个)
a) 型号: RealSense_D435i
b) 摄像头: D340
c) RGB帧分辨率: 1920*1080
d) RGB帧率: 30
e) FOV: 87*58
f) 深度帧率: 90
(5) 多线程雷达 (1个)
a) 型号: RS-Helios-16p
b) 通道数: 16
```

3. 对于重复遇到的输出操作语句，我们进行对其进行函数封包操作。

```
1 #include<string>
2 #include<iostream>
3 #include<fstream>
4 #include<vector>
5 using namespace std;
6 #include "CAR.h"
7
8 vector<Car_list> v;
9 void printf_car_information(int i);
10
11 int main(){
12     string TrolleyNo;
13     int i;
14     char z;
15     ifstream fin("Car_list.txt");
16     while (fin.peek() != EOF) // peek()方法是读取文件指针下一个位置的值, 但并不移动文件指针
17     {
18         Car_list s;
19         fin>>s.stu_name>>s.stu_id>>s.TrolleyNo>>s.chassisnumber>>s.ChassisModel>>s.
20         fin>>s.Drivingform>>s.Maximumstroke>>s.Tiretype>>s.Tyresize>>s.AGXmodel>>s.
21         fin>>s.Videostorage>>s.storage>>s.cameramodel>>s.camera>>s.framesolution>
22         fin>>s.channelNumber>>s.Testscope>>s.powerwaste>>s.Gyroscopemodel>>s.manufa
23         fin>>s.Batteryparameters>>s.Powersupply>>s.Chargingduration;
24         v.push_back(s);
25     }
26
27     i=0;//初始化进入程序默认显示第一辆车辆信息
28     printf_car_information(i);//输出第1辆小车信息
29
30     while(0<=i<=9)
31     {
32         cout<<"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
33         cout<<"按q退出"<<endl;
34         cin>>z;//键盘输入
35         if(z=='p')
36             if(i>=1&&i<10)
37             {
38                 i--;
39             }
40         if(z=='n')
41             if(i<9)
42             {
43                 i++;
44             }
45         printf_car_information(i);
46     }
47 }
```

显示输出函数提前声明

```

63     fin.close();
64 }
65
66 void printf_car_information(int i)
67 {
68
69     cout<<"(1) 编号: "<<v[i].TrolleyNo<<endl;
70     cout<<"(2) 底盘"<<endl;
71     cout<<"    a) 编号: "<<v[i].chassisnumber<<endl;
72     cout<<"    b) 型号: "<<v[i].ChassisModel<<endl;
73     cout<<"    c) 轴距: "<<v[i].wheelbase<<endl;
74     cout<<"    d) 轮距: "<<v[i].trackwidth<<endl;
75     cout<<"    e) 最小离地间隙: "<<v[i].clearance<<endl;
76     cout<<"    f) 最小转弯半径: "<<v[i].turningradius<<endl;
77     cout<<"    g) 驱动形式: "<<v[i].Drivingform<<endl;
78     cout<<"    f) 最大行程: "<<v[i].Maximumstroke<<endl;
79     cout<<"    i) 轮胎(4个)"<<endl;
80     cout<<"        i. 型号: "<<v[i].Tiretype<<endl;
81     cout<<"        ii. 尺寸: "<<v[i].Tyresize<<endl;
82     cout<<"(3) AGX套件 (1个)"<<endl;
83     cout<<"    a) 型号: "<<v[i].AGXmodel<<endl;
84     cout<<"    b) AI: "<<v[i].AI<<endl;
85     cout<<"    c) CUDA核心: "<<v[i].Cudacore<<endl;
86     cout<<"    d) Tensor CORE: "<<v[i].TensorCORE<<endl;
87     cout<<"    e) 显存: "<<v[i].Videostorage<<endl;
88     cout<<"    f) 存储: "<<v[i].storage<<endl;
89     cout<<"(4) 双目摄像头 (1个)"<<endl;
90     cout<<"    a) 型号: "<<v[i].cameramodel<<endl;
91     cout<<"    b) 摄像头: "<<v[i].camera<<endl;
92     cout<<"    c) RGB帧分辨率: "<<v[i].framesresolution<<endl;
93     cout<<"    d) RGB帧率: "<<v[i].framerate<<endl;
94     cout<<"    e) FOV: "<<v[i].FOV<<endl;
95     cout<<"    f) 深度帧率: "<<v[i].DepthFrameRate<<endl;
96     cout<<"(5) 多线激光雷达 (1个)"<<endl;

```

四、参考文献:

- [1] 老师课堂示例代码;
- [2] C++输入与输出与输出变量的值, <http://t.csdn.cn/RIJwf>
- [3] 三种循环语句的详解和使用, <http://t.csdn.cn/SKk91>
- [4] C++文件的读取和写入, <http://t.csdn.cn/18Guz>
- [5] C++学习之重载, <http://t.csdn.cn/7e0xM>
- [6] 数组和 vector 区别, <https://www.cnblogs.com/love-yh/p/7410666.html>
- [7] c++学习笔记之加法运算符重载, <http://t.csdn.cn/6Nrvv>