

车载软件开发基础

课后实践2



**2022至2023学年第 1 学期**

|  |  |
| --- | --- |
| 学号 | 姓名 |
| 2110607120 | 彭星宇 |
| 任课教师 | 刘骥 |
| 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。 |

**正文**

[**GitHub**](https://github.com/pxy522/Cqu_EIE_projects/tree/main/Object-Oriented)

1.程序设计：

1. ：程序采用面向对象设计，核心类和模块之间相互协作，主要类与函数如下：

（a）：Car 类：代表一辆智能小车，包含多个组成部分（如底盘、AGX套件、摄像头、激光雷达等）的信息。

（b）：其他与小车硬件相关的类：Chassis（底盘）、AGXKit、StereoCamera（立体摄像头）、Lidar（激光雷达）、Gyroscope（陀螺仪）、LCD（显示屏）、Battery（电池）。

（c）：utils 模块：提供一些工具函数（从文件读取信息，更改信息）。

（2）：主要功能和算法

（a）：菜单选择：程序进入一个无限循环，通过 switch 语句根据用户输入执行不同操作。操作包括：

1. 录入车辆信息：调用 Car::inputCarInfo() 方法录入每辆车的详细信息。

2. 输出车辆信息：展示车辆基本信息，并可进一步查看详细信息（底盘、摄像头、雷达等部件的状态）。

3. 修改车辆信息（详细代码尚未查看）。

4. 保存车辆信息到文件。

5. 从文件中读取车辆信息。

（3）：函数与模块关系

（a）：main() 函数通过一个 vector<Car> 来管理所有车辆。每辆车对象通过各自的成员函数打印、修改信息。车辆的详细信息分布在多个类中，每个类都负责自己对应部件的逻辑与输出。

2.程序效果展示：

程序提供了一个交互式命令行界面，用户可以通过数字选择执行各项操作。例如，选择“1”录入车辆信息、选择“2”查看车辆信息。程序可以展示每辆车的基本信息，还能详细展示每个部件的状态，如底盘、摄像头、激光雷达等。

1. 程序运行时截图

入口：

文本

描述已自动生成

设置学号和姓名并通过当前时间生成车辆编号和底盘编号：

图片包含 文本

描述已自动生成

可以选择查看完整信息：

图片包含 文本

描述已自动生成

保存成json文件:

文本

描述已自动生成

从json文件中读取：

文本

描述已自动生成

根据车辆ID修改车辆信息：

文本

描述已自动生成

Json文件：

电脑萤幕截图

描述已自动生成

3. 总结与分析

（1）： **实现效果**：程序成功实现了车辆信息的管理功能，并能够通过交互式菜单提供录入、展示、修改、保存等功能。

（2）： **差异分析**：如果存在差异，可能出现在信息展示的美观性与文件存取的灵活性方面。程序在某些地方可能缺乏错误处理机制，例如未对输入错误做详细处理。某些复杂操作（如信息的修改和保存）可能还需要进一步完善。美观问题可以通过加入命令行代码，美化终端和输出已达到更好的交互效果。