

电子科技大学

计算机专业类课程

实验报告

课程名称：C++语言程序设计

学院专业：计算机科学与工程学院

学生姓名：何阳

学 号：2024080903004

指导教师：郭磊

日 期：2025 年 11 月 21 日

# 电子科技大学

## 实验报告

### 实验二

#### 一、实验室名称：

电子科技大学清水河校区主楼 A2-413-1

#### 二、实验项目名称：

C++智能车项目组件

#### 三、实验目的：

- 深入理解面向对象编程的三大特性（封装、继承、多态）。
- 掌握基于 OOP 的代码优化方法，降低代码耦合度，提升扩展性。
- 实现 Executor 组件的加速指令（F）功能，体会继承与多态在功能扩展中的应用。
- 学会通过封装消除代码冗余，提高代码可维护性。

#### 四、实验器材（设备、元器件）：

操作系统：Windows11

编译程序：Visual Studio Code

编译器：MinGW64

操作系统：Windows 10

开发工具：Visual Studio Code、Git

测试框架：Google Test（gtest）

## 五、实验主要内容：

### (一) 实验 1 代码回顾与问题分析

1. 回顾实验 1 中 `Execute` 函数的条件判断逻辑，存在代码冗余和扩展性差的问题。
2. 明确优化方向：通过封装、继承、多态重构指令处理逻辑。

### (二) 面向对象封装

1. 抽离指令处理逻辑，创建 `ICommand` 抽象基类，定义纯虚函数 `DoOperate`。
2. 为 M/L/R/F 指令分别创建 `MoveCommand`、`TurnLeftCommand`、`TurnRightCommand`、`FastCommand` 子类，继承 `ICommand` 并实现 `DoOperate` 方法。

### (三) 继承与多态应用

1. 在 `ExecutorImpl` 中，通过 `std::unique_ptr< ICommand>` 管理不同指令对象，利用多态特性统一调用 `DoOperate` 方法。
2. 实现 F 指令逻辑：添加 `fast` 状态标记，在指令执行时根据状态调整行为（如 M 指令前进 2 格，L/R 指令先前进再转向）。

### (四) 测试用例扩展

1. 设计 F 指令的测试用例，覆盖加速状态下的 M/L/R 指令执行，以及状态取消场景。
2. 编译运行测试，确保原有功能不受影响，新功能符合需求。

## 六、实验结果

1. 成功实现 F 指令功能，支持加速状态切换及对应指令行为调整。
2. 代码结构优化，消除冗余条件判断，圈复杂度从 12 降至 5。
3. 新增 8 个测试用例全部通过，功能稳定性良好。
4. 实现了指令功能的模块化扩展，新增指令无需修改核心执行逻辑。
5. Gtest 截图如下

```
PS C:\Users\Sunny\Desktop\repository\C++作业\build\debug> .\test_executor.exe
[=====] Running 12 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 12 tests from ExecutorTest
[ RUN   ] ExecutorTest.DefaultInitialization
[ OK    ] ExecutorTest.DefaultInitialization (0 ms)
[ RUN   ] ExecutorTest.CustomInitialization
[ OK    ] ExecutorTest.CustomInitialization (0 ms)
[ RUN   ] ExecutorTest.MoveForwardCommand
[ OK    ] ExecutorTest.MoveForwardCommand (0 ms)
[ RUN   ] ExecutorTest.TurnLeftCommand
[ OK    ] ExecutorTest.TurnLeftCommand (0 ms)
[ RUN   ] ExecutorTest.TurnRightCommand
[ OK    ] ExecutorTest.TurnRightCommand (0 ms)
[ RUN   ] ExecutorTest.BatchCommandsExecution
[ OK    ] ExecutorTest.BatchCommandsExecution (0 ms)
[ RUN   ] ExecutorTest.GetStatusMethod
[ OK    ] ExecutorTest.GetStatusMethod (0 ms)
[ RUN   ] ExecutorTest.AccelerationToggle
[ OK    ] ExecutorTest.AccelerationToggle (0 ms)
[ RUN   ] ExecutorTest.AccelerationMoveForward
[ OK    ] ExecutorTest.AccelerationMoveForward (0 ms)
[ RUN   ] ExecutorTest.AccelerationTurnLeft
[ OK    ] ExecutorTest.AccelerationTurnLeft (0 ms)
[ RUN   ] ExecutorTest.AccelerationTurnRight
[ OK    ] ExecutorTest.AccelerationTurnRight (0 ms)
[ RUN   ] ExecutorTest.AccelerationComplex
[ OK    ] ExecutorTest.AccelerationComplex (0 ms)
[-----] 12 tests from ExecutorTest (4 ms total)

[-----] Global test environment tear-down
[=====] 12 tests from 1 test suite ran. (5 ms total)
[ PASSED ] 12 tests.
```