

Qt Shell

— C++课程设计答辩 —

C++ Course Design Defense

唐 雨 2021051602034 范榆康 2021051205051

> 目 录 <

1

项 目 介 绍

2

开 发 状 态

3

项 目 技 术

4

经 验 总 结

项 目 介 绍

Part 1

1

项目介绍

项目背景：

Linux shell 是操作系统中一个至关重要的组件，负责解析和执行用户输入的命令。传统的 Linux shell 基于命令行界面（CLI），对于普通用户来说可能不太友好。本项目旨在创建一个基于 Qt 6.5 和 QML 的图形化 Linux shell 程序，为用户提供一个直观且易于使用的图形用户界面（GUI），以便更方便地使用 Linux shell 功能。

制作成员：

本项目由重庆师范大学计算机与信息科学学院软件工程专业学生：范榆康、唐雨共同开发

开发环境：

基于Linux下 Qt6 C++ 开发，QML 界面开发任务提出者：学校、教育机构、企业或其他组织

用户：Unix/Linux系统用户

1

项目介绍

项目目标：

1. 用户友好性：提供一个直观的图形界面，简化用户输入和查看命令执行结果的过程。
2. 功能全面性：支持常见的 shell 命令，如 ls、cd、cat 等，和Shell脚本的执行。
3. 高可维护性：采用模块化设计，确保代码结构清晰，便于后续维护和扩展。
4. 三层架构：实现界面层（UI）、业务逻辑层（Business Logic）和数据层（Data Layer）分离，确保各层之间的良好封装和独立性。

开 发 状 态

Part 2

1 开 发 状 态

截至目前，软件已完成基础Shell命令实现，流程控制语句实现，界面实现，Shell脚本执行实现项目已完全开源，Github链接：<https://github.com/LiuZeOri/QtShell>

成员分工：

范榆康：负责Shell类、系统变量varlib代码编写、文档编写

唐雨：负责Controller类、QML界面以及流程控制代码编写

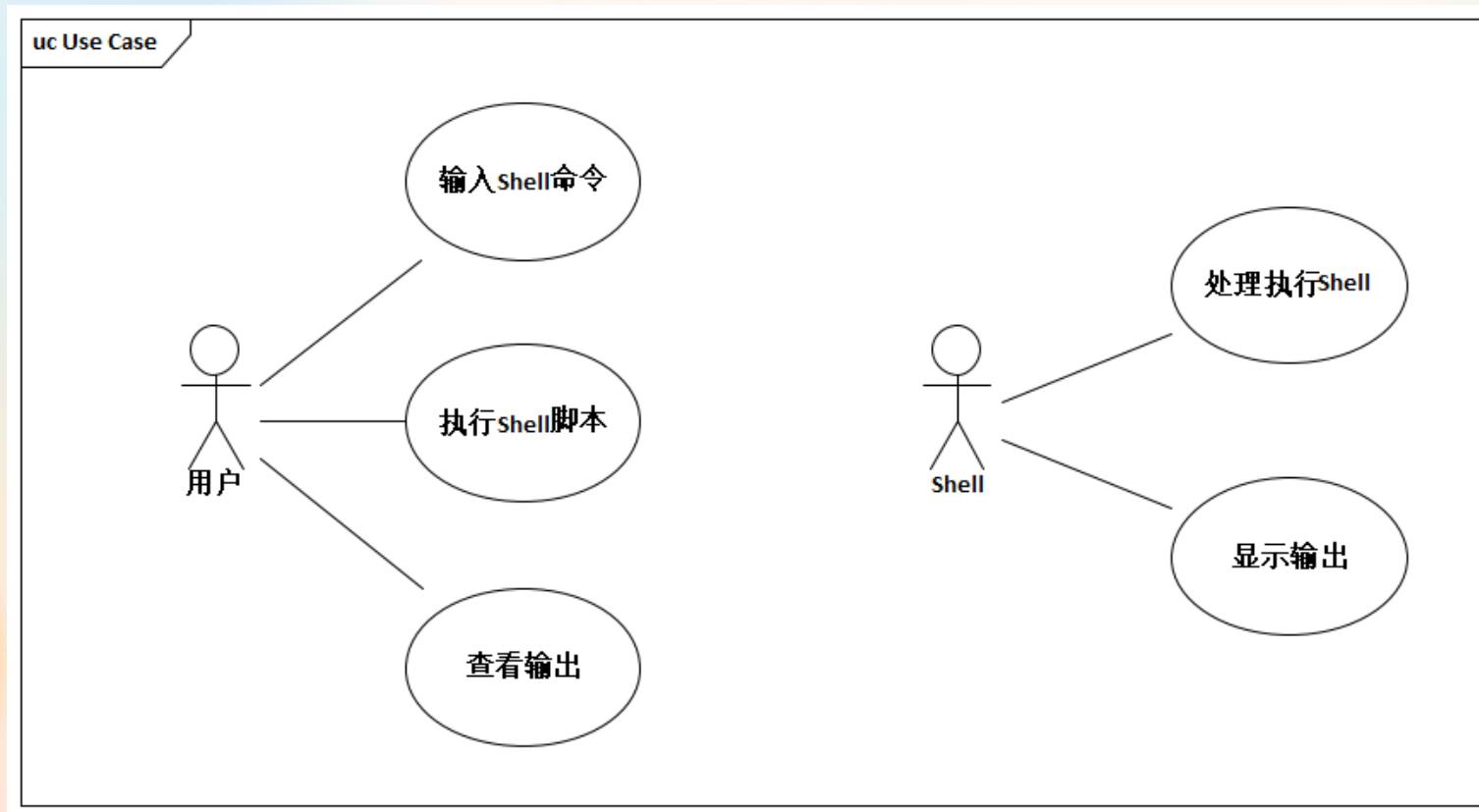
项 目 技 术

Part 3

3 项目技术

用例设计

主要用例为“用户”，用户具有输入Shell命令、执行Shell脚本、查看输出功能；Shell对用户操作进行处理，执行Shell和I/O重定向管道进行输出



3 项目技术

软件层次结构

界面层 (UI)：使用 QML 实现，包括命令输入框、输出显示区

业务逻辑层 (Business Logic)：处理用户输入的命令，调用相应的处理函数并返回结果

数据层 (Data Layer)：管理 Shell 环境变量和命令执行的具体实现

模块划分

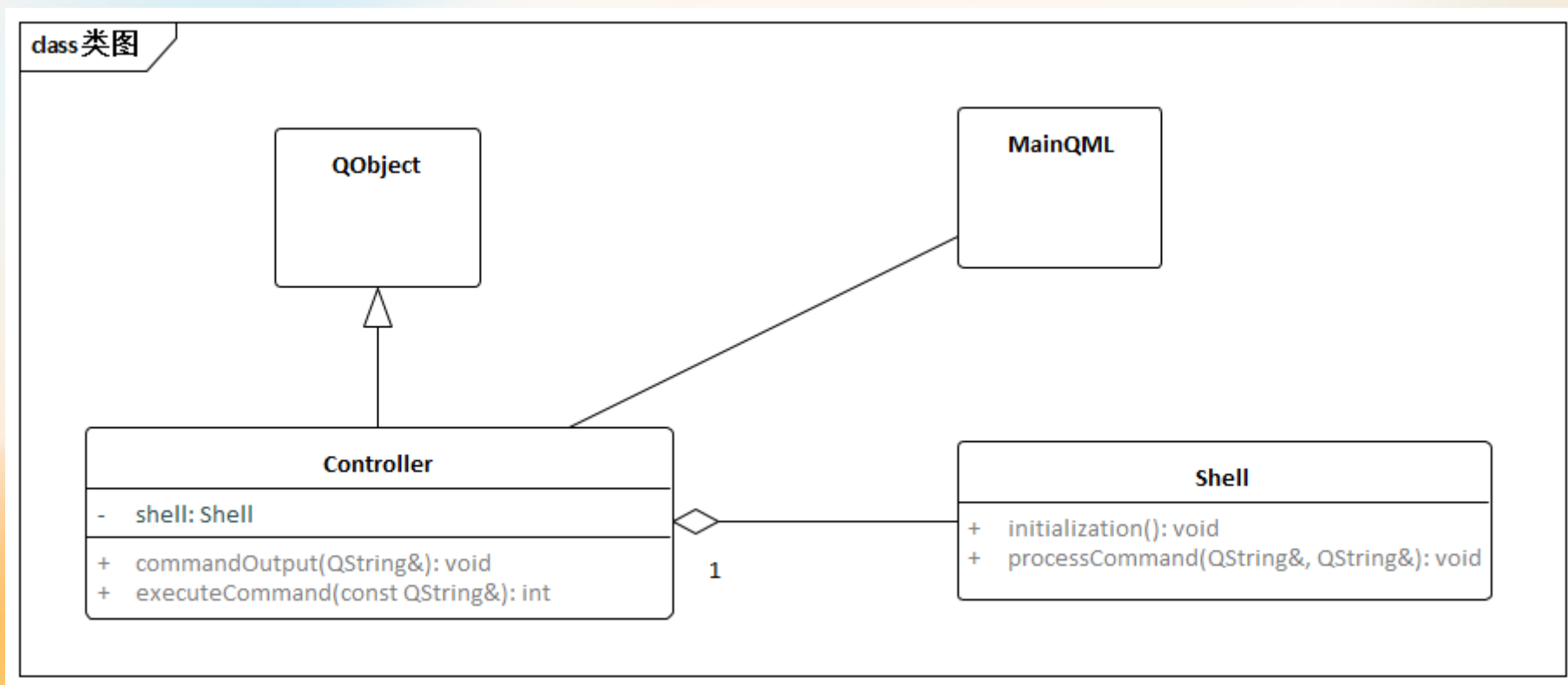
Controller：负责接收用户输入的命令，调用业务逻辑层进行处理，并将结果返回给界面层

Shell：具体处理用户输入的命令，调用数据层执行命令并返回结果

3 项目技术

类图

Controller 继承Qt 的QObject基类，具有一个Shell类实例



3 项目技术

流程图

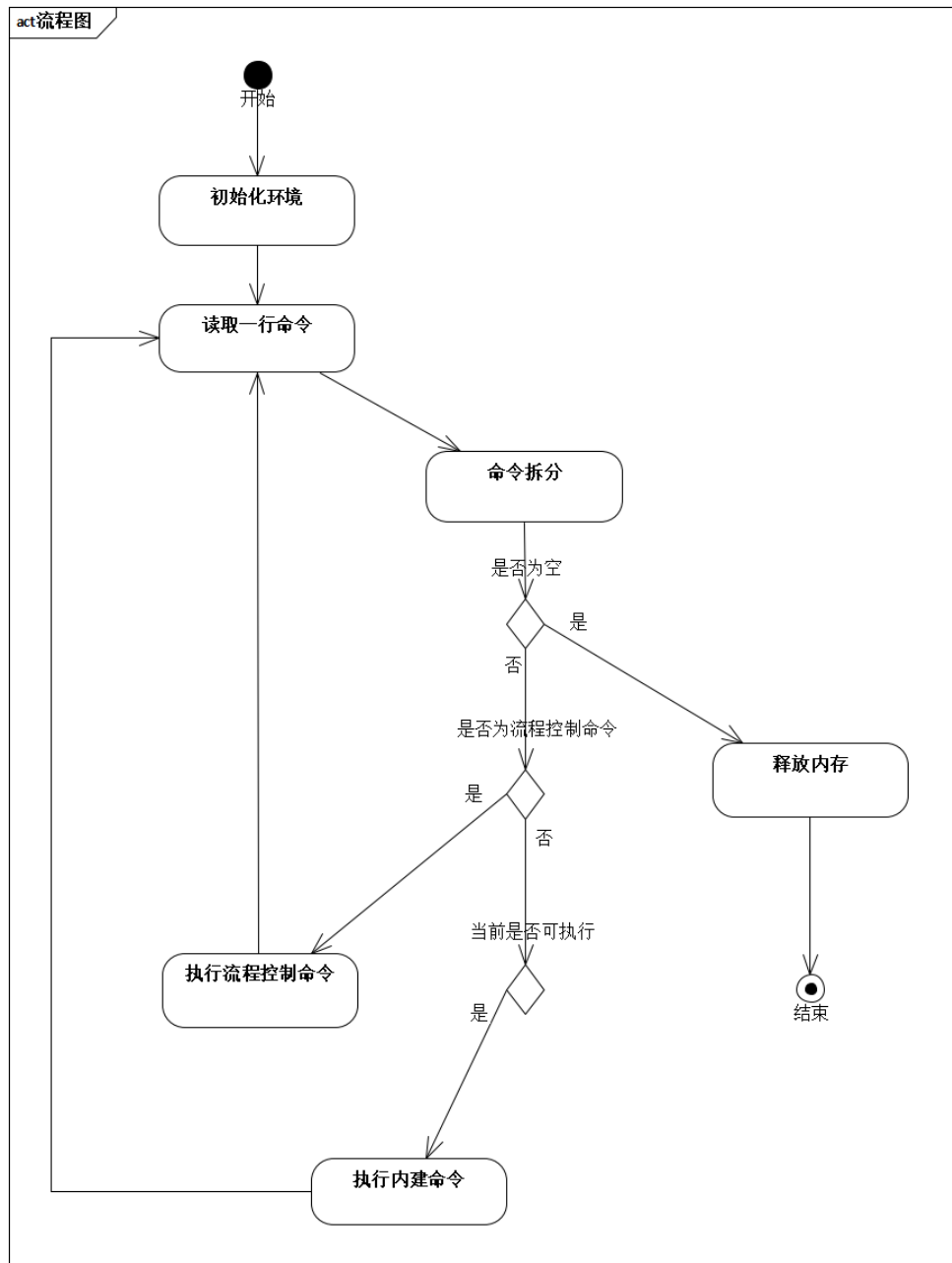
开始运行软件时，main主函数进行初始化并注册QML界面、初始化Controller类。

Controller 构造时初始化环境，开始命令的识别。

Shell 类每次识别一行命令，将命令拆分；如果为空则关闭结束，否则检测是

否为流程控制命令（if、else、fi）。是则进行流程控制命令，返回继续读取下一行

命令；否则检测是否可以执行，开始执行内建命令，完成后继续读取下一行



经验总结

Part 4

4 经验总结

1. 项目管理与需求分析

在项目初期，通过全面的需求分析和详细的项目计划，我们明确了项目的目标、核心功能和技术选型。清晰的需求文档为后续的开发提供了明确的方向，避免了开发过程中的需求变更和功能重复，我们使用Git作为版本管理，更好的协同合作。

2. 界面设计与用户体验

如何设计一个直观且易于使用的图形化界面是一个主要挑战。我们进行了多次用户调研和界面迭代，以确保最终界面能够满足用户需求。

3. 稳定性

处理用户输入的命令并返回结果需要高效且稳定的实现。在进行测试中我们发现，命令行经常卡死，我们使用DeBug寻找错误位置，从而修复漏洞。

4. 项目成果

成功开发并发布了基于 Qt 6.5 和 QML 的图形化 Shell 程序。项目实现了用户友好的图形界面和功能全面的 Shell 命令支持，提高了用户的操作效率和使用体验。



THANK YOU!