

《数据结构综合设计》

名称		作业 2 Qt&&数据结构		评分 1		评分 2		总分	
类型		<input type="checkbox"/> 验证型 <input checked="" type="checkbox"/> 设计型 <input type="checkbox"/> 综合型							
专业班级	卓软 2401	学号	5120246728						
任课教师	杨春明	姓名	华昊朗	时间			2025-07-01		
说明	评分 1 针对教学大纲课程目标 2，满分 10 分；评分 2 针对课程目标 3，满分 5 分。本次作业满分 15 分。								

一、目的

- 1、通过实现窗口布局、控件交互和信号槽机制，深入理解 Qt 界面开发的核心流程，熟悉 Qt Designer 可视化工具与代码实现的协同工作模式，掌握简单的 GUI 程序设计基础知识。
- 2、通过数据结构与 Qt 编程有机结合，进一步提高应用特定数据结构完成程序设计的能力。
- 3、能够根据程序设计的需求，具体练使用主流开发工具的能力和不断学习新技术的能力。

二、原理

通过信号与槽机制实现用户界面交互：使用 Qt Designer 设计的 UI 界面，在用户点击按钮时触发 clicked 信号，自动调用对应的槽函数进行硬编码验证，和预设的"testuser"和"password"进行比对，最后通过 QMessageBox 弹窗反馈比对结果。

三、环境

- 1、操作系统：Windows 11
- 2、集成开发环境：Qt 6.9，Qt Creator 17.0.0
- 3、编程语言：C++

四、内容与步骤

1、ui 设计

我们定义了四个可交互对象，对象名及类型、作用见下表 4-1

表 4-1 UI 对象及其功能

对象名	对象类型	对象功能
UserName	QPlainTextEdit	接受用户输入的用户名
Password	QLineEdit	接受用户输入密码
LoginButton	QPushButton	登录按钮
ExitButton	QPushButton	退出程序

因篇幅限制，暂省略 ui 设计部分代码。显示效果见图 5-1。

2、CMakeLists.txt (Qt 自动编写, 此处省略)

3、头文件 mainwindow.h 编写

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
class UserLoginForm;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_LoginButton_clicked();
    void on_ExitButton_clicked();

private:
    Ui::UserLoginForm *ui;
};

#endif // MAINWINDOW_H
```

4、登录逻辑 mainwindow.cpp 编写

```
#include "mainwindow.h"
```

```
#include "ui_mainwindow.h"
#include <QMessageBox>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::UserLoginForm)
{
    ui->setupUi(this);
    ui->Password->setEchoMode(QLineEdit::Password);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_LoginButton_clicked()
{
    QString username = ui->UserName->toPlainText().trimmed();
    QString password = ui->Password->text();

    if (username == "testuser" && password == "password") {
        QMessageBox::information(this, "登录成功", "欢迎, " + username + "!");
    } else {
        QMessageBox::warning(this, "登录失败", "用户名或密码错误");
        ui->Password->clear();
        ui->UserName->setFocus();
    }
}

void MainWindow::on_ExitButton_clicked()
{

```

```
this->close();  
}
```

5、主函数入口 main.cpp 编写

```
#include <QApplication>  
#include "mainwindow.h"  
  
int main(int argc, char *argv[])  
{  
    QApplication a(argc, argv);  
    MainWindow w;  
    w.show();  
    return a.exec();  
}
```

五、结果

经调试，基本实现了所给要求，能做到在输入账号密码与预设值不符时弹出警告，以及在校验通过后给出登录成功提示。

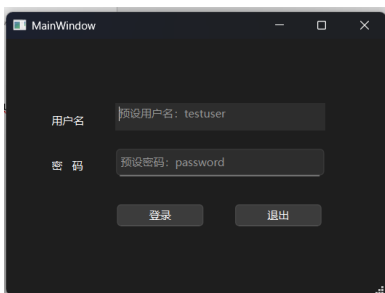


图 5-1 主界面效果



图 5-2 错误界面



图 5-3 登录成功页面

六、问题分析与解决

系统初步实现了登陆操作，并在测试中所给的测试数据都能返回预期结果。以下是几点亟需解决的问题。

1、UI 控件命名与代码引用不一致

UI 文件中控件命名为 UserName（QPlainTextEdit 类型），但代码中误写为 ui->usernameLineEdit，导致运行 Qt 生成的控件名与代码中手动编写的引用名不匹配，运行时找不到控件而崩溃。对此我们统一使用 UI 文件中的控件命名。

```
// 错误代码
QString username = ui->usernameLineEdit->toPlainText();

// 正确代码
QString username = ui->UserName->toPlainText();
```

七、总结

通过该项目的实践，我初步了解了 Qt 的基本运用。从 Qt 框架机制初步掌握了信号槽的自动/手动连接规则，理解了元对象系统如何动态关联界面交互与业务逻辑，并通过调试解决了因命名不一致导致的空指针问题。