1.新建项目：

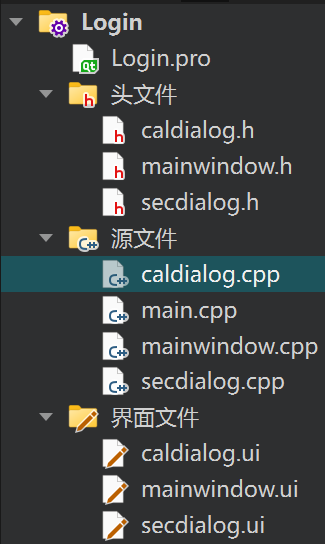


图1 项目框架

2.设置mainwindow.ui文件中的QMainWindow对象的分辨率（geometry参数）：

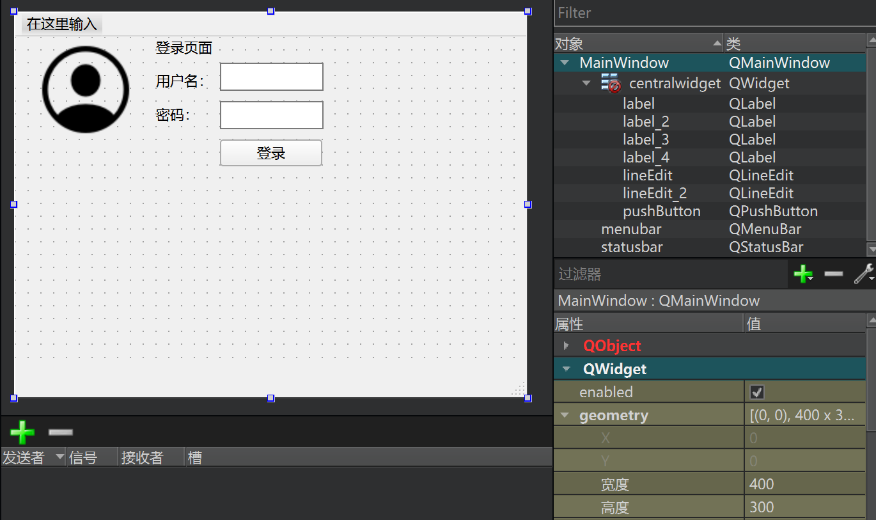


图2 QMainWindow分辨率设置

3.添加LineEdit，Label，PushButton，输入文字：

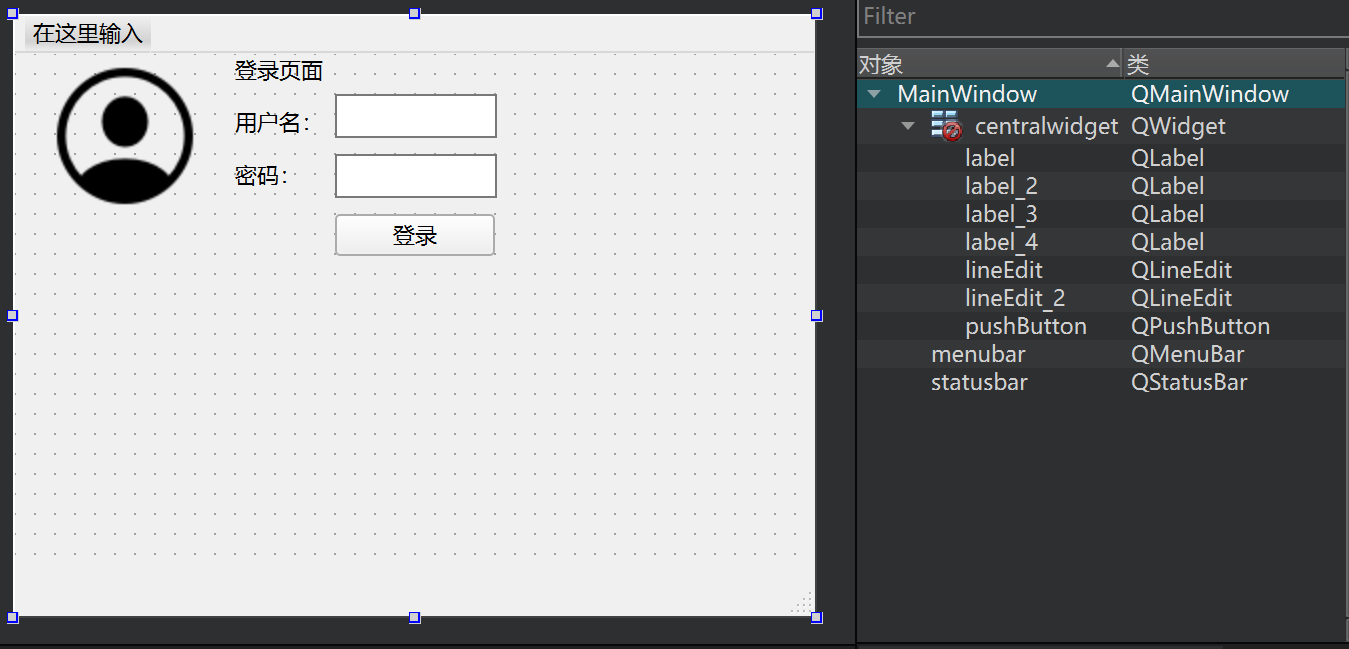


图3 添加控件

4.编写pushButton对象的代码：

void MainWindow::on\_pushButton\_clicked()

{

QString username=ui->lineEdit->text();

QString password=ui->lineEdit\_2->text();

if(username=="袁东豪"&&password=="3040213142"){

QMessageBox::information(this,"登录","密码正确");

secDialog=new SecDialog(this);

secDialog->show();

}else{

QMessageBox::information(this,"登录","用户名或密码错误");

}

}

5.编写SecDialog的代码：

* 在mainwindow.h中编写：

#include "secdialog.h"

private:

Ui::MainWindow \*ui;

SecDialog \*secDialog;

* 在secdialog.h中编写：

#include "caldialog.h"

private:

Ui::SecDialog \*ui;

calDialog \*CalDialog;

* 在secdialog.cpp中编写：

void SecDialog::on\_pushButton\_clicked()

{

close();

}

void SecDialog::on\_pushButton\_2\_clicked()

{

CalDialog=new calDialog(this);

CalDialog->show();

}

6.设置caldialog.ui文件中的QDialog对象的分辨率（geometry参数）：

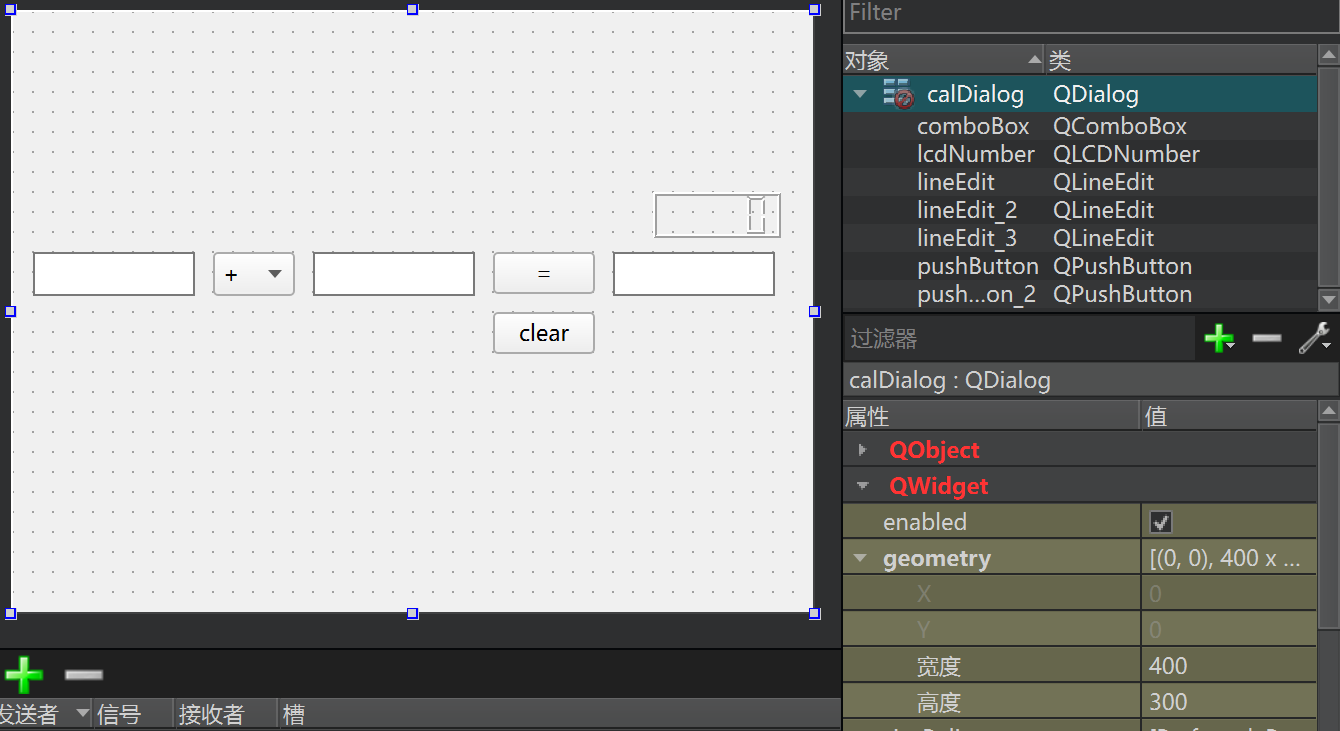


图4 QDialog分辨率设置

7.添加LineEdit，ComboxBox，PushButton，LcdNumber，输入文字：

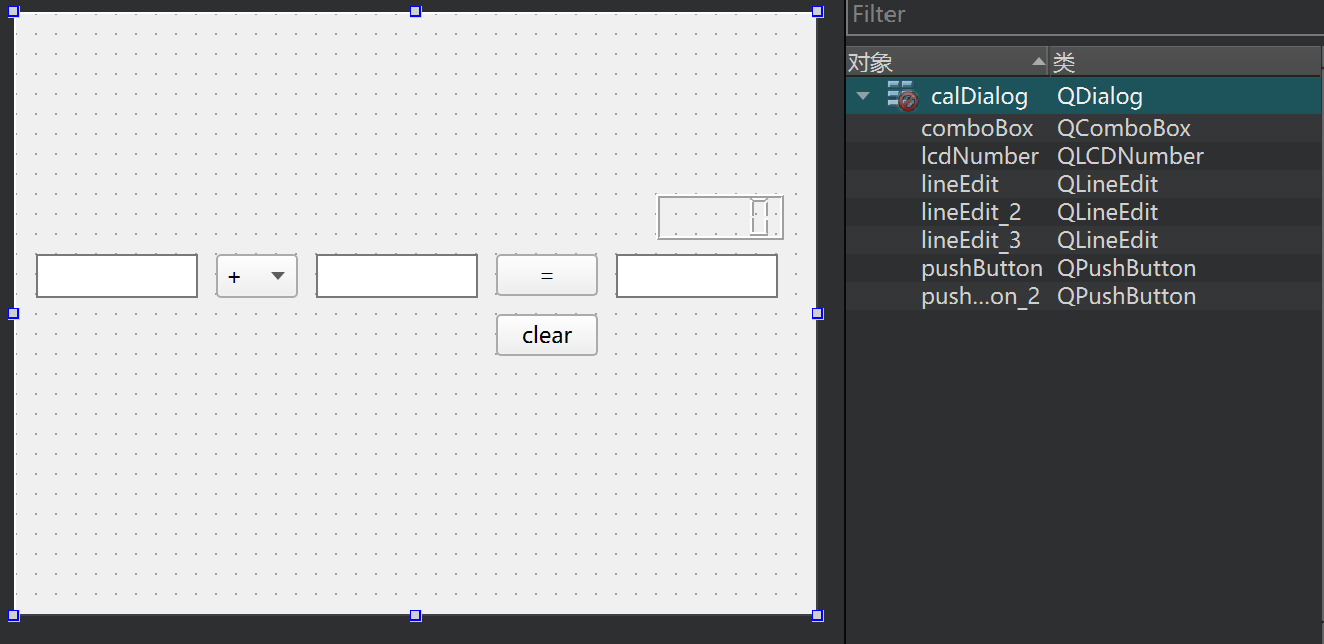


图5 添加控件

8.编写lineEdit及lineEdit\_2对象的代码：

* #include "QRegularExpression"
* 输入数字：

QRegularExpression regexp("0|[1-9]\\d{0,9}");

ui->lineEdit->setValidator(new QRegularExpressionValidator(regexp,this));

ui->lineEdit\_2->setValidator(new QRegularExpressionValidator(regexp,this));

double a = ui->lineEdit->text().toDouble();

double b = ui->lineEdit\_2->text().toDouble();

9.编写pushButton对象的代码：

* #include "QMessageBox"

在caldialog.cpp中添加槽函数（计算结果并显示，警告）：

void MainWindow::on\_pushButton\_clicked()

{

double a = ui->lineEdit->text().toDouble();

double b = ui->lineEdit\_2->text().toDouble();

double c;

if(ui->comboBox->currentText()=='+'){

c=a+b;

}

if(ui->comboBox->currentText()=='-'){

c=a-b;

}

if(ui->comboBox->currentText()=='\*'){

c=a\*b;

}

if(ui->comboBox->currentText()=='/'){

if(b==0){

QMessageBox::about(this,"error","除数不能为0");

return; }

c=a/b;

}

ui->lineEdit\_3->setText(QString::number(c));

ui->lcdNumber->display(c);

}

10.编写pushButton\_2对象的代码：

* 在mainwindow.cpp中添加槽函数（清除所有数据）：

void MainWindow::on\_pushButton\_2\_clicked()

{

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

ui->lcdNumber->display(0);

}

11.编写lcdNumber对象的代码：

* 设置字体颜色及显示位数：

ui->lcdNumber->setDigitCount(9);

ui->lcdNumber->setPalette(Qt::green);

12.项目测试：

通过构建（Build）检查是否存在error或者warning，在通过构建后点击运行按钮进行测试，经验证实现了简易登录计算器的功能。



图6 Qt Creator