1.新建项目：

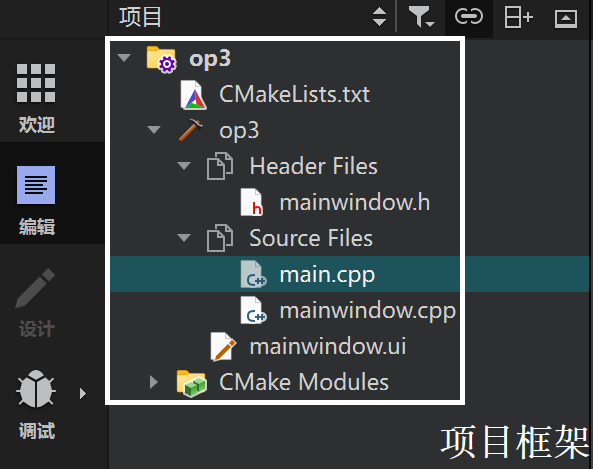


图1 项目框架

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

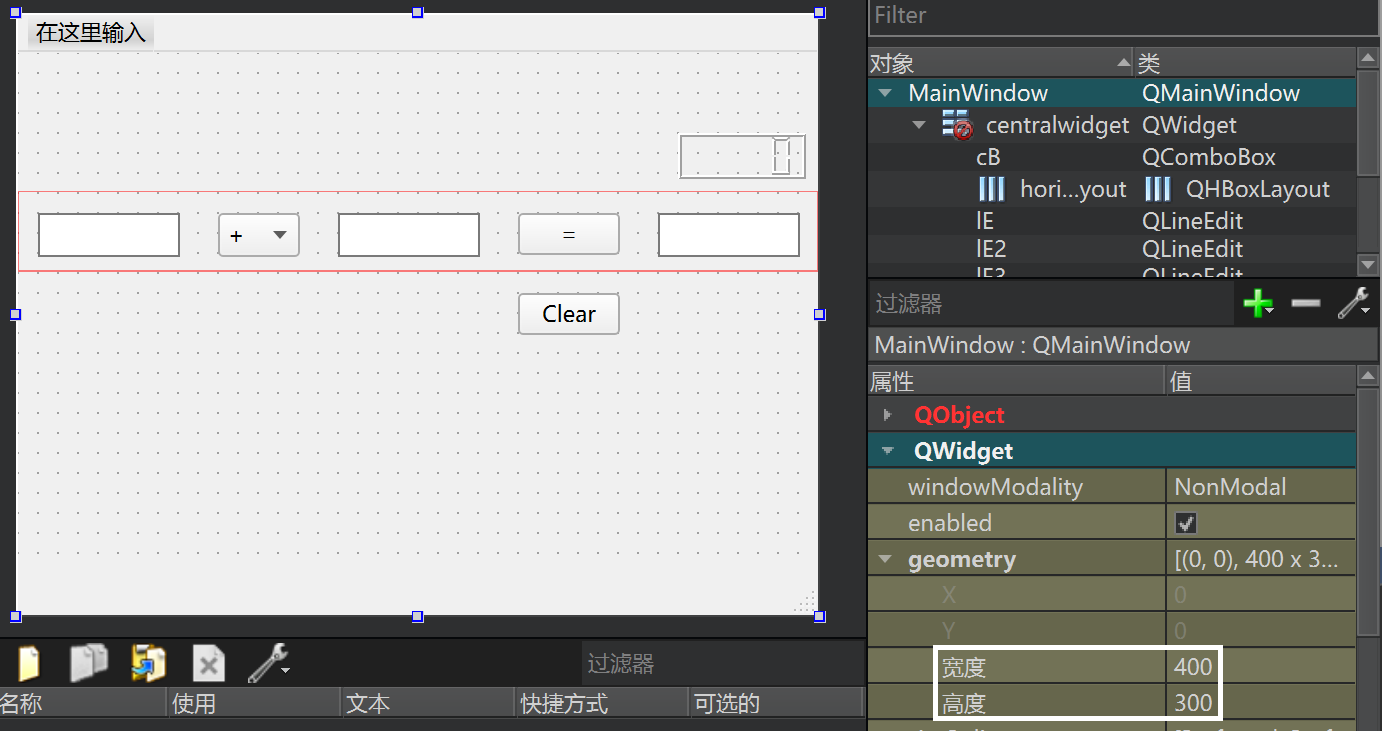


图2 QMainWindow分辨率设置

3.添加LineEdit，ComboxBox，PushButton，LcdNumber及HorizontalLayout，输入文字且水平对齐：



图3 添加控件

4.编写lE及lE2对象的代码：

* #include "QRegularExpression"
* 输入数字：

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

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

ui->lE2->setValidator(new QRegularExpressionValidator(regexp,this));

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

double b = ui->lE2->text().toDouble();

5.编写pB对象的代码：

* #include "QMessageBox"

在mainwindow.cpp中添加槽函数（计算结果并显示，警告）：

void MainWindow::on\_pB\_clicked()

{

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

double b = ui->lE2->text().toDouble();

double c;

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

c=a+b;

}

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

c=a-b;

}

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

c=a\*b;

}

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

if(b==0){

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

return; }

c=a/b;

}

ui->lE3->setText(QString::number(c));

ui->lN->display(c);

}

6.编写pB2对象的代码：

* 在mainwindow.cpp中添加槽函数（清除所有数据）：

void MainWindow::on\_pB2\_clicked()

{

ui->lE->clear();

ui->lE2->clear();

ui->lE3->clear();

ui->lN->display(0);

}

7.编写lN对象的代码：

* 设置字体颜色及显示位数：

ui->lN->setDigitCount(9);

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

8.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，经验证实现了简易计算器的功能。

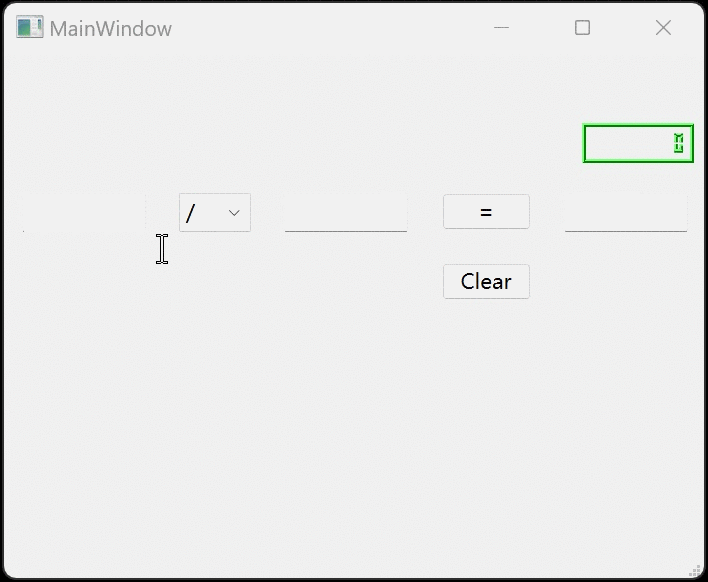


图4 Qt Creator验证测试