Archived:Creating a simple calculator using QWidgets

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on QWidget) is deprecated.

This code snippet demonstrates how to create a simple calculator using QWidget based C++ code.

Layout Design(Coded Design)

Header File

Cal.h

```
#ifndef CAL_H
#define CAL_H

#include <QtGui/QWidget>
#include "ui_cal.h"
#include<QString>
#include<QPushButton>
#include<QLineEdit>
#include<QHBoxLayout>
#include<QVBoxLayout>
#include<QFormLayout>
class cal : public QWidget
{
    Q_OBJECT

public:
 cal(QWidget *parent = 0);
    ~cal();

private:
    QPushButton *but1;
    QPushButton *sub;
    QPushButton *mul;
    QPushButton *div;
    QPushButton *eq;
    QFormLayout *lay;
    QLineEdit *line;
    QString str;
    QString str1;
    QString ch;
    QString strResult;
private slots:
   void addi();
   void subs();
   void mult();
   void divi();
   void em();
public:
        float num;
        float num1;
        float ans;
};

#endif // CAL_H
```

Source code

cal.cpp

```
#include "cal.h"

cal::cal(QWidget *parent)
    : QWidget(parent)
{
 lay=new QFormLayout(this);
        but1=new QPushButton("+",this);
        sub=new QPushButton("-",this);
        mul=new QPushButton("*",this);
        div=new QPushButton("/",this);
        eq=new QPushButton("=",this);
        line=new QLineEdit(this);
        line->setMaxLength(12);
        lay->addWidget(line);
        lay->addWidget(but1)
        lay->addWidget(sub);
        lay->addWidget(mul);
        lay->addWidget(div);
        lay->addWidget(eq);
        setLayout(lay);
        QObject::connect(but1,SIGNAL(clicked()),this,SLOT(addi()));
        QObject::connect(sub,SIGNAL(clicked()),this,SLOT(subs()));
```

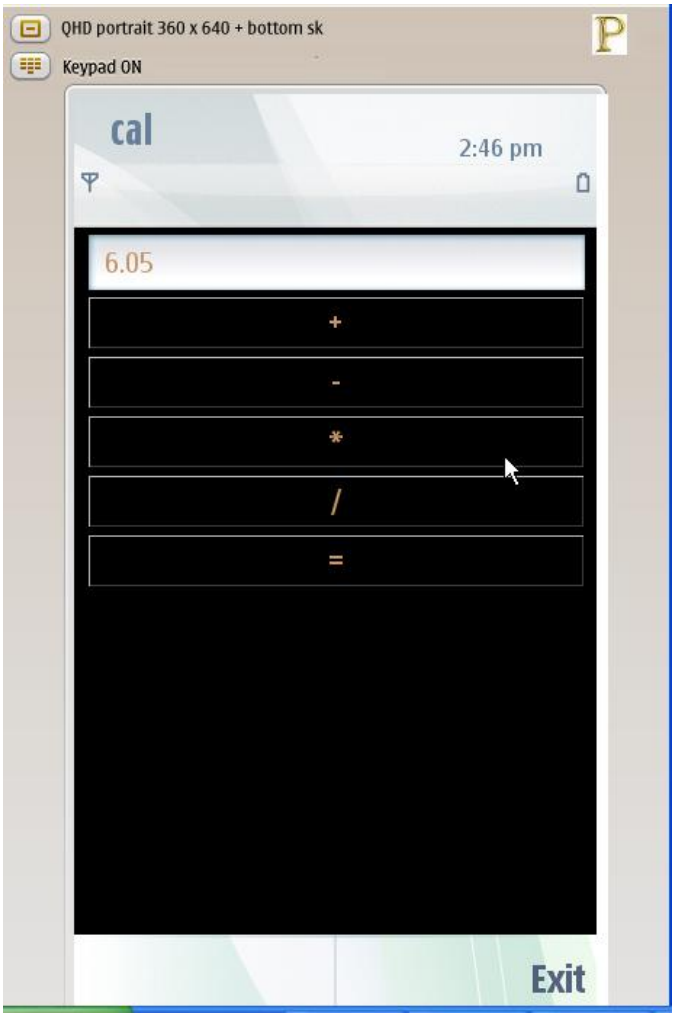

(C) Copyright Nokia 2013. All rights reserved.

```
        QObject::connect(mul,SIGNAL(clicked()),this,SLOT(mult()));
        QObject::connect(div,SIGNAL(clicked()),this,SLOT(divi()));
        QObject::connect(eq,SIGNAL(clicked()),this,SLOT(em()));
}

cal::~cal()
{
 // No need to delete any object that got a parent that is properly deleted.

}
void cal::addi()
{
                str=line->text();
                num=str.toFloat();
                ch=but1->text();
                line->clear();
}
void cal::subs()
{
                str=line->text();
                num=str.toFloat();
                ch=sub->text();
                line->clear();
}
void cal::mult()
{
                str=line->text();
                num=str.toFloat();
                ch=mul->text();
                line->clear();
}
void cal::divi()
{
                str=line->text();
                num=str.toFloat();
                ch=div->text();
                line->clear();
}
void cal::em()
{


        str1 = line->text();
        num1 = str1.toFloat();
        if (ch=="+")
        {
                ans=num+num1;
        }
        else if(ch=="-")
        {
                ans=num-num1;
        }
        else if(ch=="*")
        {
                ans=num*num1;
        }
        else if(ch=="/")
        {
                ans=num/num1;
        }
        else
        {
               strResult = strResult.number(ans);
                line->setText(strResult);
        }
        strResult = strResult.number(ans);
        line->setText(strResult);
        ch="";
}
```

Screenshot

- Input any value in the lineEdit.

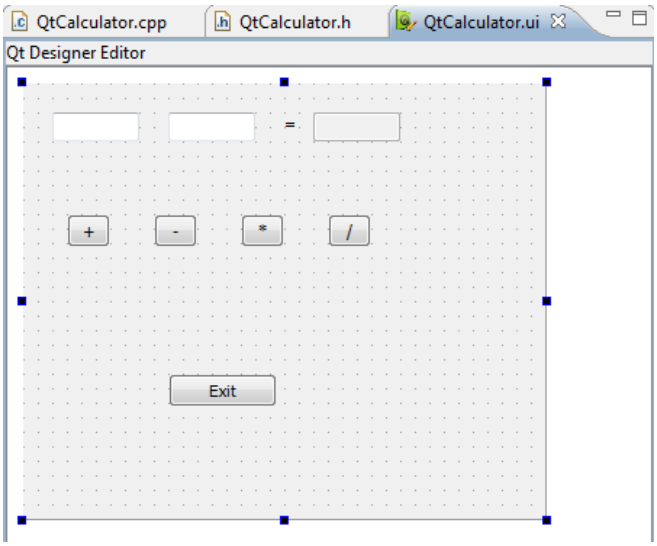
(C) Copyright Nokia 2013. All rights reserved.



- Decimal operation is also possible in this calculator

UI design

QtCalculator.ui



Header file

QtCalculator.h

(C) Copyright Nokia 2013. All rights reserved.

```
#ifndef QTCALCULATOR_H
#define QTCALCULATOR_H

#include <QtGui/QWidget>
#include "ui_QtCalculator.h"

class QtCalculator : public QWidget
{
    Q_OBJECT

public:
 QtCalculator(QWidget *parent = 0);
    ~QtCalculator();

public slots:
 void Addition();
 void Subtraction();
 void Multiplication();
 void Division();

private:
    Ui::QtCalculatorClass ui;
};

#endif // QTCALCULATOR_H
```

Source file

QtCalculator.cpp

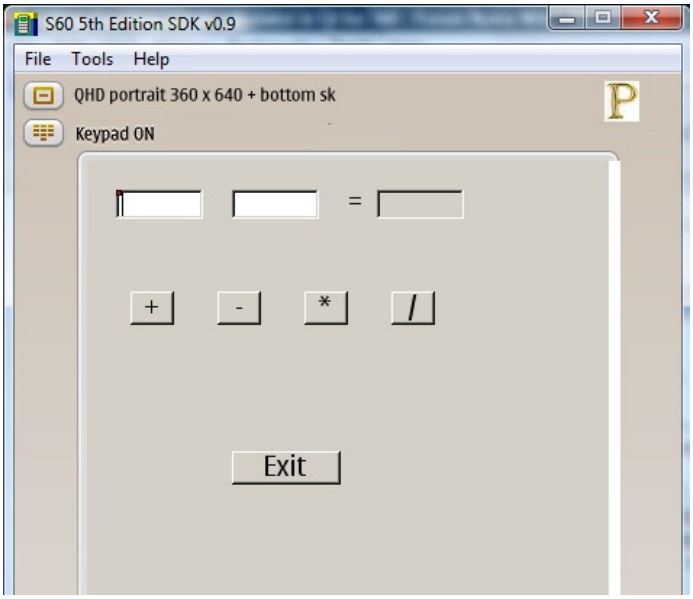
```
#include "QtCalculator.h"

QtCalculator::QtCalculator(QWidget *parent)
    : QWidget(parent)
{
 ui.setupUi(this);
 QObject::connect(ui.plusButton, SIGNAL(clicked()), this, SLOT(Addition()));
 QObject::connect(ui.minusButton,SIGNAL(clicked()), this, SLOT(Subtraction()));
 QObject::connect(ui.multiplyButton,SIGNAL(clicked()), this, SLOT(Multiplication()));
 QObject::connect(ui.divisionButton,SIGNAL(clicked()), this, SLOT(Division()));
}

QtCalculator::~QtCalculator()
{
 QString strFisrt = ui.lineEditNumber1->text();
 QString strSecond = ui.lineEditNumber2->text();

 bool ok;
 int number1 = strFisrt.toInt(&ok);
 int number2 = strSecond.toInt(&ok);
 int result = number1 + number2;

 QString strResult;
 strResult = strResult.number(result);
 ui.lineEditResult->setText(strResult);
}

void QtCalculator::Addition()
{
 QString strFisrt = ui.lineEditNumber1->text();
 QString strSecond = ui.lineEditNumber2->text();

 bool ok;
 int number1 = strFisrt.toInt(&ok);
 int number2 = strSecond.toInt(&ok);
 int result = number1 + number2;

 QString strResult;
 strResult = strResult.number(result);
 ui.lineEditResult->setText(strResult);
}

void QtCalculator::Subtraction()
{
 QString strFisrt = ui.lineEditNumber1->text();
 QString strSecond = ui.lineEditNumber2->text();

 bool ok;
 int number1 = strFisrt.toInt(&ok);
 int number2 = strSecond.toInt(&ok);
 int result = number1 - number2;

 QString strResult;
 strResult = strResult.number(result);
 ui.lineEditResult->setText(strResult);
}

void QtCalculator::Multiplication()
{
 QString strFisrt = ui.lineEditNumber1->text();
 QString strSecond = ui.lineEditNumber2->text();

 bool ok;
 int number1 = strFisrt.toInt(&ok);
 int number2 = strSecond.toInt(&ok);
 int result = number1 * number2;

 QString strResult;
 strResult = strResult.number(result);
 ui.lineEditResult->setText(strResult);
}

void QtCalculator::Division()
{
 QString strFisrt = ui.lineEditNumber1->text();
 QString strSecond = ui.lineEditNumber2->text();

 bool ok;
```


(C) Copyright Nokia 2013. All rights reserved.

```
  int number1 = strFisrt.toInt(&ok);
  int number2 = strSecond.toInt(&ok);
  int result = number1 / number2;

  QString strResult;
  strResult = strResult.number(result);
  ui.lineEditResult->setText(strResult);
 }
```

Postconditions

- It will display a simple calculator on the screen.



Example

- Download the working example from this link: QtCalculator.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.