

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

LUCRAREA DE LABORATOR#2

GUI Development

Autor:
Alexandru STAMATIN

lector asistent:
Irina COJANU
lector superior:
Radu MELNIC

Lucrarea de laborator #2

1 Scopul lucrării de laborator

Studierea dezvoltării aplicațiilor cu interfața grafică pentru utilizator

2 Obiective

- Realizarea unui simplu GUI Calculator
- Operațiile simple: +, -, *, /, putere, radical, Inversare Semn, operații cu numere zecimale
- Divizare a proiectului în două module - Interfața grafică (Modulul GUI) și Modulul de bază (Core Module)

3 Efectuarea lucrarii de laborator

3.1 Sarcinile propuse

- Realizarea unui simplu GUI calculator care suporta urmatoarele functii: +, -, ./, *, putere, radical, InversareSemn(+/-), operatii cu numere zecimale
- Divizarea proiectului in doua module - Interfata grafica (Modul GUI) si Modulul de baza (Core Module)

3.2 Analiza lucrarii de laborator

Link la repozitoriu: <https://github.com/AlexStamatin/MIDPS>

In cadrul acestei lucrari de laborator am elaborat un calculator cu interfata grafica pentru utilizator. Pentru realizarea acestui scop a fost utilizat framework-ul Qt pentru limbajul C++. Datorita add-on-ului Qt pentru Microsoft Visual studio 2015 a fost posibila dezvoltarea interfetei grafice in cadrul Qt Designer si a modului de baza in cadrul Visual Studio.

- Crearea interfetei grafice

Interfata grafica a fost creata cu ajutorul Qt Designer. Au fost plasate 2 campuri pentru afisarea textului temp label si Rez label. In temp label sunt afisate operatiunile curente iar Rez label afiseaza rezultatul. De asemenea au fost adaugate butoane pentru toate operatiunile suportate de calculator si pentru introducerea cifrelor.

Object	Class
CalcClass	QMainWindow
centralWidget	QWidget
Rez_label	QLabel
pushButton_0	QPushButton
pushButton_1	QPushButton
pushButton_2	QPushButton

Figure 1: Elementele grafice ale calculatorului

A fost editat style-sheet-ul componentelor grafice pentru a modifica aspectul acestora (culoarea, granitele, etc.). Astfel a fost obtinut un fisier Calc.ui ce reprezinta interfata grafica a calculatorului.

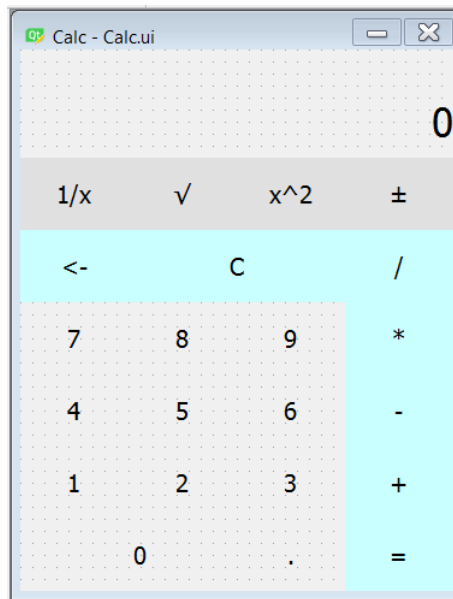


Figure 2: Calc.ui

- Dezvoltarea modului de baza

In cadrul clasei Calc au fost create private-sloturi pentru toate functionalitatile suportate de calculator.

private slots:

```
void DigitPress();
void SqrtOpPress();
void SqrOpPress();
void InvertPress();
void AddOpPress();
void MultOpPress();
void EqPress();
void DeciPress();
void SgnPress();
void BackspacePress();
void ClearPress();
```

Functia DigitPress() trateaza apasarea tastelor numerice

```
currNum = (ui->Rez_label->text() + button->text()).toDouble();
```

Functia SqrtOpPress() trateaza efectuarea operatiunii radacina patrata

```
rez = std::sqrt(oper);
```

Funcția Sqr() tratează efectuarea operațiunii ridicarea la patrat

```
rez = std::sqrt(oper);
```

Funcția InvertPress() tratează efectuarea a operațiunii de inversare a numărului

```
if (oper < 0.0)
{
    InvalidOp();
    return;
}
rez = 1.0 / oper;
```

Funcția AddOpPress() tratează efectuarea operațiilor de adunare și scădere

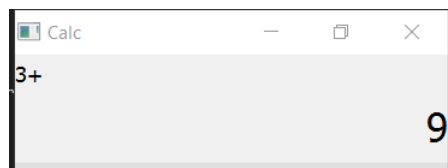


Figure 3: Adunarea a 2 numere

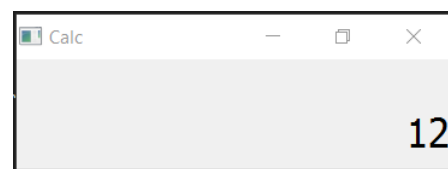


Figure 4: Rezultatul adunării

Funcția MultOpPress tratează efectuarea operațiilor de înmulțire și împărțire

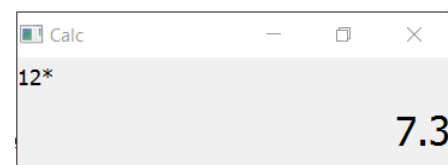


Figure 5: Înmulțirea a 2 numere

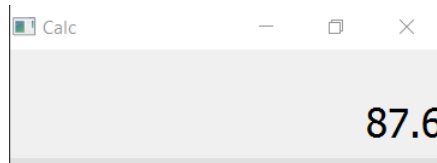


Figure 6: Rezultatul inmultirii

Functia `EqPress()` trateaza afisarea rezultatului calculelor

```
ui->Rez_label->setText ( QString::number ( CurrRez ) );
```

Functia `DeciPress()` trateaza introducerea numerelor cu parte fractionara

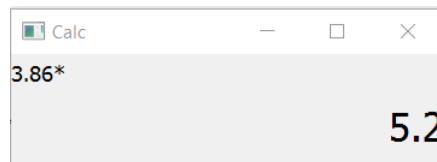


Figure 7: Introducerea numerelor cu parte fractionara

Functia `SgnPress()` trateaza schimbarea semnului numarului

```
if ( num > 0.0 ) {
    toshow.prepend ( tr ( " - " ) );
}
else if ( num < 0.0 ) {
    toshow.remove ( 0 , 1 );
}
```

Functia `BackspacePress()` trateaza eliminarea unui operand din calculul curent

```
toshow.chop ( 1 );
```

Functia `ClearPress()` trateaza eliberarea numerelor din label si stergerea valorilor curente

```
CurrRez = 0.0;
CurrOp = 0.0;
NextAddOper.clear ();
NextMultOper.clear ();
ui->Rez_label->setText ( " 0 " );
ui->temp_label->clear ();
```

Concluzie

In cadrul acestei lucrari de laborator am realizat un calculator cu interfata grafica pentru utilizator. In urma efectuarii lucrarii am obtinut deprinderi de utilizare a framework-ului Qt in limbajul C++. Am studiat procesul de instalare a add-on-urilor pentru Visual Studio si de creare a unei interfete grafice cu ajutorul Qt Designer. Am determinat un sir de avantaje al utilizarii framework-ului Qt

- Oferă toate instrumentele necesare pentru scrierea unei aplicatii
- Este valabil gratuit pentru dezvoltarea aplicatiilor open-source
- Este mai simplu in utilizare si performant ca alte produse

In acelasi timp are si anumite dezavantaje

- Nu sunt utilizati smart pointeri
- Nu sunt utilizate bibliotecile standard C++

Interfata grafica este importanta pentru asigurarea unei interactiuni eficiente intre utilizatorul final si produsul soft. Elementele grafice utilizate trebuie sa fie intuitive iar scopul lor clar si bine-definit. Este preferabil ca interfata sa utilizeze la maxim cunstintele pe care utilizatorul deja le poseda in urma utilizarii altor aplicatii.

Ca rezultat al efectuării lucrării de laborator am ajuns la concluzia ca este preferabil de separat modulul grafic si cel de baza intr-un program pentru a putea modifica un modul fara a aplica schimbari la celalalt. De asemenea am stabilit avantajele si importanta utilizării unei interfete grafice si am obtinut deprinderi de creare a acesteia.

References

- [1] <http://doc.qt.io/qt-4.8/designer-using-a-ui-file.html>
- [2] <http://doc.qt.io/qt-5/qtwidgets-widgets-calculator-example.html>