

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.3

la Medii Interactive de Dezvoltare a Produselor Soft

A efectuat:
st.gr.TI-144

Pascari Ion

A verificat:
dr.conf.univ.,

Cojocaru Svetlana

Chișinău 2016

Lucrarea de laborator nr.3

Tema: GUI Calculator

Scopul lucrării:

- Realizeaza un simplu GUI Calculator
- Operatiile simple: +, -, *, /, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.
- Divizare proiectului in doua module - Interfata grafica (Modul GUI) si Modulul de baza (Core Module).

Formularea condiției problemei (sarcina de lucru):

- *Basic Level* (nota 5 || 6):
 - Realizeaza un simplu GUI calculator care suporta functiile de baza: +, -, /, *.
- *Normal Level* (nota 7 || 8):
 - Realizeaza un simplu GUI calculator care suporta urmatoare functii: +, -, /, *, putere, radical, InversareSemn(+/-).
- *Advanced Level* (nota 9 || 10):
 - Realizeaza un simplu GUI calculator care suporta urmatoare functii: +, -, /, *, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.
 - Divizare proiectului in doua module - Interfata grafica (Modul GUI) si Modulul de baza (Core Module).

Implementarea task-urilor :

- IDE: QtCreator
- Limbaje de programare: C++
- Tehnologii si Frameworks: Qt



- *Basic Level* (nota 5 || 6):

- Realizeaza un simplu GUI calculator care suporta functiile de baza: +, -, /, *.

```
void additiveOperatorClicked();
void multiplicativeOperatorClicked();

bool Calculator::calculate(double rightOperand, const QString &pendingOperator)
{
    if (pendingOperator == tr("+")) {
        sumAccumulated += rightOperand;
    } else if (pendingOperator == tr("-")) {
        sumAccumulated -= rightOperand;
    } else if (pendingOperator == tr("\303\227")) {
        factorAccumulated *= rightOperand;
    } else if (pendingOperator == tr("\303\267")) {
        if (rightOperand == 0.0)
            return false;
        factorAccumulated /= rightOperand;
    }
    return true;
}
```

- *Normal Level* (nota 7 || 8):

- Realizeaza un simplu GUI calculator care suporta urmatoare functii: +, -, /, *, putere, radical, InversareSemn(+/-).

```
void Calculator::changeSignClicked()
{
    QString text = display->text();
    double value = text.toDouble();

    if (value > 0.0) {
        text.prepend(tr("-"));
    } else if (value < 0.0) {
        text.remove(0, 1);
    }
    display->setText(text);
}

void Calculator::unaryOperatorClicked()
{
    Button *clickedButton = qobject_cast<Button *>(sender());
    QString clickedOperator = clickedButton->text();
    double operand = display->text().toDouble();
    double result = 0.0;

    if (clickedOperator == tr("\342\210\232")) {
        if (operand < 0.0) {
            divisionByZero();
            return;
        }
        result = sqrt(operand);
    } else if (clickedOperator == tr("x\302\262"))
        result = pow(operand, 2.0);
    else if (clickedOperator == tr("x\302\263"))
```

```

        result = pow(operand, 3.0);
    else if(clickedOperator == tr("sin"))
        result = sin(operand);
    else if(clickedOperator == tr("cos"))
        result = cos(operand);
    else if(clickedOperator == tr("tan"))
        result = tan(operand);
    else if(clickedOperator == tr("atan"))
        result = atan(operand);
    else if (clickedOperator == tr("1/x")) {
        if (operand == 0.0) {
            divisionByZero();
            return;
        }
        result = 1.0 / operand;
    }
    display->setText(QString::number(result));
    waitingForOperand = true;
}

```

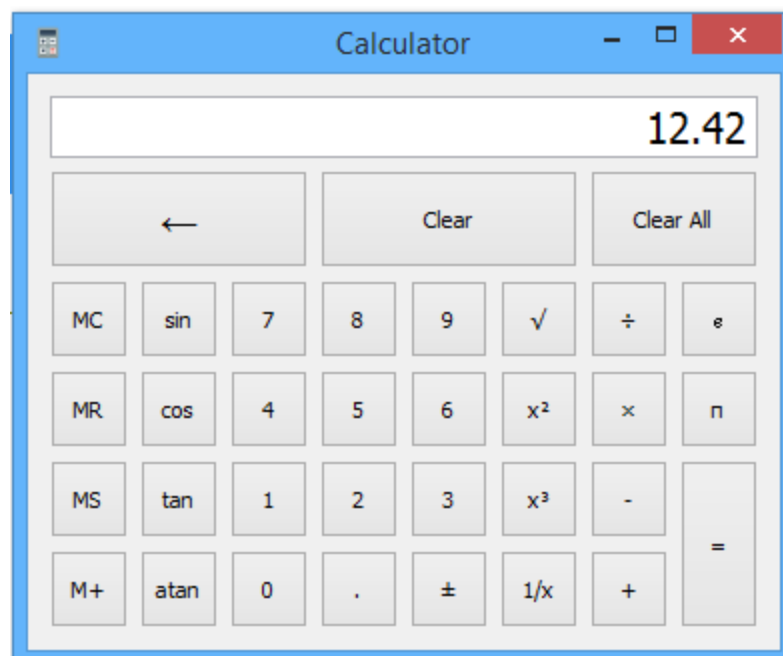
- *Advanced Level* (nota 9 || 10):

- Realizeaza un simplu GUI calculator care suporta urmatoare functii: +, -, /, *, putere, radical, InversareSemn(+/-), operatii cu numere zecimale.

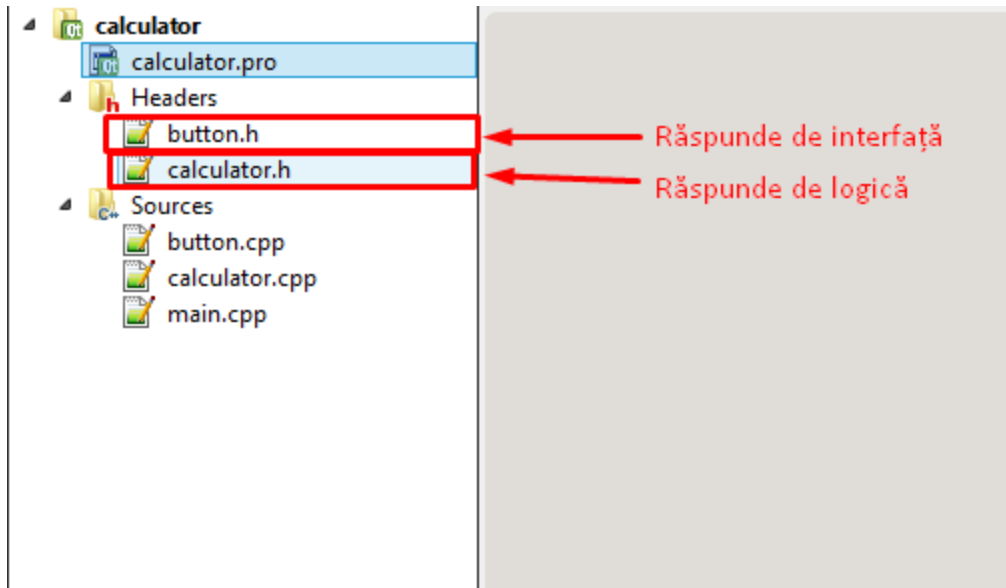
```

void Calculator::pointClicked()
{
    if (waitingForOperand)
        display->setText("0");
    if (!display->text().contains("."))
        display->setText(display->text() + tr("."));
    waitingForOperand = false;
}

```



- Divizare proiectului in doua module - Interfata grafica(Modul GUI) si Modulul de baza(Core Module).



Concluzii

În această lucrare am pus în practică Qt Creator împreună cu limbajul de programare C++ pentru a crea un GUI calculator. Lucrând în Qt am creat două clase : Clasa Button și Clasa Calculator, Button răspunde de interfață și are definit acolo un constructor pentru crearea dinamică a butoanelor și a private slot-urilor specifice fiecărui buton, în felul următor :

```
Button *plusButton = createButton(tr("+"),
SLOT(additiveOperatorClicked()));
```

În constructor-ul Calculator am inițializat o interfață de tip grid care se modifică dinamic și proporțional în timp ce se adaugă elemente :

```
QGridLayout *mainLayout = new QGridLayout;
mainLayout->setSizeConstraint(QLayout::SetFixedSize);
mainLayout->addWidget(display, 0, 0, 1, 8);
```

În final am definit câte o metodă pentru fiecare buton apăsător care are grijă de calculele necesare introduse de utilizator.

Bibliografie

- <http://doc.qt.io/qtcreator/creator-tutorials.html>
- <http://doc.qt.io/qtcreator/>
- Îndrumar metodic pentru lucrările de laborator la MIDPS