**Отчет по лабораторной работе №5**

**“Создание простейшего приложения с графическим интерфейсом пользователя на основе библиотеки Qt”**

1. **Постановка задачи:**

На основе приведенного в разделе Справочный материал текста приложения сконструировать в Qt Creator приложение с графическим интерфейсом на основе Qt, выполняющее функцию калькулятора. Простейший вариант приложения представляет собой форму, имеющую два поля для ввода данных и одно — для вывода. Вычисления выполняются при нажатии пользователем соответствующей кнопки. Минимальный набор функций калькулятора: сложение, вычитание, умножение, деление.

Дополнительное задание:

* + добавить любую из функций: вычисление квадратного корня, модуля числа, тригонометрических функций;
  + добавить очистку окон ввода;
  + запретить ввод данных в окно вывода;
  + добавить проверку правильности ввода числа пользователем;
  + разработать программу с одним полем для ввода данных и с выводом в него же результата.

1. **Структуры, использованные в программе:**

# Классы библиотеки Qt

* QString — класс строки.
* QApplication – класc приложения.
* QWidget — класс окна.
* QPushButton ­ класс кнопки.
* QLabel — класс надписи.
* QLineEdit — класс окна однострочного редактора текста.

Создание окна редактора:

В Qt Creator выполняется автоматически при добавлении объекта в главное окно. Имя объекта задается в окне Свойств (в наших примерах — line). В программе line представляет собой указатель на окно редактора, поэтому обращение к методам класса использует оператор ->.

# Создание обработчика сигнала от кнопки

В Unix­подобных системах термин «сигнал» обозначает информацию о каком­либо событии. Сигналом может быть нажатие пользователем кнопки, выбор команды меню, перемещение мыши, срабатывание таймера и т.д. Источником сигнала служит какой­либо объект, например, при нажатии пользователем кнопки источником сигнала будет объект кнопка. Сигнал адресован другому объекту, который выполнит обработку сигнала, т.е. выполнит какие­либо действия. Функция, выполняющая обработку сигнала, называется слот. Для создания в Qt Creator слота надо перейти в режим дизайна, вызвать команду «Перейти к слоту...» из контекстного меню кнопки и выбрать стандартный сигнал clicked() (однократное нажатие кнопки мыши).

В результате в файле mainwindow.cpp будет создана пустая функция:

void MainWindow::on\_pushButton\_clicked()

{

}

В нее надо добавить код обработчика. Одновременно в mainwindow.h появится раздел слотов:

private slots: void on\_pushButton\_clicked();

#### **Сигналы и слоты**

Для коммуникации между объектами в Qt, существует специальная техника - сигналы и слоты. Сигнал вырабатывается, когда происходит определенное событие. Слот — это функция, которая вызывается в ответ на определенный сигнал. Виджеты Qt имеют много предопределенных сигналов и слотов, но мы всегда можем сделать дочерний класс и добавить наши сигналы и слоты в нем.

1. **Описание программы**

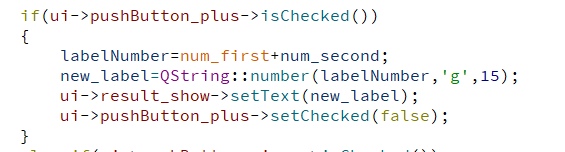
Данное приложение написана в среде Qt creator для Windows 10, представляющая собой приложение Калькулятор. В ней реализованы следующие функции:

- Математические (Сложение, вычитание, умножение, деление, извлечение корня, возведение в степень, вычисление процентов)

- Тригонометрические (Арксинус)

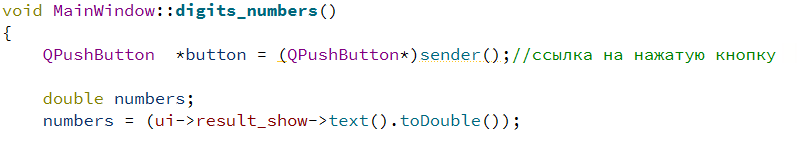
- Для работы с вводимыми данными (Равно, AC, +/-)\

Ввод чисел реализован с помощью клавиатуры на кнопках. При нажатии на любую из них, получаем ссылку на эту кнопку, считываем число с неё, добавляем его к уже введённому числу, преобразуем в QString и записываем в result\_show (QLabel)

Для выполнения математических операций при нажатии одной из кнопок, считывается введённое число, запоминается нажатая операция, и после ввода 2-го числа и нажатия на “=” выполняется нажатая операция (после нажатия на одну из кнопок математических операций, эта операция фиксируется с помощью изменения значения функции, и при нажатии на “=”, в зависимости от того какая операция зафиксирована, она выполняется) с 2-мя введёнными числами, после чего полученное число помещается в result\_show, и мы можем проводить с ним другие операции.

Для обработки ошибок используется конструкция try catch

Например:



Ввели в поле данные типа string, и преобразовали его в тип double

1. **Результаты работы программы**

На рисунках ниже представлена информация по вызову различных функций программы:

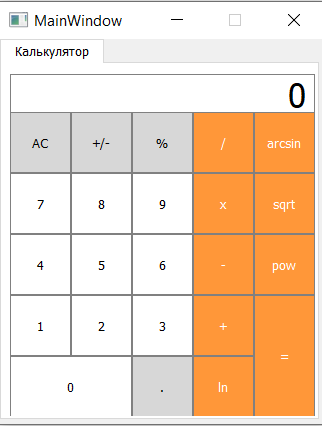


Рис.1 Основной экран калькулятора

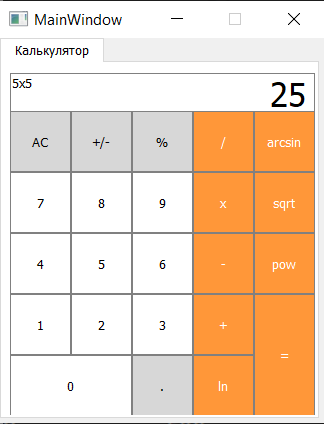


Рис.2 Пример работы функции “умножение”

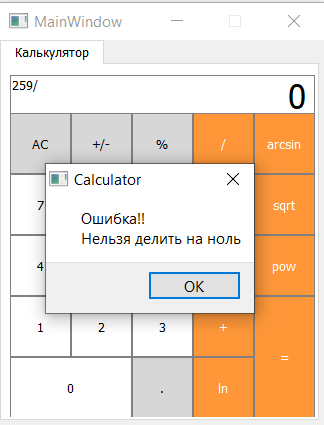


Рис. 3 Пример работы функции “Деление”

1. **Выводы:**
2. Освоена среда Qt creator
3. Изучены методы работы с классами в программе Qt creator
4. Разработана программа “калькулятор”, на базе Qt creator
5. **Код программы**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QMessageBox>

#include <QKeyEvent>

#include <cmath>

double num\_first=0;

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

ui->pushButton\_plus->setCheckable(true);

ui->pushButton\_minus->setCheckable(true);

ui->pushButton\_mult->setCheckable(true);

ui->pushButton\_div->setCheckable(true);

connect(ui->pushButton\_0,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_1,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_2,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_3,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_4,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_5,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_6,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_7,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_8,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_9,SIGNAL(clicked()),this,SLOT(digits\_numbers()));

connect(ui->pushButton\_change,SIGNAL(clicked()),this,SLOT(operations()));

connect(ui->pushButton\_procent,SIGNAL(clicked()),this,SLOT(operations()));

connect(ui->pushButton\_plus,SIGNAL(clicked()),this,SLOT(math\_operations()));

connect(ui->pushButton\_minus,SIGNAL(clicked()),this,SLOT(math\_operations()));

connect(ui->pushButton\_mult,SIGNAL(clicked()),this,SLOT(math\_operations()));

connect(ui->pushButton\_div,SIGNAL(clicked()),this,SLOT(math\_operations()));

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**digits\_numbers**()

{

QPushButton \*button = (QPushButton\*)sender();//ссылка на нажатую кнопку

double numbers;

numbers = (ui->result\_show->text().toDouble());

try

{

if(!( ui->result\_show->text().size()>14) && !(ui->result\_show->text().contains("e")))

{

QString new\_label;

if(ui->result\_show->text().contains(".") && button->text()=="0")

{

new\_label=ui->result\_show->text() + button ->text();

}

else

{

numbers=(ui->result\_show->text() + button ->text()).toDouble();

new\_label=QString::number(numbers,'g',10);

}

ui->result\_show->setText(new\_label);

}

else

{

throw numbers;

}

}

catch (double a)

{

QMessageBox msgBox;

msgBox.setText("Ошибка!!\nЧисло слишком длинное");

msgBox.*exec*();

}

}

void MainWindow::**operations**()

{

QPushButton \*button = (QPushButton\*)sender();//ссылка на нажатую кнопку

double numbers;

QString new\_label;

if(button->text() == "+/-")

{

numbers=(ui->result\_show->text()).toDouble();

numbers= numbers\*(-1);

new\_label=QString::number(numbers,'g',15);

ui->result\_show->setText(new\_label);

}

else if(button->text()=="%")

{

numbers=(ui->result\_show->text()).toDouble();

numbers= numbers\*(0.01);

new\_label=QString::number(numbers,'g',15);

ui->result\_show->setText(new\_label);

}

}

void MainWindow::**on\_pushButton\_dot\_clicked**()

{

if(!(ui->result\_show->text().contains('.')))

ui->result\_show->setText(ui->result\_show->text() + ".");

}

void MainWindow::**math\_operations**()

{

QPushButton \*button = (QPushButton\*)sender();//ссылка на нажатую кнопку

num\_first=ui->result\_show->text().toDouble();

ui->result\_show->setText("");

ui->EnterLabel->setText("");

ui->pushButton\_plus->setChecked(false);

ui->pushButton\_minus->setChecked(false);

ui->pushButton\_mult->setChecked(false);

ui->pushButton\_div->setChecked(false);

ui->EnterLabel->setText(QString::number(num\_first,'g',15)+button->text());

button->setChecked(true);

}

void MainWindow::**on\_pushButton\_clear\_clicked**()

{

ui->pushButton\_plus->setChecked(false);

ui->pushButton\_minus->setChecked(false);

ui->pushButton\_mult->setChecked(false);

ui->pushButton\_div->setChecked(false);

ui->EnterLabel->setText("");

ui->result\_show->setText("0");

}

void MainWindow::**on\_pushButton\_res\_clicked**()

{

try

{

double labelNumber,num\_second;

QString new\_label;

num\_second=ui->result\_show->text().toDouble();

if(ui->pushButton\_plus->isChecked())

{

labelNumber=num\_first+num\_second;

new\_label=QString::number(labelNumber,'g',15);

ui->result\_show->setText(new\_label);

ui->pushButton\_plus->setChecked(false);

}

else if(ui->pushButton\_minus->isChecked())

{

labelNumber=num\_first-num\_second;

new\_label=QString::number(labelNumber,'g',15);

ui->result\_show->setText(new\_label);

ui->pushButton\_minus->setChecked(false);

}

else if(ui->pushButton\_mult->isChecked())

{

labelNumber=num\_first\*num\_second;

new\_label=QString::number(labelNumber,'g',15);

ui->result\_show->setText(new\_label);

ui->pushButton\_mult->setChecked(false);

}

else if(ui->pushButton\_div->isChecked())

{

if(num\_second==0)

{

throw 2;

ui->result\_show->setText("Error");

}

else

{

labelNumber=num\_first/num\_second;

new\_label=QString::number(labelNumber,'g',15);

ui->result\_show->setText(new\_label);

}

ui->pushButton\_div->setChecked(false);

}

ui->EnterLabel->setText(ui->EnterLabel->text()+QString::number(num\_second,'g',15));

}

catch (int a)

{

switch(a)

{

case 1:

{

QMessageBox msgBox;

msgBox.setText("Ошибка!!");

msgBox.*exec*();

break;

}

case 2:

{

QMessageBox msgBox;

msgBox.setText("Ошибка!!\nНельзя делить на ноль");

msgBox.*exec*();

break;

}

}

}

}

void MainWindow::**on\_pushButton\_ln\_clicked**()

{

double number=ui->result\_show->text().toDouble();

number=log(number);

QString new\_label=QString::number(number,'g',15);

ui->result\_show->setText(new\_label);

}

void MainWindow::**on\_pushButton\_arcsin\_clicked**()

{

double number=ui->result\_show->text().toDouble();

number=asin(number);

QString new\_label=QString::number(number,'g',15);

ui->result\_show->setText(new\_label);

}

void MainWindow::**on\_pushButton\_sqrt\_clicked**()

{

double number=ui->result\_show->text().toDouble();

number=sqrt(number);

QString new\_label=QString::number(number,'g',15);

ui->result\_show->setText(new\_label);

}

void MainWindow::**on\_pushButton\_pow\_clicked**()

{

try

{

double number=ui->result\_show->text().toDouble();

number=pow(number,2.0);

QString new\_label=QString::number(number,'g',15);

if(new\_label.contains("inf"))

{

throw number;

}

else

{

ui->result\_show->setText(new\_label);

}

}

catch (double a)

{

QMessageBox msgBox;

msgBox.setText("Ошибка вычисления числа\nПолученное число слишком большое");

msgBox.*exec*();

}

}