



**Цель работы:** Написать программу-калькулятор, позволяющую выполнять простейшие математические действия: 

* Сложение
* Вычитание
* Умножение
* Деление
* Возведение в произвольную степень
* Взятие корня произвольной степени
* Синус
* Косинус
* Тангенс
* Логарифм

При вычислении должны обрабатываться все соответствующие ошибки и должны быть соответствующие сообщения об ошибках, а не просто возможность выполнения вычисления.

**Код программы:**

Файл main.cpp:

#include "calculator.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

calculator w;

w.show();

return a.exec();

}

Файл calculator.cpp:

#include "calculator.h"

#include "ui\_calculator.h"

#include <QGridLayout>

#include <QLineEdit>

#include <QPushButton>

#include "action.h"

calculator::calculator(QWidget \*parent): QWidget(parent), ui(new Ui::calculator)

{

this->actionObject = new action();

QGridLayout\* layout = new QGridLayout();

QLineEdit\* screen = new QLineEdit();

screen->setAlignment(Qt::AlignRight);

screen->setText("0");

QFont font = screen->font();

font.setPointSize(15);

screen->setFont(font);

screen->setReadOnly(true);

QPushButton \*btn0 = new QPushButton("0");

QPushButton \*btn1 = new QPushButton("1");

QPushButton \*btn2 = new QPushButton("2");

QPushButton \*btn3 = new QPushButton("3");

QPushButton \*btn4 = new QPushButton("4");

QPushButton \*btn5 = new QPushButton("5");

QPushButton \*btn6 = new QPushButton("6");

QPushButton \*btn7 = new QPushButton("7");

QPushButton \*btn8 = new QPushButton("8");

QPushButton \*btn9 = new QPushButton("9");

QPushButton \*btnEquals = new QPushButton("=");

QPushButton \*btnSqrt = new QPushButton("Sqrt");

QPushButton \*btnC = new QPushButton("C");

QPushButton \*btnCE = new QPushButton("CE");

QPushButton \*btnPlus = new QPushButton("+");

QPushButton \*btnMinus = new QPushButton("-");

QPushButton \*btnDivision = new QPushButton("/");

QPushButton \*btnMultiplexion = new QPushButton("\*");

QPushButton \*btnPow = new QPushButton("x^n");

QPushButton \*btnSin = new QPushButton("sin");

QPushButton \*btnCos = new QPushButton("cos");

QPushButton \*btnTan = new QPushButton("tan");

QPushButton \*btnLog = new QPushButton("log");

layout->addWidget(screen, 0, 0, 1, 6);

layout->addWidget(btn7, 1, 0);

layout->addWidget(btn8, 1, 1);

layout->addWidget(btn9, 1, 2);

layout->addWidget(btn4, 2, 0);

layout->addWidget(btn5, 2, 1);

layout->addWidget(btn6, 2, 2);

layout->addWidget(btn1, 3, 0);

layout->addWidget(btn2, 3, 1);

layout->addWidget(btn3, 3, 2);

layout->addWidget(btn0, 4, 0);

layout->addWidget(btnEquals, 4, 1, 1, 2);

layout->addWidget(btnSqrt, 1, 3);

layout->addWidget(btnPow, 1, 4);

layout->addWidget(btnC, 2, 3);

layout->addWidget(btnCE, 2, 4);

layout->addWidget(btnMinus, 3, 3);

layout->addWidget(btnDivision, 3, 4);

layout->addWidget(btnPlus, 4, 3);

layout->addWidget(btnMultiplexion, 4, 4);

layout->addWidget(btnSin, 1, 5);

layout->addWidget(btnCos, 2, 5);

layout->addWidget(btnTan, 3, 5);

layout->addWidget(btnLog, 4, 5);

QObject::connect(btn0, &QPushButton::clicked, this->actionObject, &action::init0);

QObject::connect(btn1, &QPushButton::clicked, this->actionObject, &action::init1);

QObject::connect(btn2, &QPushButton::clicked, this->actionObject, &action::init2);

QObject::connect(btn3, &QPushButton::clicked, this->actionObject, &action::init3);

QObject::connect(btn4, &QPushButton::clicked, this->actionObject, &action::init4);

QObject::connect(btn5, &QPushButton::clicked, this->actionObject, &action::init5);

QObject::connect(btn6, &QPushButton::clicked, this->actionObject, &action::init6);

QObject::connect(btn7, &QPushButton::clicked, this->actionObject, &action::init7);

QObject::connect(btn8, &QPushButton::clicked, this->actionObject, &action::init8);

QObject::connect(btn9, &QPushButton::clicked, this->actionObject, &action::init9);

QObject::connect(btnPlus, &QPushButton::clicked, this->actionObject, &action::initPlus);

QObject::connect(btnMinus, &QPushButton::clicked, this->actionObject, &action::initMinus);

QObject::connect(btnMultiplexion, &QPushButton::clicked, this->actionObject, &action::initMultiplexion);

QObject::connect(btnDivision, &QPushButton::clicked, this->actionObject, &action::initDivision);

QObject::connect(btnEquals, &QPushButton::clicked, this->actionObject, &action::initEquals);

QObject::connect(btnCE, &QPushButton::clicked, this->actionObject, &action::initCE);

QObject::connect(btnC, &QPushButton::clicked, this->actionObject, &action::initC);

QObject::connect(btnSqrt, &QPushButton::clicked, this->actionObject, &action::initSqrt);

QObject::connect(btnSin, &QPushButton::clicked, this->actionObject, &action::initSin);

QObject::connect(btnCos, &QPushButton::clicked, this->actionObject, &action::initCos);

QObject::connect(btnTan, &QPushButton::clicked, this->actionObject, &action::initTan);

QObject::connect(btnLog, &QPushButton::clicked, this->actionObject, &action::initLog);

QObject::connect(btnPow, &QPushButton::clicked, this->actionObject, &action::initPow);

QObject::connect(this->actionObject, &action::updateScreen, screen, &QLineEdit::setText);

this->setMaximumWidth(150);

this->setMaximumHeight(350);

this->setLayout(layout);

ui->setupUi(this);

}

calculator::~calculator()

{

delete ui;

}

Файл calculator.h:

#ifndef CALCULATOR\_H

#define CALCULATOR\_H

#include <QWidget>

#include "action.h"

QT\_BEGIN\_NAMESPACE

namespace Ui { class calculator; }

QT\_END\_NAMESPACE

class calculator : public QWidget

{

Q\_OBJECT

public:

calculator(QWidget \*parent = nullptr);

~calculator();

private:

Ui::calculator \*ui;

action\* actionObject;

};

#endif // CALCULATOR\_H

Файл action.cpp:

#include "action.h"

action::action(QObject \*parent) : QObject(parent)

{

}

Файл action.h:

#ifndef ACTION\_H

#define ACTION\_H

#include <QObject>

#include <cmath>

class action : public QObject

{

Q\_OBJECT

public:

explicit action(QObject \*parent = nullptr);

private:

double firstNum=0, secondNum=0;

char operation=NULL;

public slots:

void init0() {

firstNum=firstNum\*10;

emit updateScreen(QString::number(firstNum));

}

void init1() {

firstNum=firstNum\*10+1;

emit updateScreen(QString::number(firstNum));

}

void init2() {

firstNum=firstNum\*10+2;

emit updateScreen(QString::number(firstNum));

}

void init3() {

firstNum=firstNum\*10+3;

emit updateScreen(QString::number(firstNum));

}

void init4() {

firstNum=firstNum\*10+4;

emit updateScreen(QString::number(firstNum));

}

void init5() {

firstNum=firstNum\*10+5;

emit updateScreen(QString::number(firstNum));

}

void init6() {

firstNum=firstNum\*10+6;

emit updateScreen(QString::number(firstNum));

}

void init7() {

firstNum=firstNum\*10+7;

emit updateScreen(QString::number(firstNum));

}

void init8() {

firstNum=firstNum\*10+8;

emit updateScreen(QString::number(firstNum));

}

void init9() {

firstNum=firstNum\*10+9;

emit updateScreen(QString::number(firstNum));

}

void initPlus(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='+';

emit updateScreen("");

}

void initMinus(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='-';

emit updateScreen("");

}

void initMultiplexion(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='\*';

emit updateScreen("");

}

void initDivision(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='/';

emit updateScreen("");

}

void initEquals(){

if(operation=='+')

firstNum+=secondNum;

else if(operation=='-')

firstNum=secondNum-firstNum;

else if(operation=='\*')

firstNum\*=secondNum;

else if(operation=='/')

firstNum=secondNum/firstNum;

else if (operation=='s')//sqrt

firstNum=pow(secondNum, 1/firstNum);

else if (operation=='l')//log

firstNum=log(secondNum)/log(firstNum);

else if (operation=='p')//pow

firstNum=pow(firstNum, secondNum);

secondNum=0;

operation=NULL;

emit updateScreen(QString::number(firstNum));

}

void initCE(){

firstNum=0;

emit updateScreen(QString::number(firstNum));

}

void initC(){

operation=NULL;

firstNum=0;

secondNum=0;

emit updateScreen("0");

}

void initSin(){

firstNum=sin(firstNum);

emit updateScreen(QString::number(firstNum));

}

void initCos(){

firstNum=cos(firstNum);

emit updateScreen(QString::number(firstNum));

}

void initTan(){

firstNum=tan(firstNum);

emit updateScreen(QString::number(firstNum));

}

void initSqrt(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='s';

emit updateScreen("");

}

void initLog(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='l';

emit updateScreen("");

}

void initPow(){

if(operation==NULL){

secondNum=firstNum;

firstNum=0;

}

operation='p';

emit updateScreen("");

}

signals:

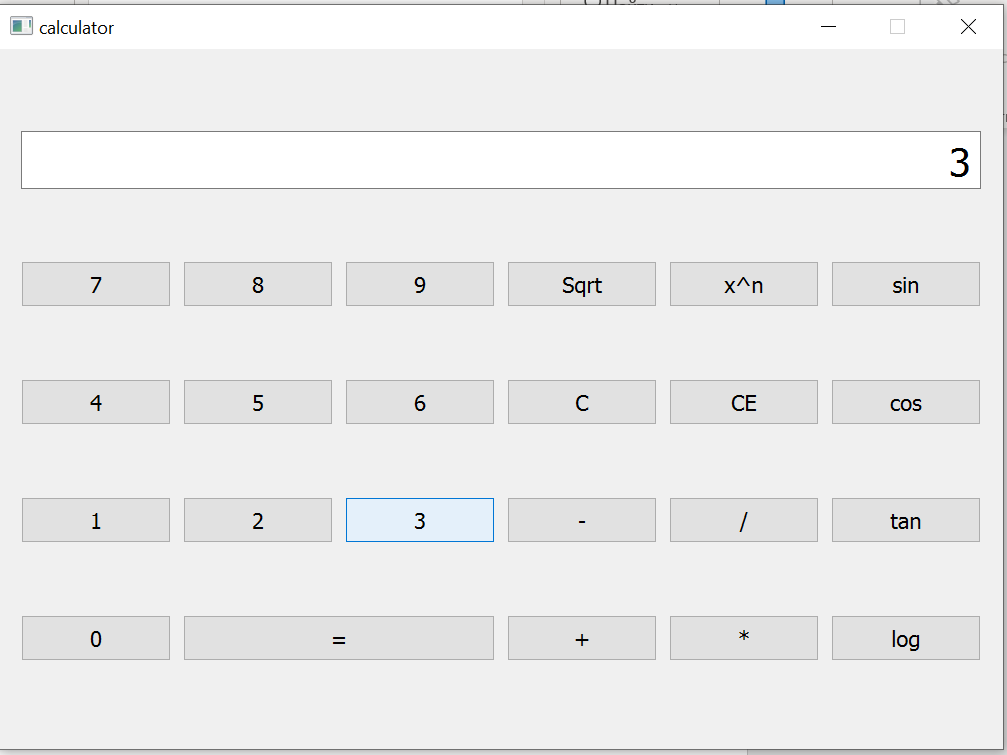
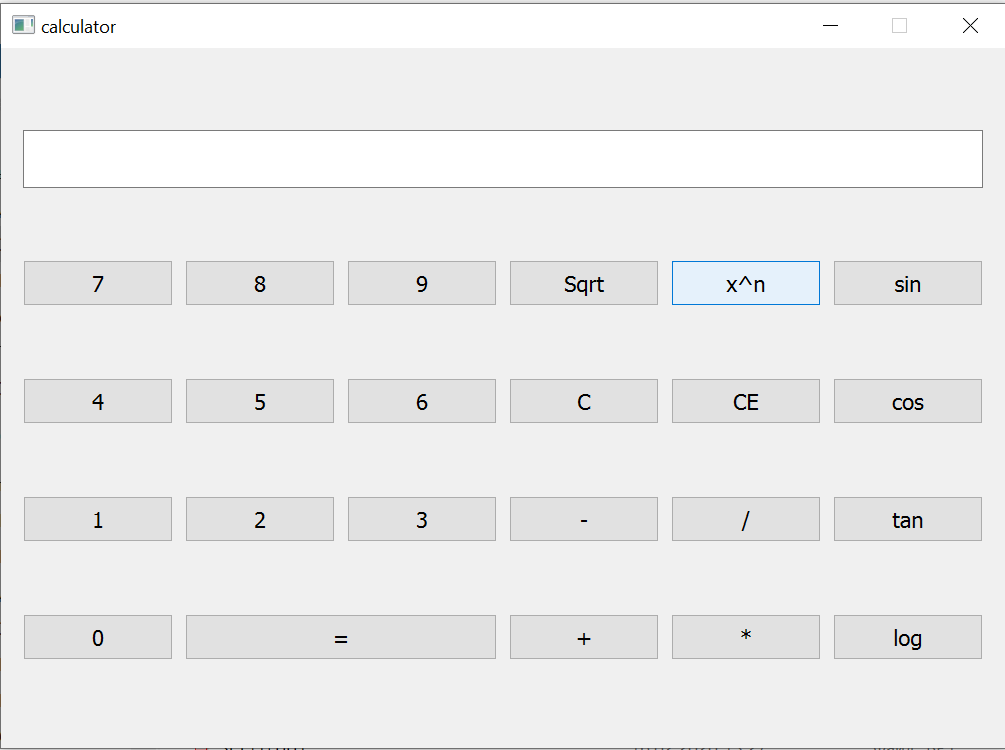
void updateScreen(QString result);

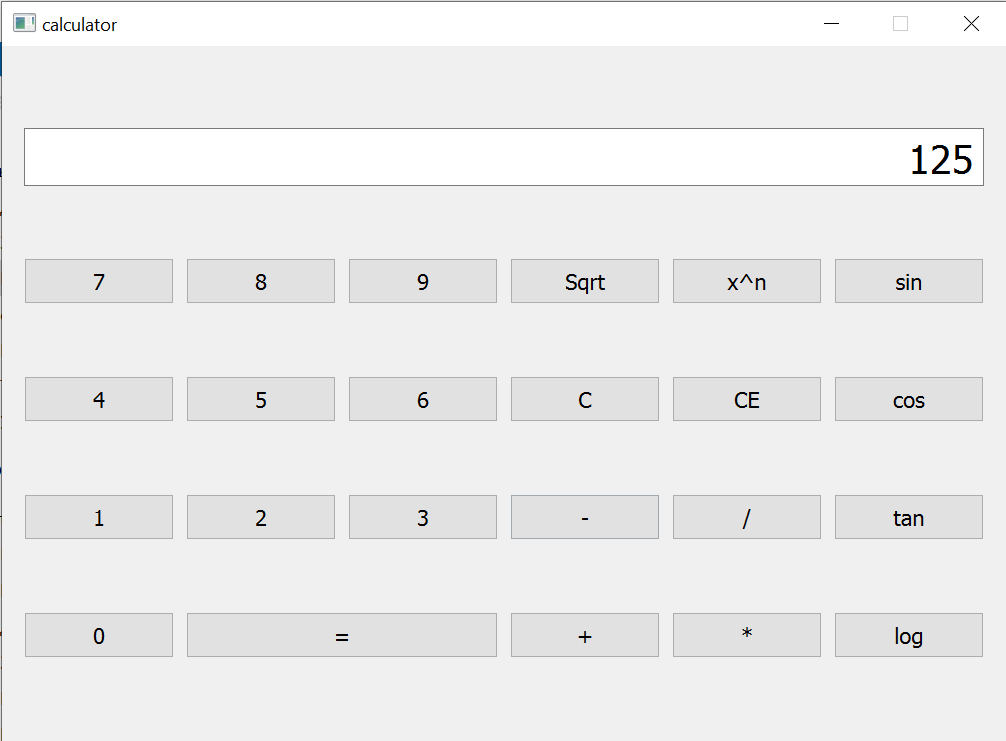
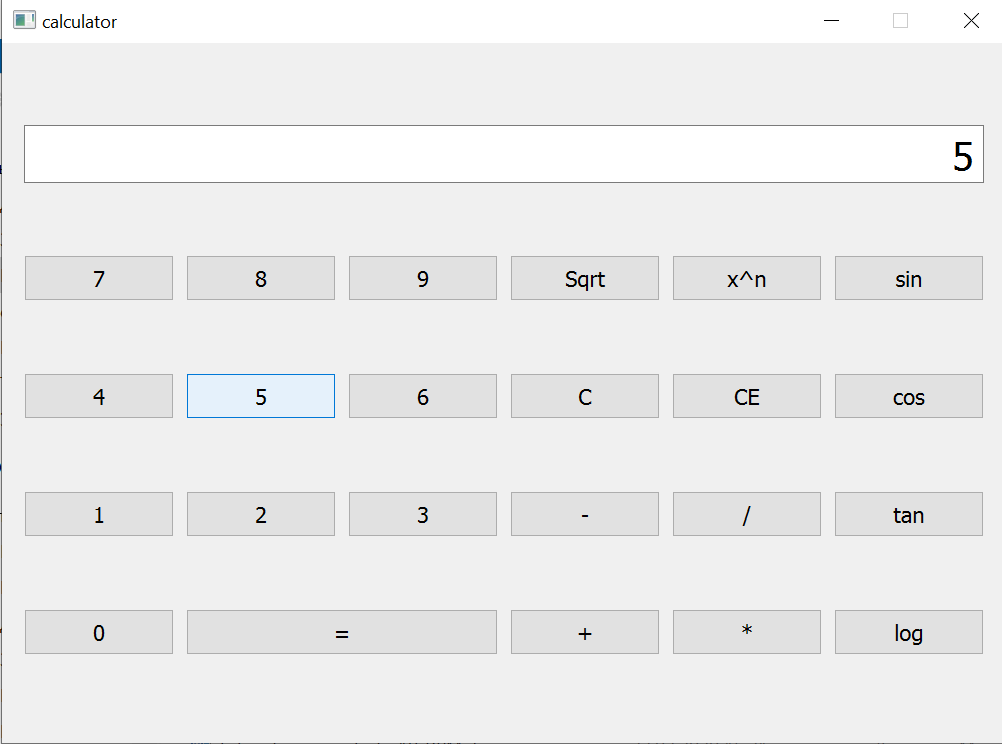
};

#endif // ACTION\_H

**Протокол программы:**

На рисунке 1 представлен результат работы программы.

** **

**

*Рисунок 1 - Результат работы программы*

**Вывод:** в ходе лабораторной работы были изучены основы разработки оконных приложений на примере Qt.