**Пермский национальный исследовательский  
политехнический университет**

**О Т Ч Ё Т**

**по творческой работе №1**

Калькулятор

Дисциплина: информатика

Вариант: 18

Выполнил работу

студент группы РИС-20-1б

Копейкин Дмитрий Евгеньевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

Пермь, 2021

**Постановка задачи**

1. Разработать код программы калькулятора в соответствии с заданием.
2. Реализовать алгоритм в виде программы на языке С++.
3. Разработать интерфейс средствами Qt.

Разработка калькулятора с Польской записью:

Калькулятор вычисляет выражения в польской инверсной записи. Здесь вначале вводятся 2 действительных числа – операнда, а затем вводится знак операции и выдаётся ответ. Допустимые операции: 1. Сложение 2. Вычитание 3. Умножение 4. Деление 5. Возведение в целую степень На панели калькулятора отображаются исходные данные и результаты, операции и служебные

клавиши, клавиши изменения знака числа. Калькулятор должен позволять корректировать последнее введённое число и выдавать сообщения об ошибках.

**Анализ задачи**

Сокрытие рамки и фона окна для того, чтобы калькулятор выглядел стильно

w.setWindowFlags(Qt::FramelessWindowHint);

w.setAttribute((Qt::WA\_TranslucentBackground));

Установка коннектов для кнопок от 0 до 9, чтобы уменьшить объем кода

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

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

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

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

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

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

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

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

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

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

Функция ввода числа

void MainWindow::enter\_numbers()

{

QPushButton \*button = (QPushButton\*)sender();

QString lab;

num = (lab\_next+button->text()).toDouble();

lab = QString::number(num,'g',14);

lab\_next = lab\_next+button->text();

count\_lab\_next++;

if (button->text() == '0' && lab\_next.indexOf('.') != -1)

ui->label->setText(lab\_prev+lab\_next);

else

ui->label->setText(lab\_prev+lab);

res = 1;

}

Функция кнопки «Равно»

void MainWindow::on\_pushButton\_result\_clicked()

{

if (count\_stack != 1)

{

QMessageBox msgBox;

msgBox.setText("Ошибка!!!");

msgBox.exec();

}

else

{

lab\_next = "";

count\_lab\_next = 0;

lab\_prev = "";

ui->label->setText(QString::number(steck.top(), 'g', 14));

steck.pop();

count\_stack = 0;

check\_dot = 1;

}

}

Функция кнопки «Ресет»

void MainWindow::on\_pushButton\_reset\_clicked()

{

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

for (int i=1; i<=count\_stack; i++)

steck.pop();

count\_stack = 0;

lab\_prev = "";

lab\_next = "";

count\_lab\_next = 0;

res = 0;

check\_dot = 1;

}

Функция ввода числа

void MainWindow::on\_pushButton\_enter\_clicked()

{

steck.push(num);

count\_stack++;

lab\_prev = lab\_prev + lab\_next + " ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

check\_dot = 1;

num = 0;

}

Функция кнопки «Плюс»

void MainWindow::on\_pushButton\_plus\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для сложения!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " + ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_1 + top\_2);

}

}

Функция кнопки «Минус»

void MainWindow::on\_pushButton\_minus\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для вычитания!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " - ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 - top\_1);

}

}

Функция кнопки «Деление»

void MainWindow::on\_pushButton\_divide\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для деления!");

msgBox.exec();

} else

{

if (steck.top()==0)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Попытка деления на ноль!");

msgBox.exec();

}

else

{

lab\_prev = lab\_prev + lab\_next + " / ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 / top\_1);

}

}

}

Функция кнопки «Умножение»

void MainWindow::on\_pushButton\_mult\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для умножения!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " x ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 \* top\_1);

}

}

Функция кнопки «Редактирование»

void MainWindow::on\_pushButton\_edit\_clicked()

{

if (count\_lab\_next != 0)

{

count\_lab\_next--;

lab\_next = lab\_next.left(count\_lab\_next);

ui->label->setText(lab\_prev+lab\_next);

if (lab\_next.endsWith('.')) check\_dot = 1;

}

}

Функция добавления точки

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

{

if (check\_dot)

{

QString lab;

num = (lab\_next+'.').toDouble();

lab = QString::number(num,'g',14);

lab\_next = lab\_next+'.';

count\_lab\_next++;

ui->label->setText(lab\_prev+lab+'.');

res = 1;

check\_dot = 0;

}

}

Функция кнопки изменения знака числа

void MainWindow::on\_pushButton\_plus\_minus\_clicked()

{

num = num \* -1;

lab\_next = '-' + lab\_next;

ui->label->setText(lab\_prev+lab\_next);

}

Функция кнопки степень числа

void MainWindow::on\_pushButton\_stepen\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для возведения в степень!");

msgBox.exec();

} else

{

if (lab\_next.indexOf('.') != -1 || lab\_next.indexOf('-') != -1)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Степень должна быть натуральной!");

msgBox.exec();

}

else

{

lab\_prev = lab\_prev + lab\_next + " ^ ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(pow(top\_2, top\_1));

}

}

}

**Код**

**Mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private:

Ui::MainWindow \*ui;

private slots:

void enter\_numbers();

void on\_pushButton\_result\_clicked();

void on\_pushButton\_reset\_clicked();

void on\_pushButton\_enter\_clicked();

void on\_pushButton\_plus\_clicked();

void on\_pushButton\_minus\_clicked();

void on\_pushButton\_divide\_clicked();

void on\_pushButton\_mult\_clicked();

void on\_pushButton\_edit\_clicked();

void on\_pushButton\_dot\_clicked();

void on\_pushButton\_plus\_minus\_clicked();

void on\_pushButton\_stepen\_clicked();

void on\_pushButton\_clicked();

void on\_pushButton\_10\_clicked();

};

#endif // MAINWINDOW\_H

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.setWindowFlags(Qt::FramelessWindowHint);

w.setAttribute((Qt::WA\_TranslucentBackground));

w.show();

return a.exec();

}

**Mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "QString"

#include <stack>

#include "QMessageBox"

#include <cmath>

using namespace std;

stack <double> steck;

int count\_stack = 0;

int count\_lab\_next = 0;

double num;

bool res = 0;

bool check\_dot = 1;

bool size\_window = 1;

QString lab\_next, lab\_prev;

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

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

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

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

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

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

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

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

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

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

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

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::enter\_numbers()

{

QPushButton \*button = (QPushButton\*)sender();

QString lab;

num = (lab\_next+button->text()).toDouble();

lab = QString::number(num,'g',14);

lab\_next = lab\_next+button->text();

count\_lab\_next++;

if (button->text() == '0' && lab\_next.indexOf('.') != -1)

ui->label->setText(lab\_prev+lab\_next);

else

ui->label->setText(lab\_prev+lab);

res = 1;

}

void MainWindow::on\_pushButton\_result\_clicked()

{

if (count\_stack != 1)

{

QMessageBox msgBox;

msgBox.setText("Ошибка!!!");

msgBox.exec();

}

else

{

lab\_next = "";

count\_lab\_next = 0;

lab\_prev = "";

ui->label->setText(QString::number(steck.top(), 'g', 14));

steck.pop();

count\_stack = 0;

check\_dot = 1;

}

}

void MainWindow::on\_pushButton\_reset\_clicked()

{

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

for (int i=1; i<=count\_stack; i++)

steck.pop();

count\_stack = 0;

lab\_prev = "";

lab\_next = "";

count\_lab\_next = 0;

res = 0;

check\_dot = 1;

}

void MainWindow::on\_pushButton\_enter\_clicked()

{

steck.push(num);

count\_stack++;

lab\_prev = lab\_prev + lab\_next + " ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

check\_dot = 1;

num = 0;

}

void MainWindow::on\_pushButton\_plus\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для сложения!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " + ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_1 + top\_2);

}

}

void MainWindow::on\_pushButton\_minus\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для вычитания!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " - ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 - top\_1);

}

}

void MainWindow::on\_pushButton\_divide\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для деления!");

msgBox.exec();

} else

{

if (steck.top()==0)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Попытка деления на ноль!");

msgBox.exec();

}

else

{

lab\_prev = lab\_prev + lab\_next + " / ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 / top\_1);

}

}

}

void MainWindow::on\_pushButton\_mult\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для умножения!");

msgBox.exec();

} else

{

lab\_prev = lab\_prev + lab\_next + " x ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(top\_2 \* top\_1);

}

}

void MainWindow::on\_pushButton\_edit\_clicked()

{

if (count\_lab\_next != 0)

{

count\_lab\_next--;

lab\_next = lab\_next.left(count\_lab\_next);

ui->label->setText(lab\_prev+lab\_next);

if (lab\_next.endsWith('.')) check\_dot = 1;

}

}

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

{

if (check\_dot)

{

QString lab;

num = (lab\_next+'.').toDouble();

lab = QString::number(num,'g',14);

lab\_next = lab\_next+'.';

count\_lab\_next++;

ui->label->setText(lab\_prev+lab+'.');

res = 1;

check\_dot = 0;

}

}

void MainWindow::on\_pushButton\_plus\_minus\_clicked()

{

num = num \* -1;

lab\_next = '-' + lab\_next;

ui->label->setText(lab\_prev+lab\_next);

}

void MainWindow::on\_pushButton\_stepen\_clicked()

{

if (res)

{

steck.push(num);

count\_stack++;

res = 0;

}

if (count\_stack < 2)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Недостаточно чисел для возведения в степень!");

msgBox.exec();

} else

{

if (lab\_next.indexOf('.') != -1 || lab\_next.indexOf('-') != -1)

{

QMessageBox msgBox;

msgBox.setText("Ошибка! Степень должна быть натуральной!");

msgBox.exec();

}

else

{

lab\_prev = lab\_prev + lab\_next + " ^ ";

lab\_next = "";

count\_lab\_next = 0;

ui->label->setText(lab\_prev);

double top\_1 = steck.top();

steck.pop(); count\_stack--;

double top\_2 = steck.top();

steck.pop();

steck.push(pow(top\_2, top\_1));

}

}

}

void MainWindow::on\_pushButton\_clicked()

{

if (size\_window){

setGeometry(geometry().x(),geometry().y(),950,411);

ui->label\_2->setGeometry(ui->label\_2->geometry().x(), ui->label\_2->geometry().y(), 891, 371);

ui->pushButton\_10->setGeometry(857, 11, ui->pushButton\_10->geometry().width(), ui->pushButton\_10->geometry().height());

ui->label\_4->setGeometry(ui->label\_4->geometry().x(), ui->label\_4->geometry().y(), 891, 33);

ui->label\_5->setGeometry(ui->label\_5->geometry().x(), ui->label\_5->geometry().y(), 891, 21);

size\_window = 0;

}

else {

setGeometry(geometry().x(),geometry().y(),519,411);

ui->label\_2->setGeometry(ui->label\_2->geometry().x(), ui->label\_2->geometry().y(), 501, 371);

ui->pushButton\_10->setGeometry(469, 11, ui->pushButton\_10->geometry().width(), ui->pushButton\_10->geometry().height());

ui->label\_4->setGeometry(ui->label\_4->geometry().x(), ui->label\_4->geometry().y(), 501, 33);

ui->label\_5->setGeometry(ui->label\_5->geometry().x(), ui->label\_5->geometry().y(), 501, 21);

size\_window = 1;

}

}

void MainWindow::on\_pushButton\_10\_clicked()

{

qApp->quit();

}

**Скриншот**

