**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Дисциплина: «Технология программирования»

Лабораторная работа №9.

Тема: «Создание простейшего приложения с графическим интерфейсом на основе QT»

Выполнили: Студенты группы РЦТ-22

Балан К. А.

Орехва В. Э.

Принял: к.т.н, доцент кафедры ПИиВТ

Дагаев А. В.

Санкт-Петербург

2023

**Цель работы**

Познакомиться с объектно-ориентированной библиотекой Qt и ее классами, позволяющими построить простейшее приложение с графическим интерфейсом, с инструментальной средой разработки Qt creator.

**Постановка задачи**

Сконструировать в Qt creator приложение с графическим интерфейсом на основе Qt, выполняющее функцию калькулятора. Простейший вариант приложения представляет собой форму, имеющую два поля ввода и одно - для вывода.

Вычисления выполняются при нажатии пользователем соответствующей кнопки. Минимальный набор функций калькулятора: сложение, вычитание, умножение, деление.

**Описание программы**

Данная программа, разработанная в Qt Creator, представляет собой приложение-калькулятор с графическим пользовательским интерфейсом (GUI). Она выполняет стандартные математические операции, такие как сложение, вычитание, умножение и деление, а также предоставляет некоторые специализированные функции, например, вычисление процента, факториала, ряда Фибоначчи и логарифма.

**main.cpp:**

* Создает экземпляр QApplication, который является базовым классом для любого приложения с GUI в Qt.
* Создает экземпляр MainWindow, который является наследником QMainWindow и представляет основное окно приложения.
* Отображает главное окно приложения (w.show()).
* Запускает цикл обработки событий (a.exec()), который ожидает, например, нажатия кнопок или другие действия пользователя.

**mainwindow.cpp:**

* В конструкторе осуществляется настройка UI, созданного с помощью Qt Designer, и соединение сигналов кнопок с соответствующими слотами (функциями-обработчиками).
* У каждой кнопки цифр (от 0 до 9, включая десятичную точку) происходит привязка к слоту digits\_numbers(), который добавляет числа в дисплей калькулятора.
* Кнопки выполнения математических операций привязаны к слоту operation(), где каждая кнопка выполняет различные операции как унарные, так и бинарные - изменение знака числа, вычисление процента от числа, обращение числа (1/x), квадратный корень, факториал, числа Фибоначчи, логарифм.
* Кнопки арифметических операций (сложение, вычитание, умножение и деление) соединены с функцией math\_operation(), которая запоминает введенное число и выбранную операцию.
* В глобальной переменной double num\_first, хранится первое число для последующего выполнения бинарной арифметической операции.
* Функция on\_result\_clicked() выполняется при нажатии кнопки равно (=) и управляет последовательностью выполнения арифметических действий.
* Обработчики кнопок с id on\_pushButton\_11\_clicked() и т.д., в данной части кода содержат пустые реализации, но они могут быть использованы для различных функций калькулятора.

**mainwindow.h:**

* Объявляет класс MainWindow, который наследует QMainWindow.
* Внутри пространства имен Ui, которое является частью библиотеки Qt, находится объявление вложенного класса MainWindow, которое предоставляет доступ к элементам пользовательского интерфейса, созданного с помощью Qt Designer.
* Объявлены приватные функции (слоты), которые связаны с кнопками калькулятора.

Стоит отметить, что в данном коде множество слотов оставлены не связанными и не реализованными, что обычно означает, что разработка интерфейса еще не завершена или часть функций приложения еще не разработана.

**Описание классов**

1. *Класс QMainWindow:*

QMainWindow является базовым классом для создания главного окна приложения в Qt.

Он предоставляет основные функции и возможности для работы с оконным интерфейсом, такие как меню, панели инструментов, статусная строка и т. д.

Этот класс управляет распределением и отображением различных виджетов и элементов интерфейса в главном окне.

Кроме того, класс QMainWindow обрабатывает события, связанные с изменением размеров окна, закрытием, переключением между различными подокнами и т. д.

2. *Класс MainWindow:*

Класс MainWindow является производным от класса QMainWindow и обеспечивает специфические для приложения функции и элементы управления.

Он может содержать логику и интерфейс для работы конкретного приложения, такие как кнопки, текстовые поля, таблицы, компоненты ввода-вывода и т. д.

Класс MainWindow также может включать обработчики событий (слоты), которые реагируют на пользовательские действия, как, например, нажатие кнопок, изменение значений полей и т. д.

Оба класса MainWindow и QMainWindow позволяют разработчикам создавать главные окна приложений с графическими интерфейсами, управлять их отображением и взаимодействием с пользователем, и обрабатывать действия пользователей через события и сигнал.

**Описание слотов**

1. *digits\_numbers()*
   * Слот, который обрабатывает нажатия на цифровые кнопки калькулятора. Когда кнопка числа нажимается, он добавляет соответствующую цифру к отображаемому числу.
2. *operation()*
   * Слот, обрабатывающий нажатия на операционные кнопки (+/-, %, 1/x, √, n!, fib, lg). Он выполняет заданную одноместную операцию на числе, отображаемом в момент нажатия.
3. *math\_operation()*
   * Слот, который сохраняет текущее значение и инициализирует математическую операцию (такую как сложение, вычитание, умножение, деление или возведение в степень) при нажатии соответствующей кнопки.
4. *on\_pushButton\_19\_clicked()*
   * Слот, который выполняет двуместную операцию между сохраненным в памяти первым числом и текущим отображаемым числом, когда пользователь нажимает кнопку равенства.
5. *on\_pushButton\_13\_clicked(), on\_pushButton\_18\_clicked(), on\_pushButton\_11\_clicked(), ...*
   * Это шаблонные слоты, которые могут быть настроены для других действий, связанных с кнопками, но на данный момент в коде они не имеют реализации, поэтому они ничего не дела\

**Результаты работы программы**

На рис. 8.1 показан пример результаты вычислений 5+5

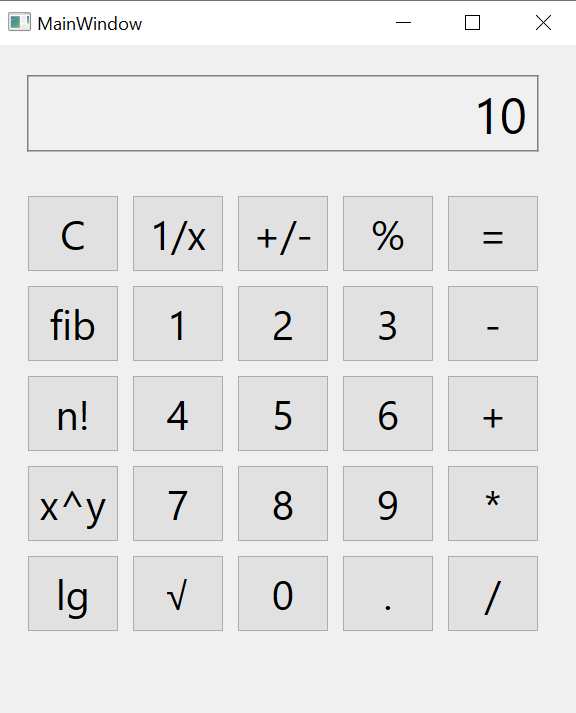


Рис. 8.1 Пример вычисления 5+5

На рис. 8.2 приведен пример расчёта результата вычислений 7^3

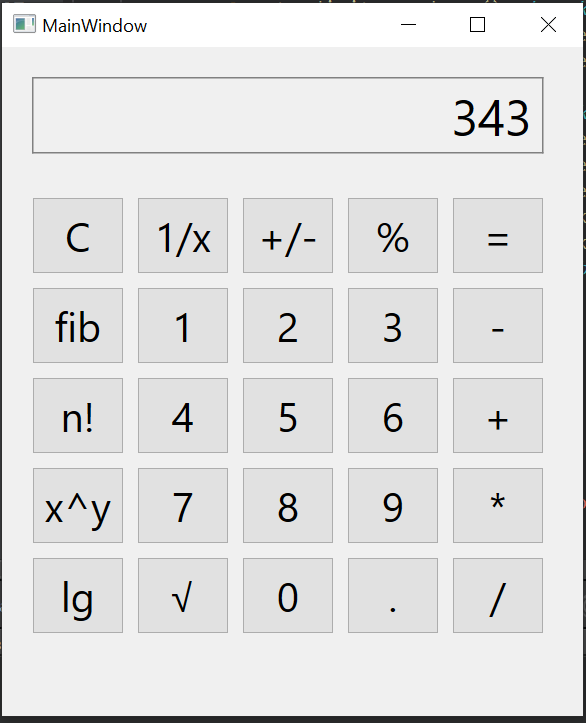


Рис. 8.2 Пример вычислений 7^3

**Вывод**

В ходе проделанной лабораторной работы получены следующие результаты:

Написали программу, которая представляет из себя калькулятор, написанный с использованием фреймворка Qt в среде разработки Qt Creator. Это графическое приложение для операционной системы, которое включает базовые функции калькулятора, такие как сложение, вычитание, умножение и деление, а также специфические функции:

* Ввод цифр и десятичной точки для формирования чисел.
* Операции с десятичными и отрицательными числами.
* Вычисление процента от числа.
* Обращение числа (вычисление значения "1/x").
* Извлечение квадратного корня.
* Вычисление факториала числа.
* Определение числа Фибоначчи.
* Логарифмирование по основанию 10.

Каждая кнопка на калькуляторе связана с функцией (слотом) через механизм сигналов и слотов Qt, который обеспечивает реакцию интерфейса на действия пользователя. Пользовательский интерфейс состоит из кнопок для ввода чисел и выполнения операций, а также метки (label) для отображения вводимых и вычисленных значений.

Калькулятор включает в себя возможность выбора математических операций, также он поддерживает выполнение операций над двумя числами (например, числом, сохраненным в памяти калькулятора, и числом на экране).

Программа написана на C++ с использованием объектно-ориентированных практик, характерных для использования Qt фреймворка.

Освоили практические и теоретические навыки написания программ с пользовательским интерфейсом в среде разработки QT Creator.

**Листинг**

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**Mainwindow.cpp**

#include "mainwindow.h"

#include "./ui\_mainwindow.h"

#include "math.h"

double num\_first;

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

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\_10,SIGNAL(clicked()), this,SLOT(digits\_numbers()));

connect(ui->pushButton\_12,SIGNAL(clicked()), this,SLOT(digits\_numbers()));

connect(ui->pushButton\_11,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->Convert,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->OnePercent,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->fib,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->Factorial,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->divisionByX,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->pushButton\_26,SIGNAL(clicked()), this,SLOT(operation()));

connect(ui->pow,SIGNAL(clicked()), this,SLOT(math\_operation()));

connect(ui->pushButton\_14,SIGNAL(clicked()), this,SLOT(math\_operation()));

connect(ui->pushButton\_15,SIGNAL(clicked()), this,SLOT(math\_operation()));

connect(ui->pushButton\_16,SIGNAL(clicked()), this,SLOT(math\_operation()));

connect(ui->pushButton\_17,SIGNAL(clicked()), this,SLOT(math\_operation()));

ui->pushButton\_14->setCheckable(true);

ui->pushButton\_15->setCheckable(true);

ui->pushButton\_16->setCheckable(true);

ui->pushButton\_17->setCheckable(true);

ui->pow->setCheckable(true);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::digits\_numbers(){

//Получаем свойства нажатой кнопки

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

// Строка для конвертирования значения в текст

QString new\_label;

if(button->text() == "."){ // Проверяем если была нажата кнопка с десятичной точкой

if(!(ui->label->text().contains('.'))) // Проверяем, содержит ли label уже точку

new\_label = ui->label->text() + ".";

} else {

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

}

ui->label->setText(new\_label);

}

void MainWindow::on\_pushButton\_13\_clicked()

{

if(!(ui->label->text().contains('.')))

ui->label->setText(ui->label->text() + ".");

}

void MainWindow::on\_pushButton\_18\_clicked()

{

ui->pushButton\_14->setCheckable(true);

ui->pushButton\_15->setCheckable(true);

ui->pushButton\_16->setCheckable(true);

ui->pushButton\_17->setCheckable(true);

ui->pow->setCheckable(true);

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

}

//Вычисления факториала рекурсией

double Factorial(double n)

{

if (n == 1 || n == 0) return 1;

return n \* Factorial(n - 1);

}

//Вычисления ряда фиббоначи

double fib(double n)

{

return n == 0 || n == 1 ? n : fib(n - 1) + fib(n - 2);

}

void MainWindow::operation(){

//Получаем свойства нажатой кнопки

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

QString new\_label;

double numbers;

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

//Конвертируем текст label в число

numbers = (ui->label->text().toDouble());

numbers = numbers \* -1;

//Конвертируем число в строку(15знаков после запятой)

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

ui->label->setText(new\_label);

}

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

numbers = (ui->label->text().toDouble());

numbers = numbers \* 0.01;

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

ui->label->setText(new\_label);

}

else if(button->text() == "1/x"){

numbers = (ui->label->text().toDouble());

numbers = 1/numbers;

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

ui->label->setText(new\_label);

}

else if(button->text() == "√"){

numbers = (ui->label->text().toDouble());

numbers = std::sqrt(numbers);

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

ui->label->setText(new\_label);

}

else if(button->text() == "n!"){

numbers = (ui->label->text().toDouble());

numbers = Factorial(numbers);

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

ui->label->setText(new\_label);

}

else if(button->text() == "fib"){

numbers = (ui->label->text().toDouble());

numbers = fib(numbers);

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

ui->label->setText(new\_label);

}

else if(button->text() == "lg"){

numbers = (ui->label->text().toDouble());

numbers = log10(numbers);

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

ui->label->setText(new\_label);

}

}

void MainWindow::math\_operation(){

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

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

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

button->setChecked(true);

}

void MainWindow::on\_pushButton\_19\_clicked()

{

double secondNum, labelNum;

QString newLabel;

secondNum = ui->label->text().toDouble();

if(ui->pushButton\_14->isChecked()){

labelNum = num\_first + secondNum;

newLabel = QString::number(labelNum, 'g',15);

ui->pushButton\_14->setChecked(false);

ui->label->setText(newLabel);

}

else if(ui->pushButton\_15->isChecked()){

labelNum = num\_first \* secondNum;

newLabel = QString::number(labelNum, 'g',15);

ui->pushButton\_15->setChecked(false);

ui->label->setText(newLabel);

}

else if(ui->pushButton\_16->isChecked()){

labelNum = num\_first - secondNum;

newLabel = QString::number(labelNum, 'g',15);

ui->pushButton\_16->setChecked(false);

ui->label->setText(newLabel);

}

else if(ui->pushButton\_17->isChecked()){

if(secondNum != 0){

labelNum = num\_first / secondNum;

newLabel = QString::number(labelNum, 'g',15);

ui->pushButton\_17->setChecked(false);

ui->label->setText(newLabel);

}else{

ui->label->setText("0");

}

}else if(ui->pow->isChecked()){

//x^y

labelNum = pow(num\_first, secondNum);

newLabel = QString::number(labelNum, 'g',15);

ui->pow->setChecked(false);

ui->label->setText(newLabel);

}

}

void MainWindow::on\_pushButton\_11\_clicked()

{

}

void MainWindow::on\_pushButton\_21\_clicked()

{

}

void MainWindow::on\_pushButton\_23\_clicked()

{

}

void MainWindow::on\_pushButton\_24\_clicked()

{

}

void MainWindow::on\_pushButton\_22\_clicked()

{

}

**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 digits\_numbers();

void on\_pushButton\_13\_clicked();

void on\_pushButton\_18\_clicked();

void operation();

void math\_operation();

void on\_pushButton\_19\_clicked();

void on\_pushButton\_11\_clicked();

void on\_pushButton\_12\_clicked();

void on\_pushButton\_21\_clicked();

void on\_pushButton\_23\_clicked();

void on\_pushButton\_24\_clicked();

void on\_pushButton\_22\_clicked();

};

#endif // MAINWINDOW\_H