МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный университет геодезии и картографии»

(МИИГАиК)

Факультет геоинформатики и информационной безопасности

Кафедра геоинформационных систем и технологий

**Лабораторная работа №1  
 «Разработка калькулятора с GUI на С++»**

Выполнил:

2023-ФГиИБ-ПИ-1б

Беспечалов Артём Алексанрович

Проверил:

Лебедев Евгений Денисович

Москва 2023

Оглавление

[Глава 1. Разработка интерфейса 3](#_Toc152793583)

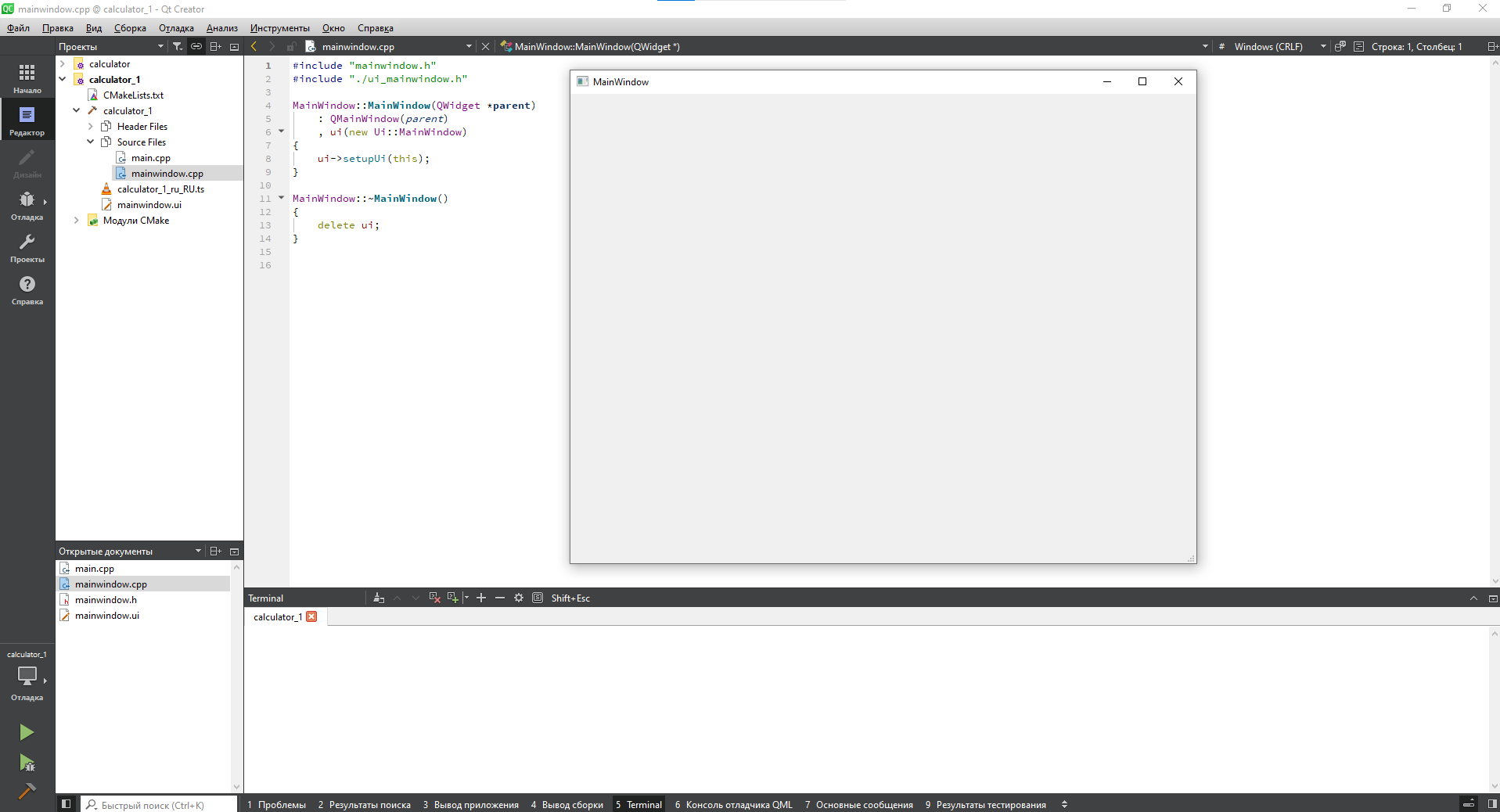
[Глава 2. Разработка калькулятора. 16](#_Toc152793584)

[Глава 3. Листинг “mainwindow.cpp”: 17](#_Toc152793585)

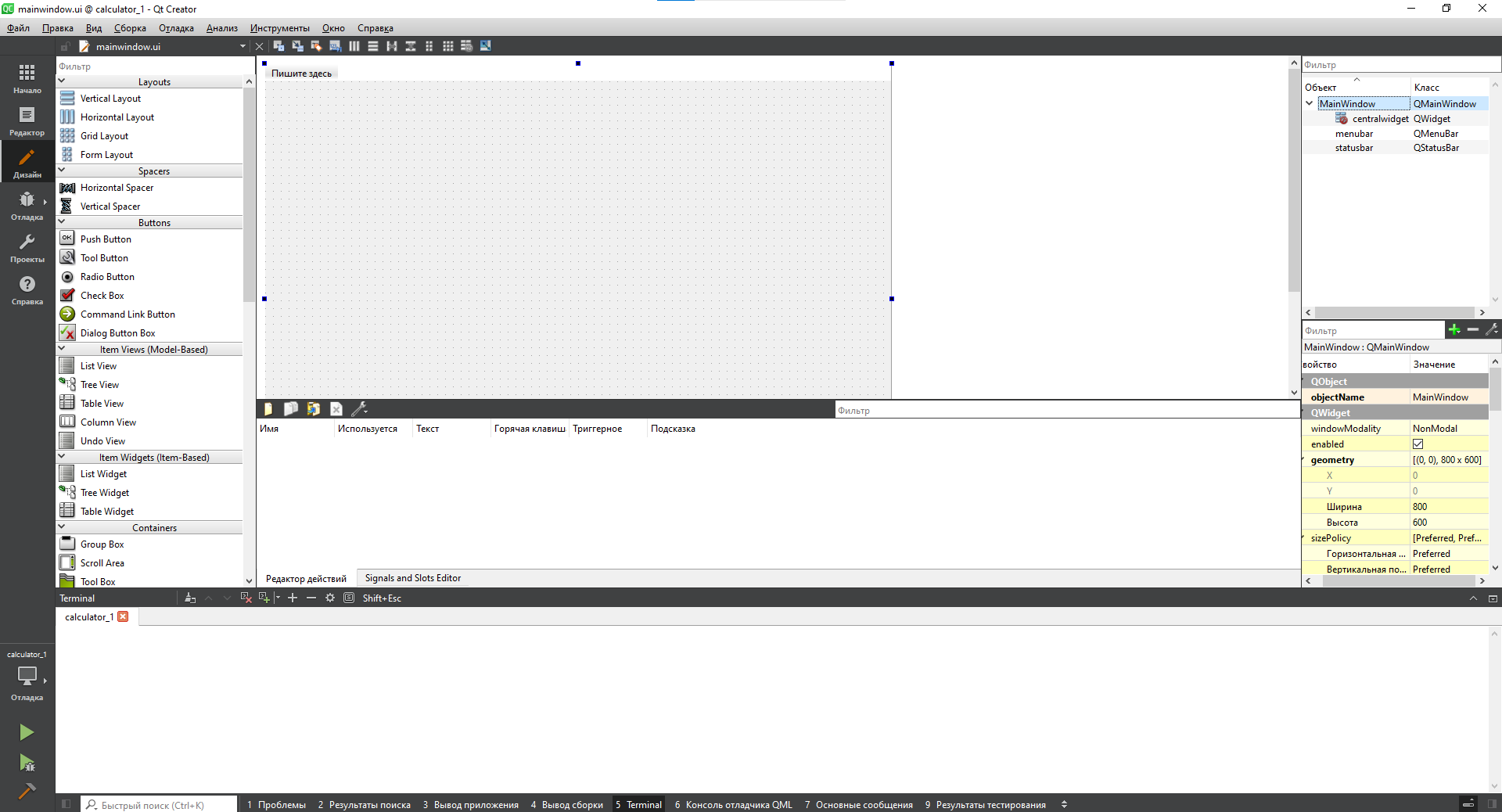
[Глава 4. Листинг “mainwindow.h”: 22](#_Toc152793586)

[Тестирование. 24](#_Toc152793587)

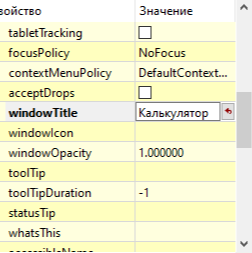
# Глава 1. Разработка интерфейса

****

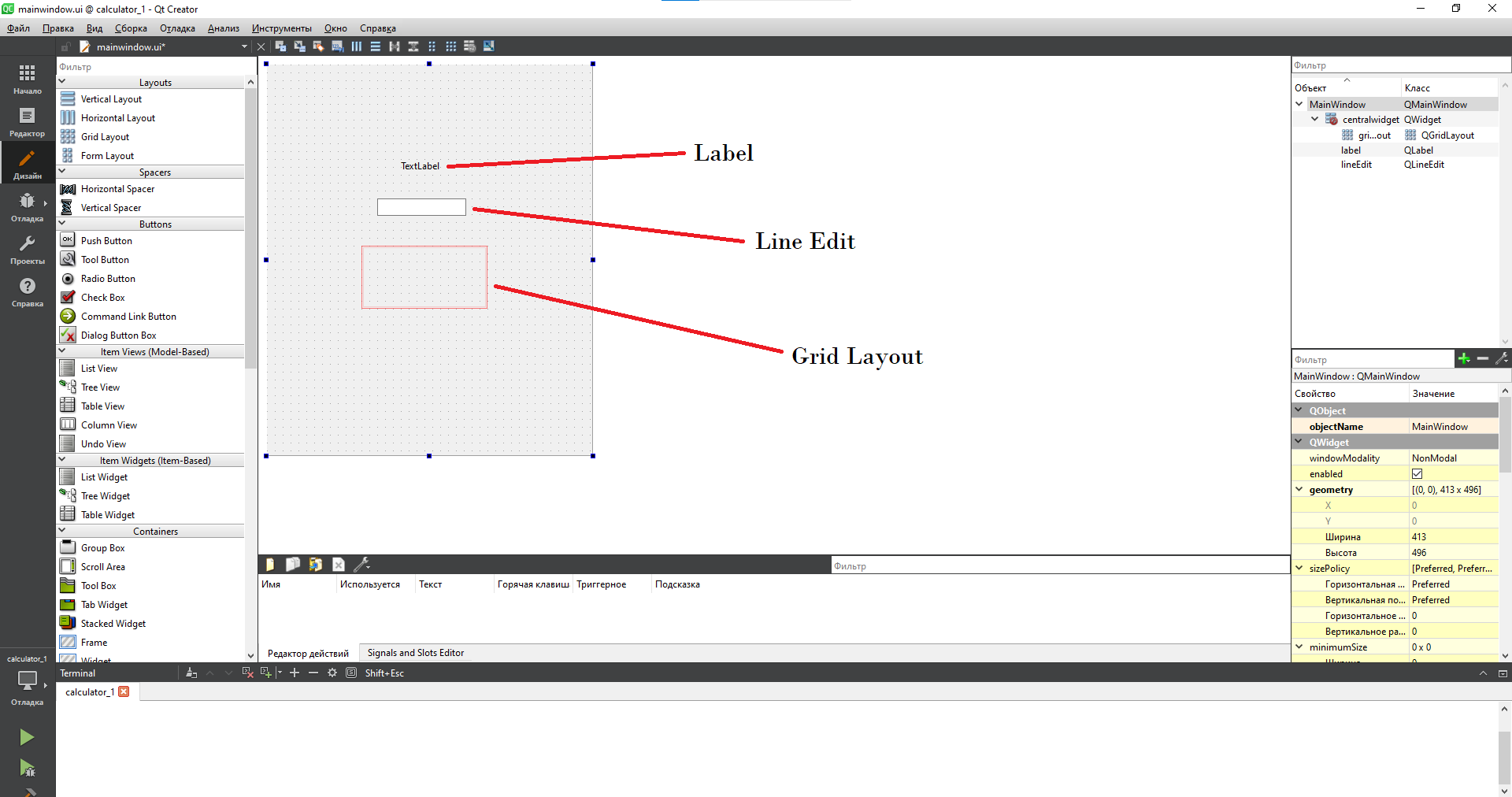
Запустим редактор интерфейса при помощи открытия файла “mainwindow.cpp” и нажав сочетание клавиш “SHIFT + F4”.

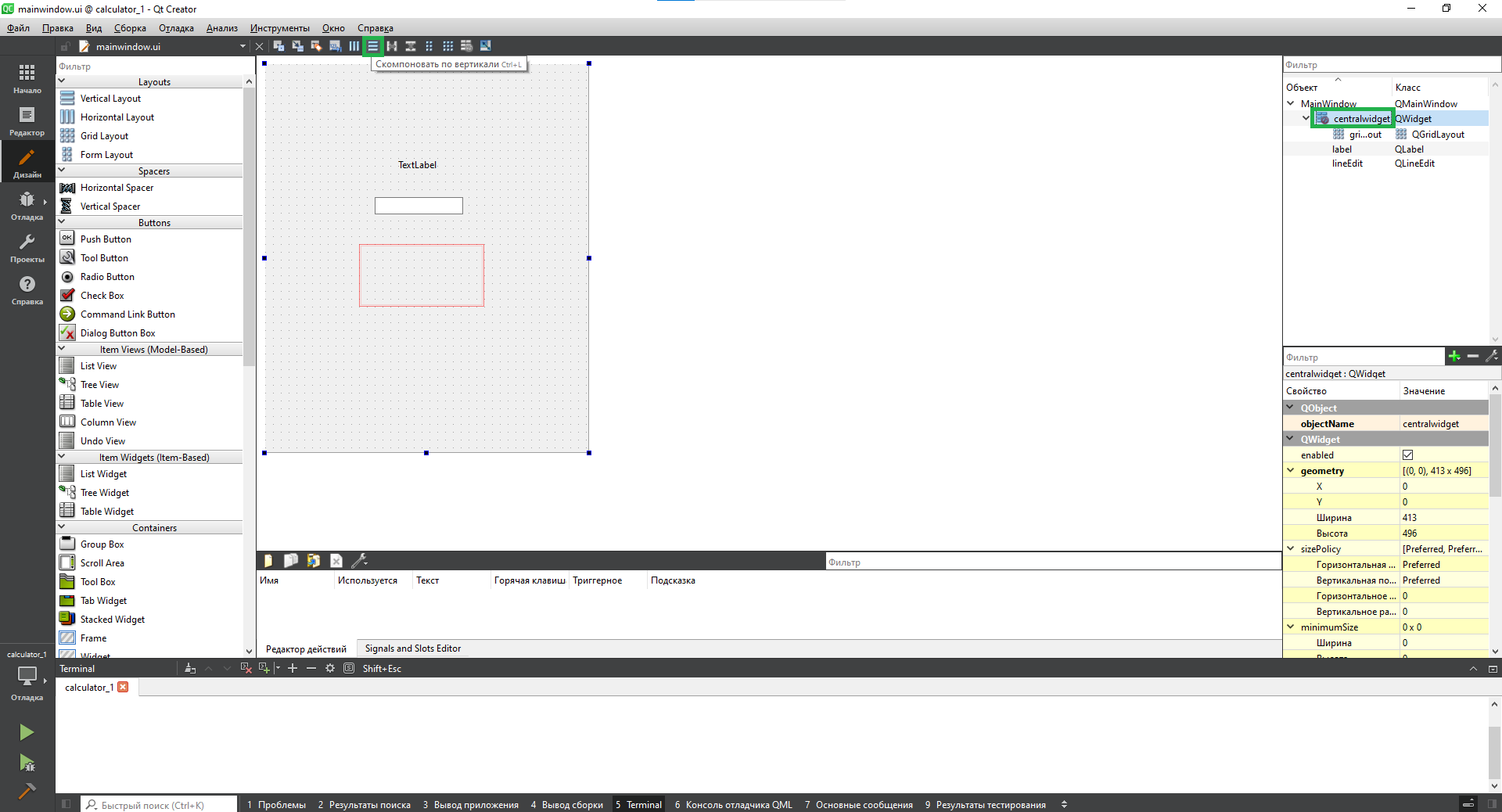


Поменяем в редакторе название программы на “Калькулятор”. Выделим в инспекторе “MainWindow” и поменяем в свойстве “windowTitle” значение на “Калькулятор”. И удалим в инспекторе объекты: menubar, statusbar.

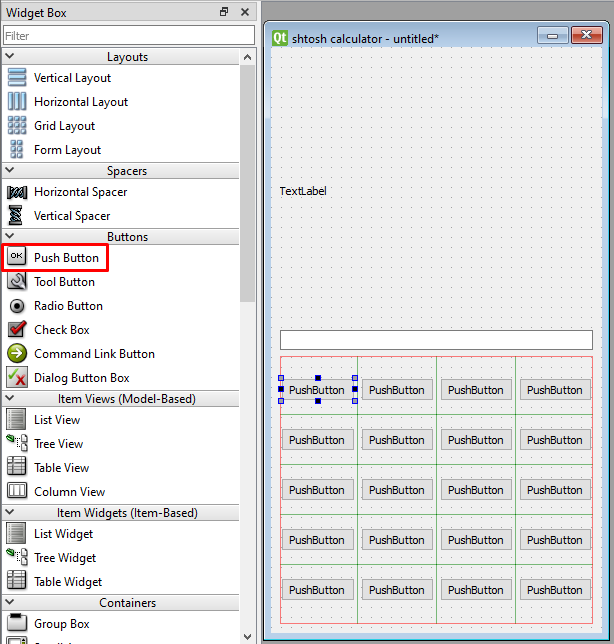


Перетащим в интерфейс компоненты: Line Edit, Label, Grid Layout.

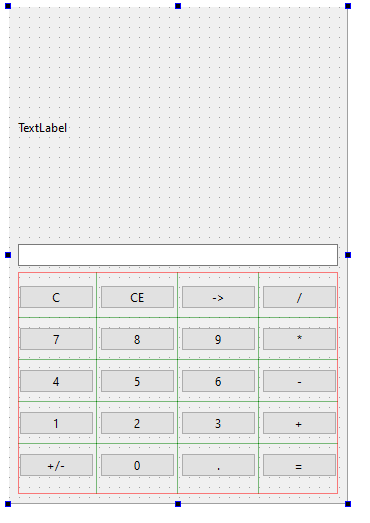


Нажмём на кнопку “Скомпоновать по вертикали”.

Кнопки (Push Button) в Grid Layout в 4 колонки и 5 рядов.



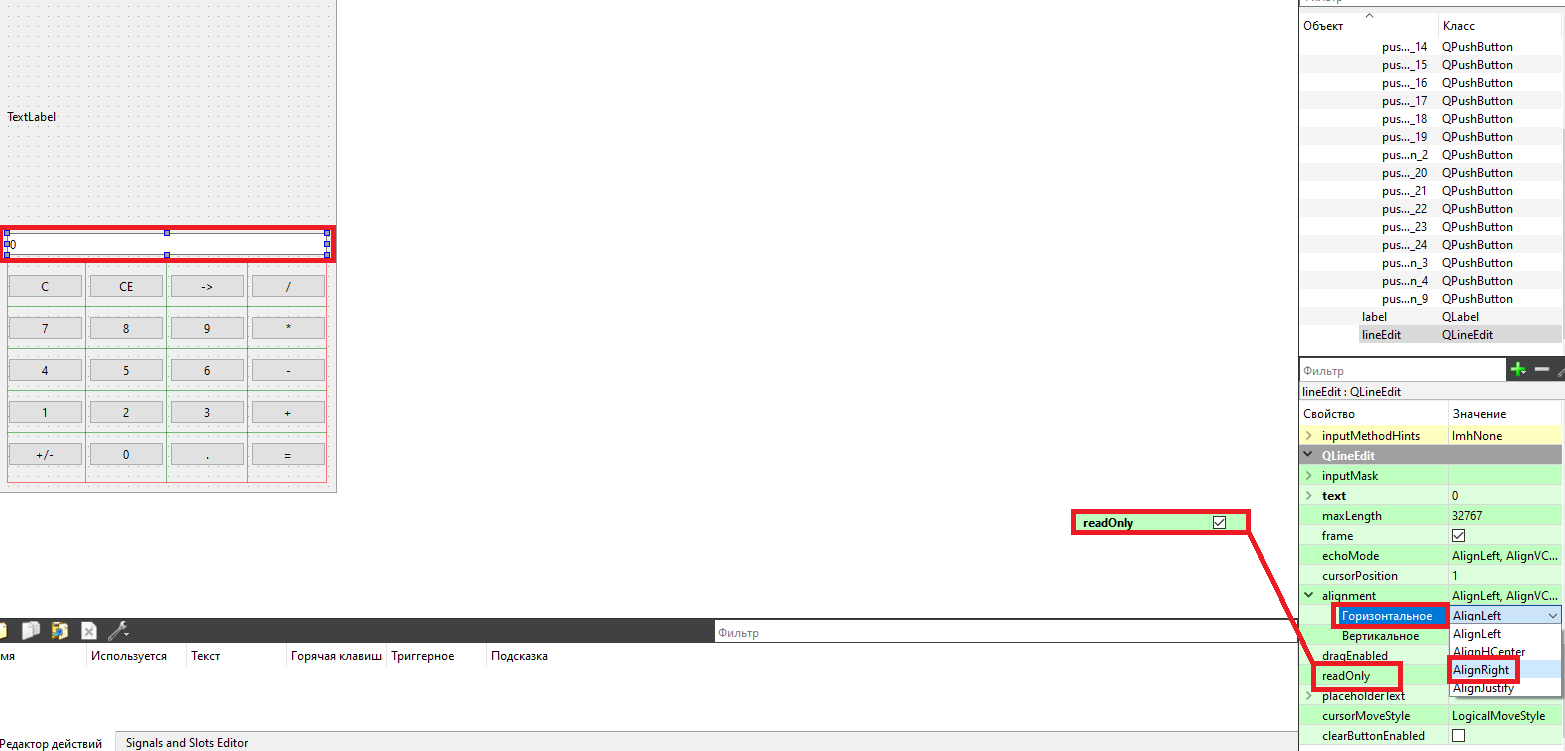
Меняю текст в кнопках.



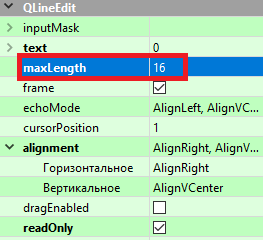
Для каждой кнопки проставим горячие клавиши. В кнопках меняем значения в свойстве “shortcut” (кроме кнопок “+/-“, “C”, “CE”).



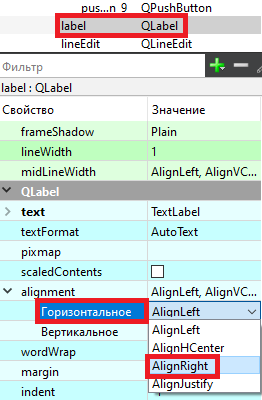
Запишем 0 в “Line Edit” и выберем правое горизонтальное выравнивание для текста. И включим свойство “readOnly”.



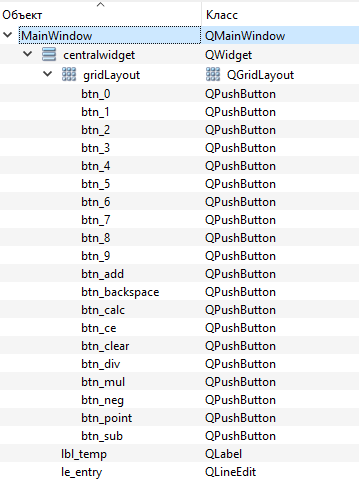
Максимальная длина строки в 16 символов.



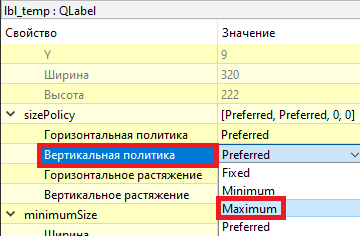
Поставим правое выравнивание для объекта “label”.



Назовём все элементы.



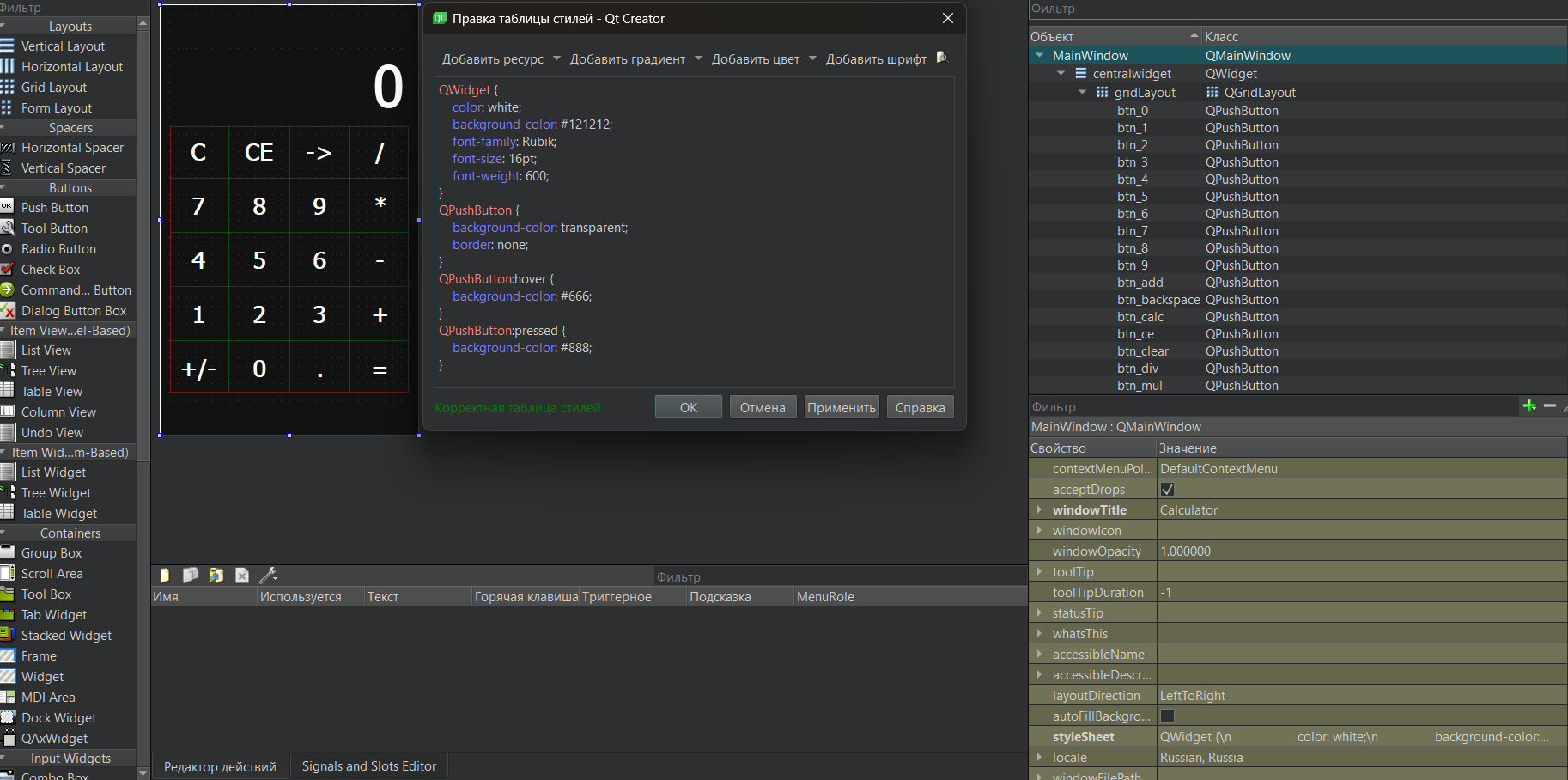
Для “Label” и “Line Edit” поставим во свойстве “Вертикальная политика” поставим значение “Maximum”.



Для всех кнопок во свойстве “ Вертикальная политика” поставим значение “ Expanding”.



Выделим объект “MainWindow” и нажмём на “Изменить styleSheet…”.



И вставим коды:

QWidget {

color: white; //меняет цвет

background-color: #121212; \\Фон

font-family: Rabik; //меняет шрифт

font-size: 16pt; \\ Размер шрифта

font-weight: 600; \\ Убирает границы кнопки

}

QPushButton { \\ Это указывает на то, что стили будут применяться ко всем элементам QPushButton в вашем приложении.

background-color: transparent; \\ Эта строка устанавливает прозрачный цвет фона для кнопки. Кнопка не будет иметь видимого фона и будет наследовать фон от своего родительского элемента или нижележащих элементов.

border: none; \\ Эта строка убирает границу у QPushButton. Кнопка не будет иметь видимой границы, что придает ей более минималистичный вид.

}

QPushButton:hover { \\ Эта часть кода указывает, что следующие стили будут применяться к QPushButton только в том случае, если курсор находится над кнопкой (то есть, когда происходит событие "hover").

background-color: #666; \\ Эта строка устанавливает цвет фона кнопки при наведении курсора. В данном случае, цвет установлен как тёмно-серный (#666), что означает, что при наведении курсора на кнопку, ее фон будет изменяться на тёмно-серный.

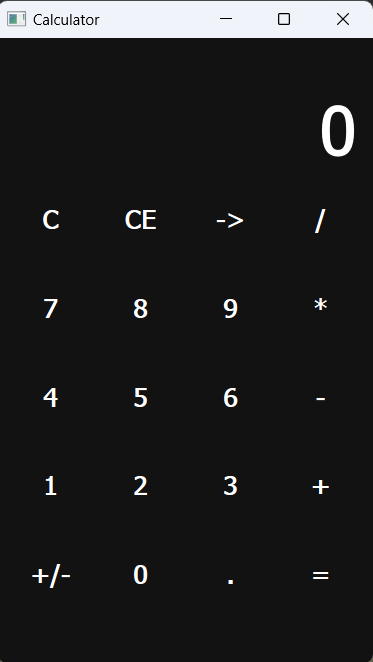
}

QPushButton:pressed { \\Эта часть кода указывает, что следующие стили будут применяться к QPushButton только в том случае, если кнопка находится в состоянии "pressed" (нажата).

background-color: #888; \\ Эта строка устанавливает цвет фона кнопки при ее нажатии. В данном случае, цвет установлен как серый (#888), что означает, что при нажатии на кнопку, ее фон будет изменяться на серый.

}

Результат:

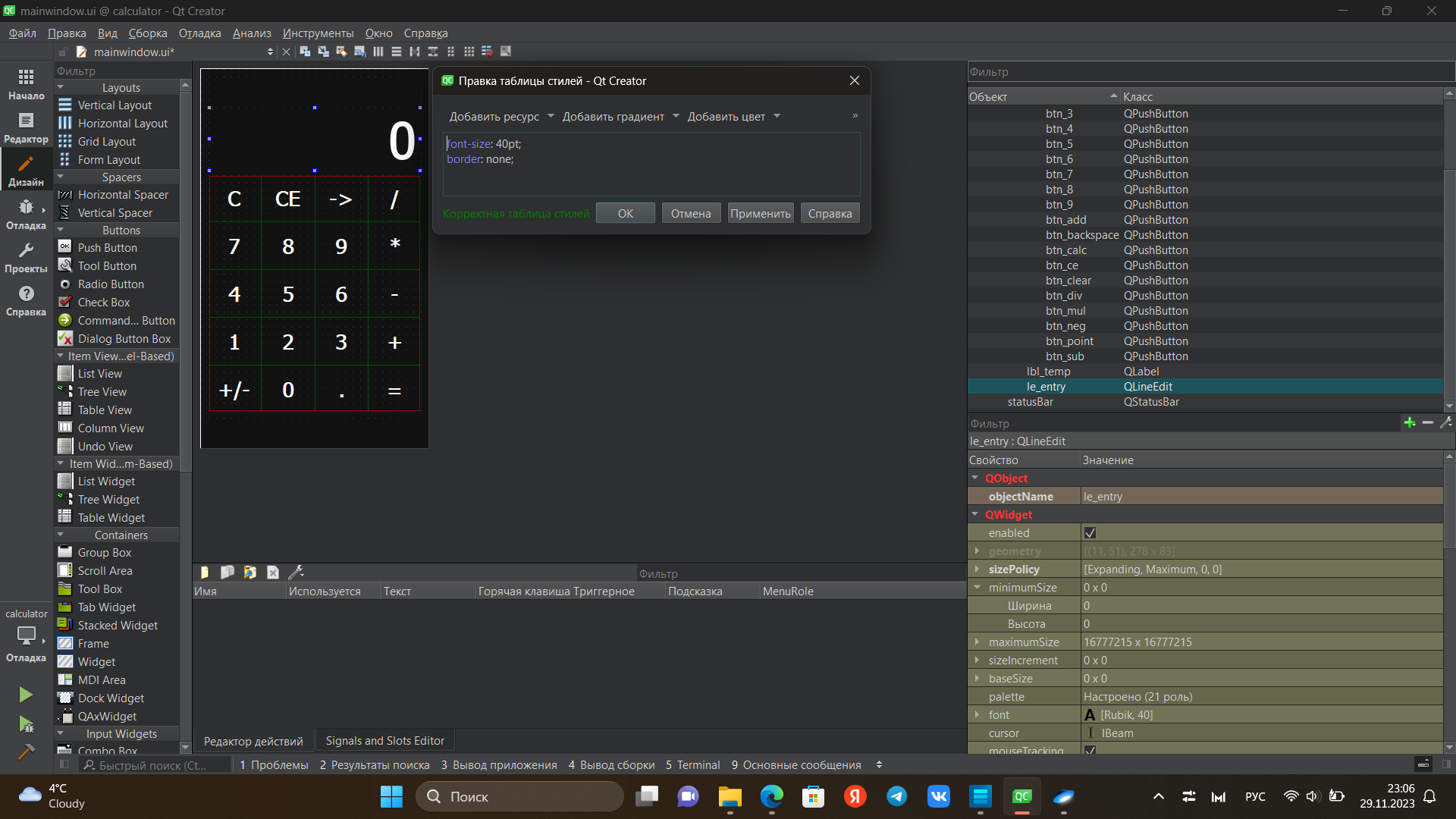
****

Поменяем стиль у Line Edit и Label:

1)Line Edit: \\ Эта строка указывает, что стили будут применяться ко всем элементам QLineEdit в вашем приложении. QLineEdit представляет собой виджет для ввода текста

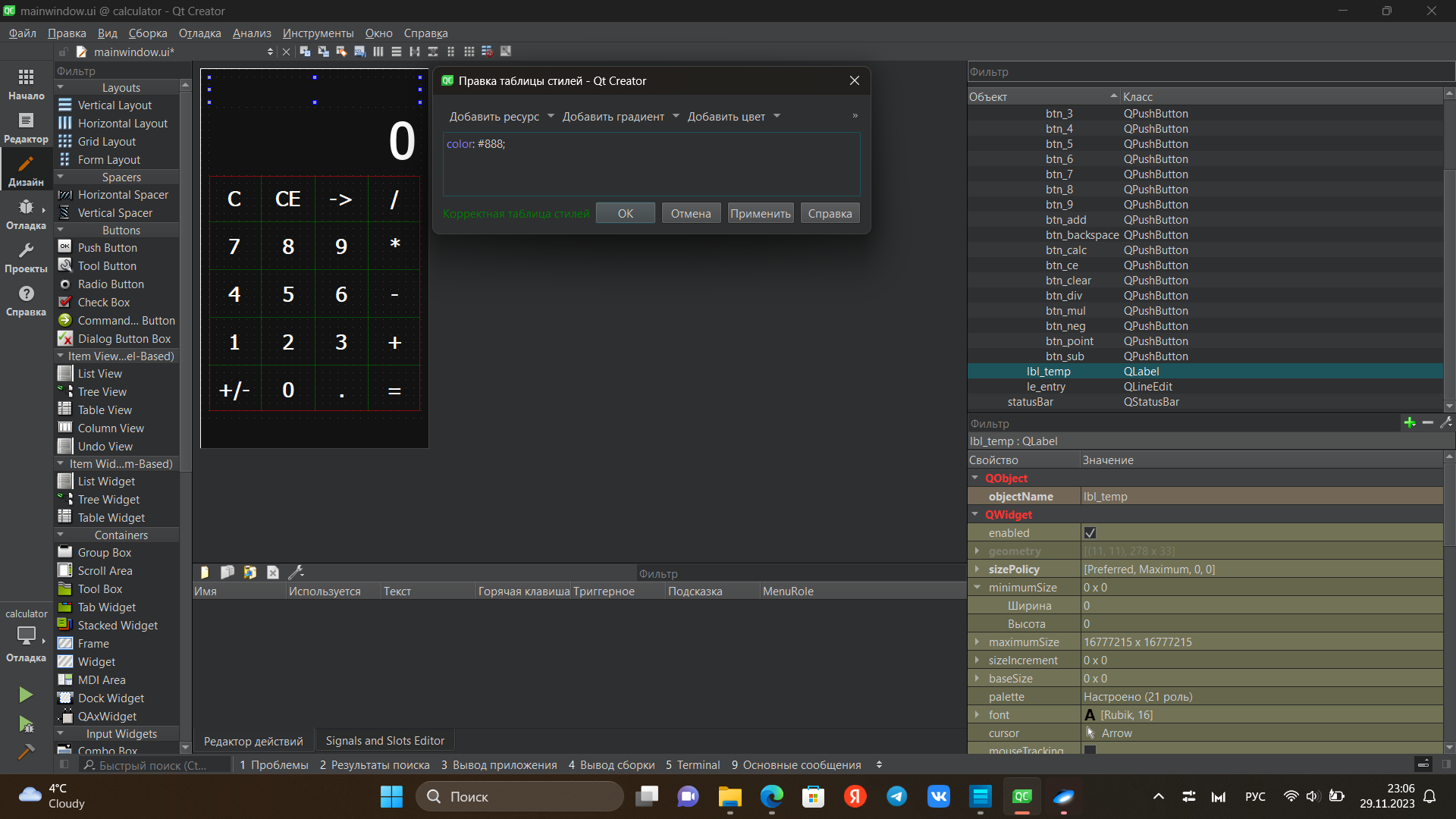
font-size: 40pt; \\ Эта строка устанавливает размер шрифта для текста, вводимого в QLineEdit. В данном случае, размер шрифта установлен как 40pt

border: none; \\ Эта строка убирает границу у QLineEdit. Таким образом, виджет не будет иметь видимой границы, что придает ему более минималистичный вид

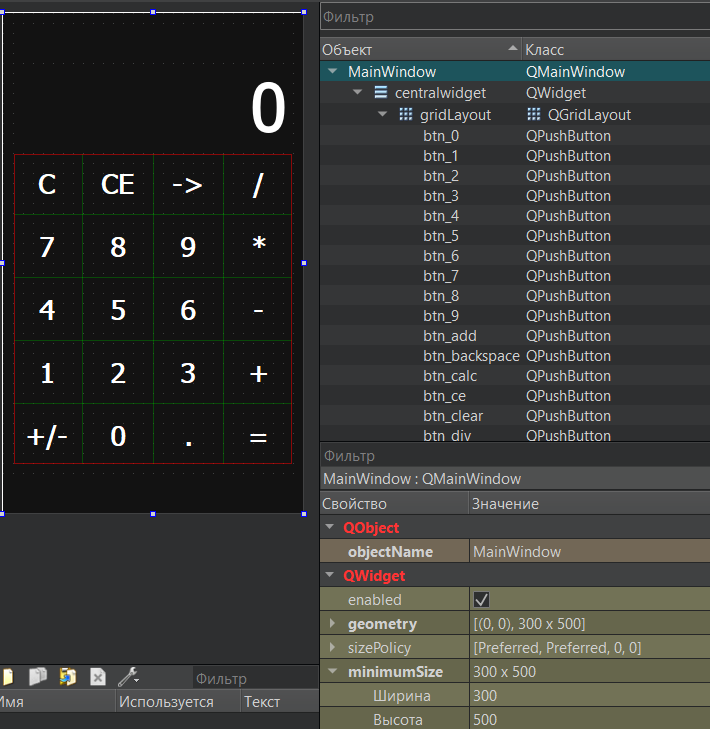


2)Label:

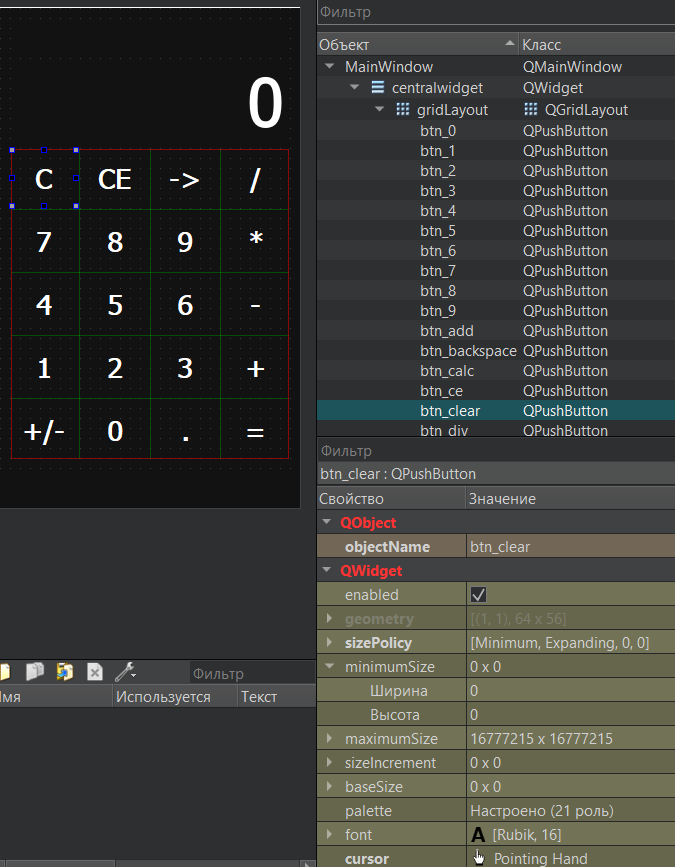
color: #888;



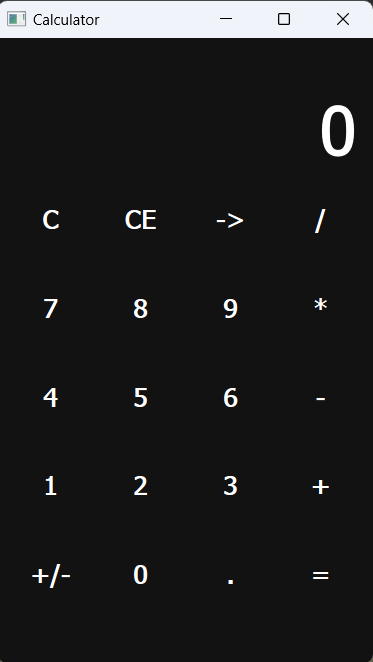
Уберём текст из лейбла. Зададим размер главного окна 350 на 500 пикселей, также поставим для минимального размера окна.



Добавим курсор “указывающая рука” для кнопки “C”. Он будет активироваться при наведении курсора на кнопку.



# Глава 2. Разработка калькулятора.

****

Кнопки «1»/«2»/«3»/«4»/«5»/«6»/«7»/«8»/«9»/«.» добавляют соответственное число в строку.   
Кнопки «=»/«+»/«-»/«\*»/«/»/ - выполняют арифметические действия над числами: вычисление, сложение, вычитание, умножение, деление.  
Кнопка «CE» - полный сброс.  
Кнопка «C» - удаление строки, которая введена после нажатия на знак «=».

Кнопка «+/-» - добавляет минус в начало строки.

Кнопка «->» - стирает символ.

# Глава 3. Листинг “mainwindow.cpp”:

#include "mainwindow.h"

#include "./ui\_mainwindow.h"

//#include <string>

QString otvet = ""; \\ Добавлена глобальная переменна otvet для хранения предыдущего ответа

MainWindow::MainWindow(QWidget \*parent) \\ В конструкторе (MainWindow::MainWindow) инициализируется пользовательский интерфейс

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_btn\_0\_clicked() \\ Каждый метод-обработчик (on\_btn\_...\_clicked()) вызывается при нажатии соответствующей кнопки и обновляет текст в lbl\_temp

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_0 ->text());

}

void MainWindow::on\_btn\_1\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_1 ->text());

}

void MainWindow::on\_btn\_2\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_2 ->text());

}

void MainWindow::on\_btn\_3\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_3 ->text());

}

void MainWindow::on\_btn\_4\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_4 ->text());

}

void MainWindow::on\_btn\_5\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_5 ->text());

}

void MainWindow::on\_btn\_6\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_6 ->text());

}

void MainWindow::on\_btn\_7\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_7 ->text());

}

void MainWindow::on\_btn\_8\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_8 ->text());

}

void MainWindow::on\_btn\_9\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_9 ->text());

}

void MainWindow::on\_btn\_add\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + " " + ui->btn\_add ->text() + " ");

}

void MainWindow::on\_btn\_backspace\_clicked()

{

QString text\_entry = ui->le\_entry->text();

QString text\_temp = ui->lbl\_temp->text();

text\_entry.chop(1);

text\_temp.chop(1);

ui->le\_entry->setText(text\_entry);

ui->lbl\_temp->setText(text\_temp);

}

void MainWindow::on\_btn\_calc\_clicked()

{

QString expression = ui->lbl\_temp->text();

QString result = calculateExpression(expression); \\ Этот метод принимает математическое выражение в виде строки, разбивает его на числа и операторы, выполняет умножение и деление, а затем сложение и вычитание, чтобы вернуть результат

ui->le\_entry->setText(result);

otvet = expression; \\Вызывается при нажатии кнопки "=" и использует calculateExpression для вычисления результата и отображения его в le\_entry

}

void MainWindow::on\_btn\_ce\_clicked()

{

ui->le\_entry->clear();

ui->lbl\_temp->clear();

}

void MainWindow::on\_btn\_clear\_clicked()

{

ui->lbl\_temp->clear();

ui->lbl\_temp->setText(ui->lbl\_temp->text() + otvet);

}

void MainWindow::on\_btn\_div\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + " " + ui->btn\_div ->text() + " ");

}

void MainWindow::on\_btn\_mul\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + " " + ui->btn\_mul ->text() + " ");

}

void MainWindow::on\_btn\_neg\_clicked()

{

QString text = ui->lbl\_temp->text();

if (text.isEmpty()) {

return;

}

if (text.at(0) == '-') {

text.remove(0, 1);

}

else {

text.prepend("-");

}

ui->lbl\_temp->setText(text);

}

void MainWindow::on\_btn\_point\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + ui->btn\_point ->text());

}

void MainWindow::on\_btn\_sub\_clicked()

{

ui->lbl\_temp->setText(ui->lbl\_temp->text() + " " + ui->btn\_sub ->text() + " ");

}

QString MainWindow::calculateExpression(const QString &expression) {

qDebug() << "Expression:" << expression;

QStringList tokens = expression.split(" ");

qDebug() << "Tokens:" << tokens;

QList <QString> numbers;

QList <QString> operators;

for (const QString &token: tokens) {

if (token != "+" && token != "-" && token != "\*" && token != "/") {

numbers.push\_back(token);

}

else {

operators.push\_back(token);

}

}

for (int i = 0; i < operators.size(); i++) {

if (operators.at(i) == "\*" || operators.at(i) == "/") {

double a = numbers.at(i).toDouble();

QString op = operators.at(i);

double b = numbers.at(i + 1).toDouble();

if (op == "\*") {

numbers[i] = QString::number(a \* b);

}

else if (op == "/") {

if (b == 0.0) {

return "Error: Division by zero";

}

numbers[i] = QString::number(a / b);

}

operators.removeAt(i);

numbers.removeAt(i + 1);

--i;

}

}

while (!operators.isEmpty()) {

double a = numbers.takeFirst().toDouble();

QString op = operators.takeFirst();

double b = numbers.takeFirst().toDouble();

if (op == "+") {

numbers.push\_front(QString::number(a + b));

}

else if (op == "-") {

numbers.push\_front(QString::number(a - b));

}

}

if (numbers.size() == 1) {

QString result = numbers.takeFirst();

qDebug() << "Result:" << result;

return result;

}

else {

return "Error";

}

}

# Глава 4. Листинг “mainwindow.h”:

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QLabel>

#include <QGridLayout>

#include <QTextEdit>

QT\_BEGIN\_NAMESPACE

namespace Ui {

class MainWindow;

}

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public: // Конструктор и деструктор

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private slots: // Объявления слотов для обработки событий

void on\_btn\_0\_clicked();

void on\_btn\_1\_clicked();

void on\_btn\_2\_clicked();

void on\_btn\_3\_clicked();

void on\_btn\_4\_clicked();

void on\_btn\_5\_clicked();

void on\_btn\_6\_clicked();

void on\_btn\_7\_clicked();

void on\_btn\_8\_clicked();

void on\_btn\_9\_clicked();

void on\_btn\_add\_clicked();

void on\_btn\_backspace\_clicked();

void on\_btn\_calc\_clicked();

void on\_btn\_ce\_clicked();

void on\_btn\_clear\_clicked();

void on\_btn\_div\_clicked();

void on\_btn\_mul\_clicked();

void on\_btn\_neg\_clicked();

void on\_btn\_point\_clicked();

void on\_btn\_sub\_clicked();

QString calculateExpression(const QString &expression);

private: // Указатель на объект интерфейса, указатели на другие виджеты

Ui::MainWindow \*ui;

QLabel \*lbl\_temp;

QGridLayout \*gridLayout\_button;

QTextEdit \*le\_entry;

};

#endif // MAINWINDOW\_H

# Тестирование.

|  |  |  |  |
| --- | --- | --- | --- |
| **Описание теста** | **Вводные данные** | **Выходные данные** | **Результат** |
| Сложение двух чисел | 5 + 2 = | 7 | Получили число 7, равное сумме 5 и 2 |
| Убрать лишнее действие после числа 7 | 5 + 2 + 5, кнопка C | 7 | + 5 убралось из строки ввода |
| Убрать + 2 | 5 + 2, кнопка -> | - | В строке ввода осталось 5 + |
| Стереть всё | 5 +, кнопка СЕ | - | Убрались все символы в строке ввода и вывода |
| Произведение | 0 \* 5 = | 0 | 0 |
| Деление на 0 | 5 / 0 | Error:Division | На ноль делить нельзя, поэтому выдает ошибку |
| Поменять знак | 2.5 \* 2, +/-, = | -5 | Вставили в начале минус и получили отрицательное число |
| Вычитание | 0 - 6 = | -6 | -6 |
| Сложение с дробным | .3 + 3 = | 3.3 | 3.3 |
| Описка | ..3 + 3 | 3 | Не считывает первое число |
| Деление | 42 / 25 = | 7.44 | Дробное число |
| Умножение | 89 \* 1000000000000000000 | 8.9e+19 | Большое число |
| Описка №2 | 2 \*\* 2 | 0 | После умножения ожидается число, а там был знак умножения |
| Сложение дробных | 0.0 + 3.25 | 3.25 | Из разных разрядов сложил и получил с большим разрядом |
| Сложение дробных ещё раз | 3.261 + 4.3790 | 8 | Из сложения типа float получает int |