

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 C++»

Выполнил:

Студент группы: 2023-ФГиИБ-ПИ-16

Корязов Дмитрий Ильич

Проверил(а):

Лебедев Евгений Денисович

Москва 2023

Оглавление

1.Оглавление.	Стр. 2
2.Глава 1.	Стр. 3-12
а.Пункт 1.	Стр. 3-5
б.Пункт 2.	Стр. 6-12
3.Глава 2.	Стр. 13-15

Задание:

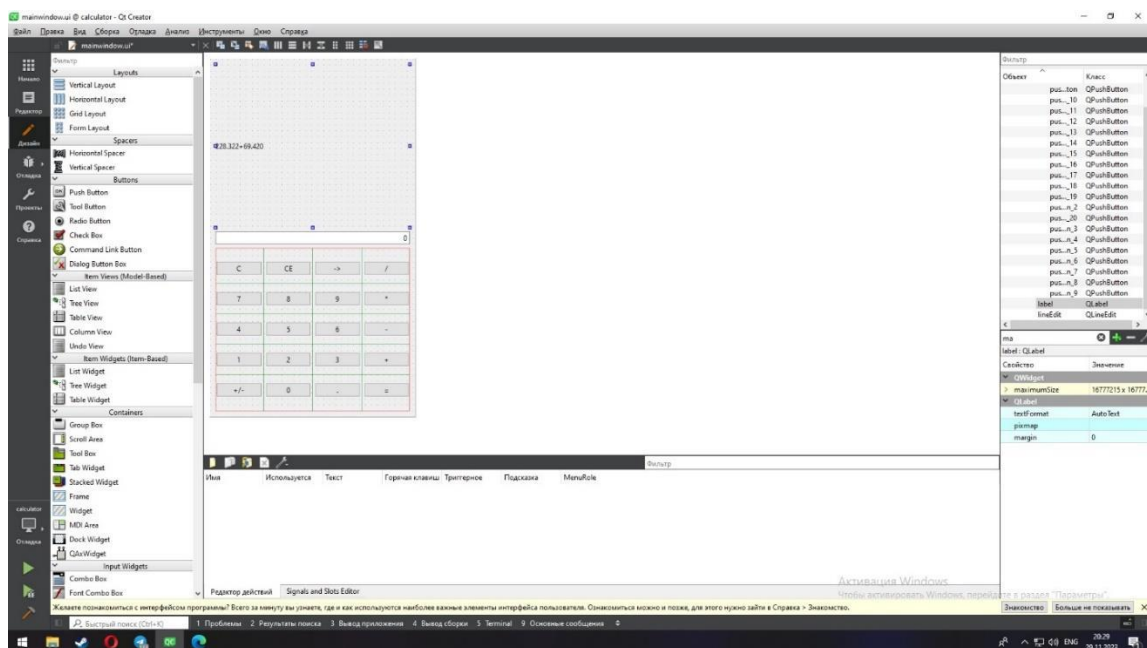
Разработка приложения «калькулятор», который будет выполнять основные арифметические операции (сложение, вычитание, умножение, деление). Калькулятор должен обладать простым и понятным пользовательским интерфейсом и обеспечивать корректное выполнение операций.

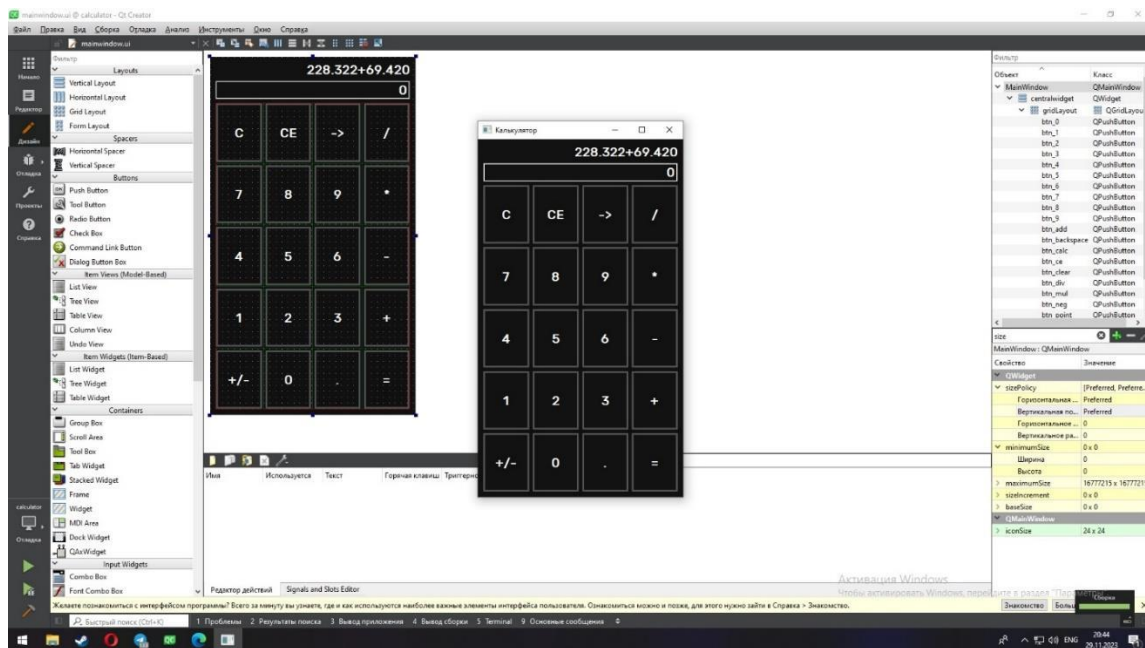
Глава1.

Разработка приложения «калькулятор»

Пункт 1.

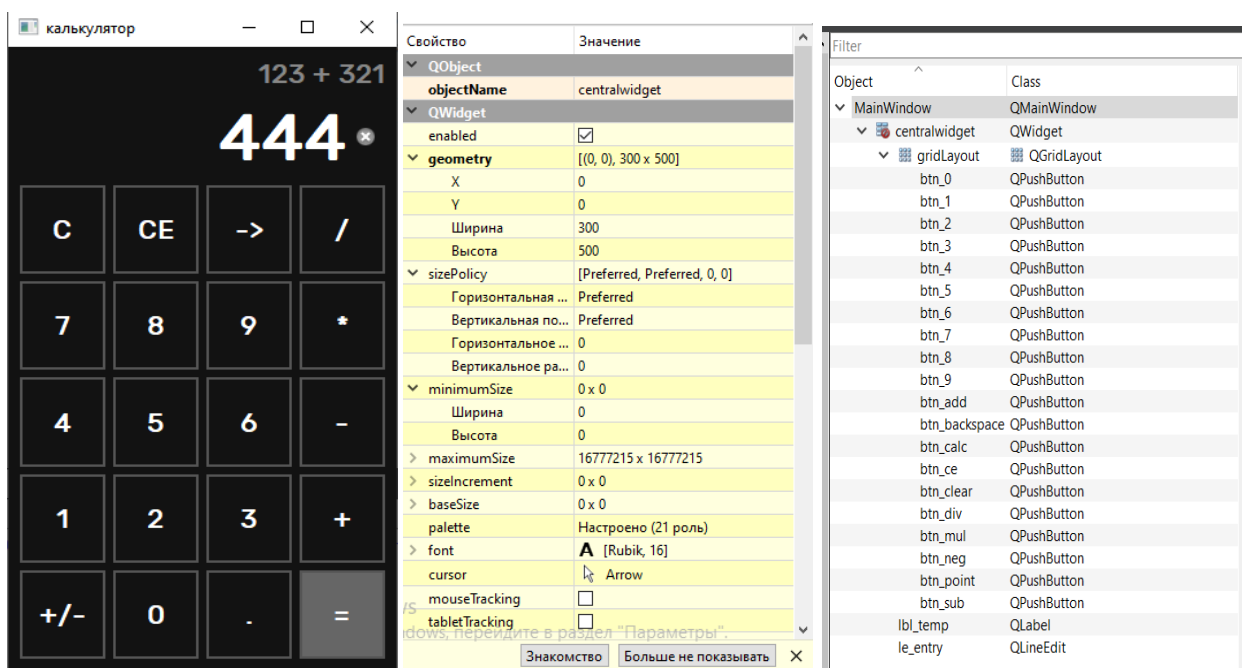
Разработка дизайна





(рис 1, 2)

Шаги работы и проектирования:



(рис. 3,4,5)

Интерфейс и виджеты калькулятора:

-для кнопок мы использовали:

- ```
QWidget {
 •color: white;
 •background-color: #121212;
 •font-family: Rubik;
 •font-size: 16pt;
 •font-weight: 600;
```

```
}
QPushButton {
•background-color: transparent;
•border:2px solid #555;
}
QPushButton:hover {
•background-color: #666;
}
QPushButton:pressed {
•background-color: #888;
}
```

## Пункт 2.

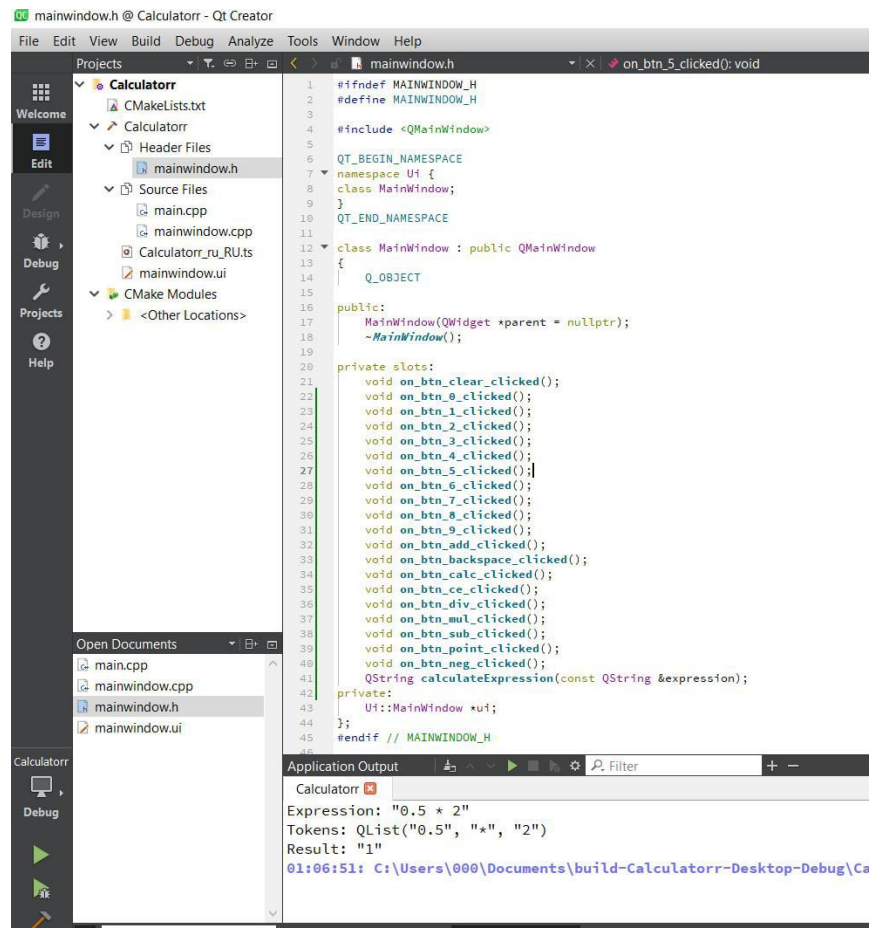
### Разработка логики событий

Помимо непосредственно кнопок, упомянутых выше, или “PushButtons”, я использовал Label из Display Widgets и Line Edit из Input Widgets. Все это было нужно для ввода и вывода данных.



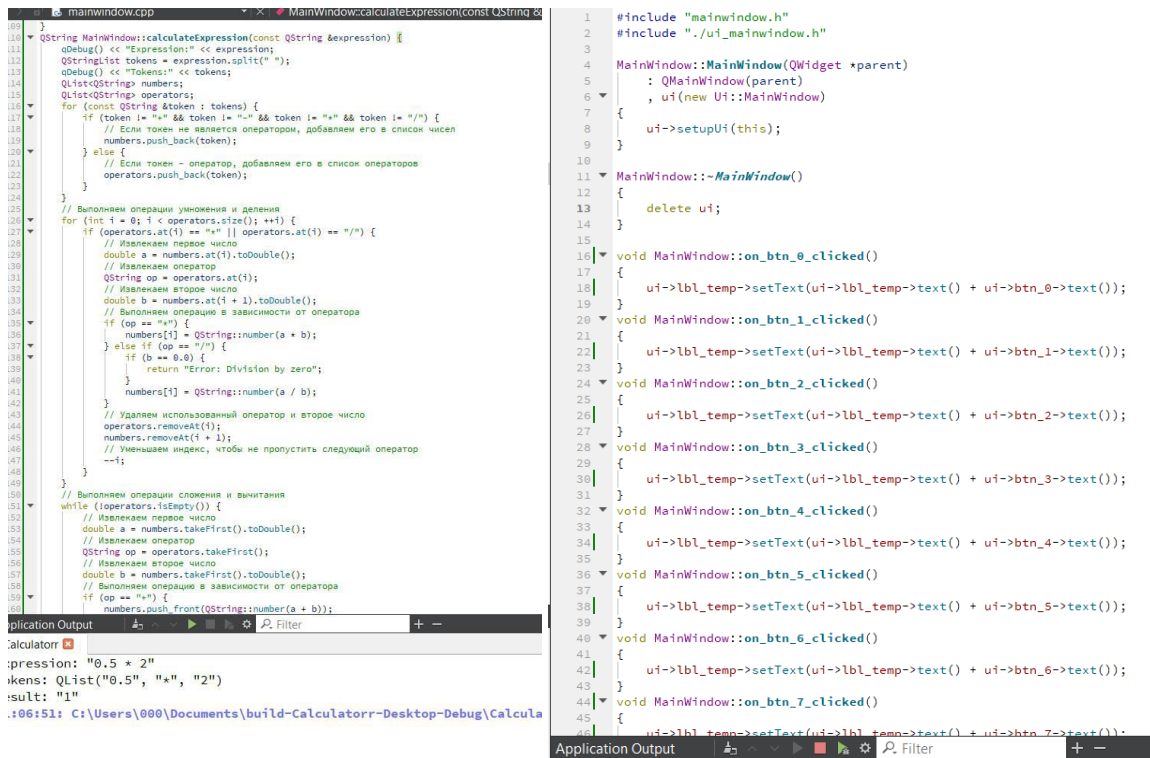
(рис. 6)

Далее к каждой кнопке я обратился с функцией clicked(). –Это нужно для того чтобы клавиши были “кликабельными”, проще говоря нажимались.



(рис. 7)

Обращаясь к каждой кнопке пишем код, для точки и = код будет уникальным и не похожим на другие (как например коды для цифр 0-9 ).



(рис. 8, 9)

Далее следует Листинг получившегося кода, разделенный над .h и .cpp

## Листинг кода:

### 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 slots:
 void on_btn_clear_clicked();
 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_div_clicked();

void on_btn_mul_clicked();

void on_btn_sub_clicked();

void on_btn_point_clicked();

void on_btn_neg_clicked();

QString calculateExpression(const QString &expression);

private:

 Ui::MainWindow *ui;

};

#endif // MAINWINDOW_H

```

## Mainwindow.cpp

```

#include "mainwindow.h"

#include "../ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
 : QMainWindow(parent)
 , ui(new Ui::MainWindow)
{
 ui->setupUi(this);
}

MainWindow::~MainWindow()
{
 delete ui;
}

void MainWindow::on_btn_0_clicked()
{
 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_ce_clicked()
{
 ui->le_entry->clear();
 ui->lbl_temp->clear();
}

void MainWindow::on_btn_clear_clicked()
{
}

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->le_entry->text();

 if (text.isEmpty()) {
 return;
 }

 if (text.at(0) == '-') {
 text.remove(0, 1);
 } else {
 text.prepend("-");
 }

 ui->le_entry->setText(text);
}

void MainWindow::on_btn_point_clicked()

```

```

{
 ui->lbl_temp->setText(ui->lbl_temp->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 != "/") {
 // Если токен не является оператором, добавляем его в список чисел
 numbers.push_back(token);
 } else {
 // Если токен - оператор, добавляем его в список операторов
 operators.push_back(token);
 }
 }

 // Выполняем операции умножения и деления
 for (int i = 0; i < operators.size(); ++i) {
 if (operators.at(i) == "*" || operators.at(i) == "/") {
 // Извлекаем первое число
 double a = numbers.at(i).toDouble();

 // Извлекаем оператор
 QString op = operators.at(i);

 // Извлекаем второе число
 double b = numbers.at(i + 1).toDouble();

 // Выполняем операцию в зависимости от оператора
 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);
 }
 }
 }
}

```

```

 }

 // Удаляем использованный оператор и второе число
 operators.removeAt(i);
 numbers.removeAt(i + 1);

 // Уменьшаем индекс, чтобы не пропустить следующий оператор --
i;
 }
}

// Выполняем операции сложения и вычитания
while (!operators.isEmpty()) {
 // Извлекаем первое число
 double a = numbers.takeFirst().toDouble();

 // Извлекаем оператор
 QString op = operators.takeFirst();

 // Извлекаем второе число
 double b = numbers.takeFirst().toDouble();

 // Выполняем операцию в зависимости от оператора
 if (op == "+") {
 numbers.push_front(QString::number(a + b));
 } else if (op == "-") {
 numbers.push_front(QString::number(a - b));
 }
}

// В списке должен остаться один элемент - результат выражения if
(numbers.size() == 1) {
 QString result = numbers.takeFirst();

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

 return result;
} else {
 return "Error";
}
}

void MainWindow::on_btn_calc_clicked()
{
 QString expression = ui->lbl_temp->text();

 QString result = calculateExpression(expression);

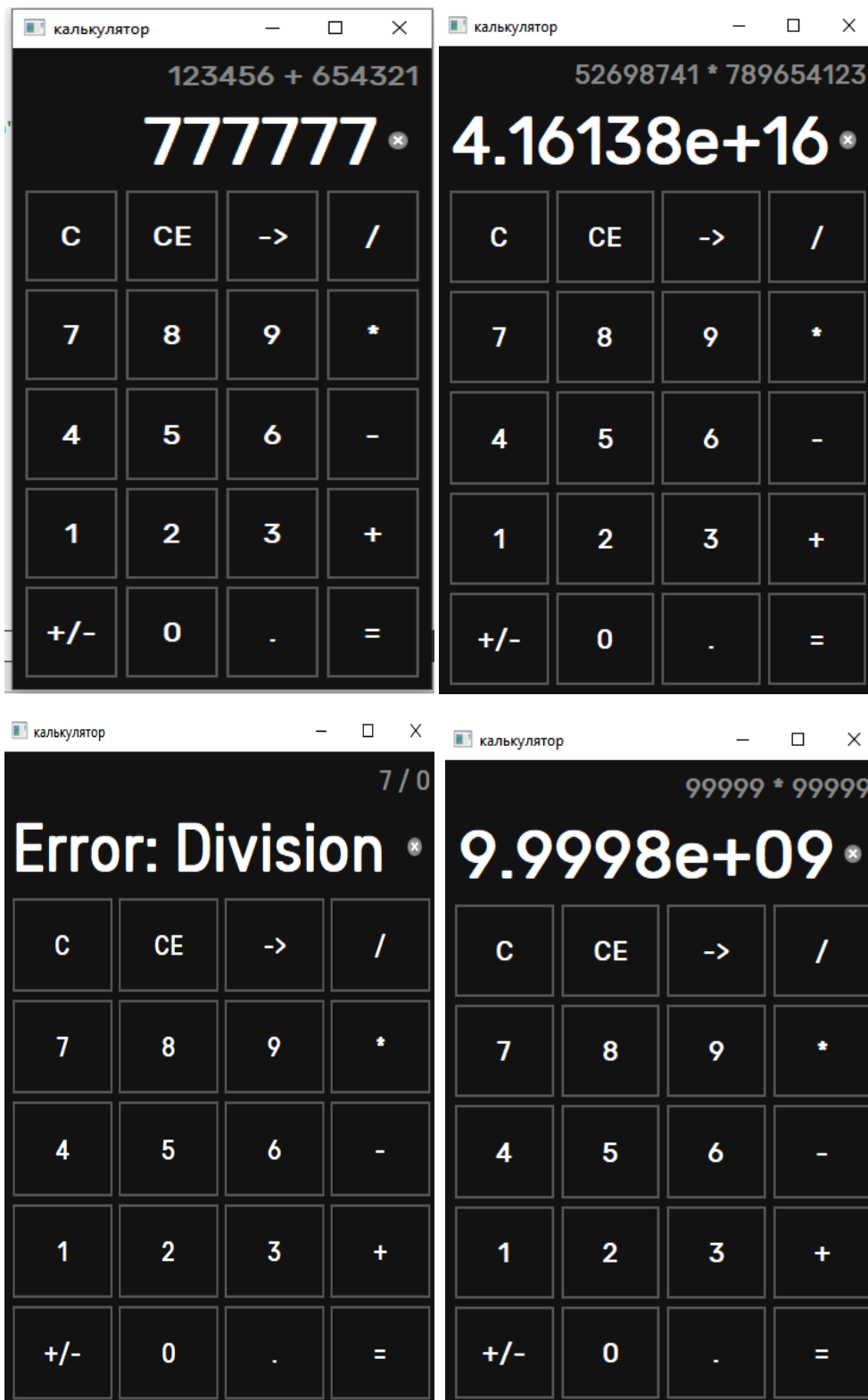
 ui->le_entry->setText(result);
}

```

## Глава 2

### Тестирование Программы

Последним шагом идет тестирование и подготовка результатов/вывод:



(рис. 10, 11, 12, 13)

«Кнопка «1» добавляет 1 в строку."

«Кнопка «2» добавляет 2 в строку."

«Кнопка «3» добавляет 3 в строку."

«Кнопка «4» добавляет 4 в строку."

«Кнопка «5» добавляет 5 в строку."

«Кнопка «6» добавляет 6 в строку."

«Кнопка «7» добавляет 7 в строку."

«Кнопка «8» добавляет 8 в строку."

«Кнопка «9» добавляет 9 в строку."

«Кнопка «0» добавляет 0 в строку."

«Кнопка «/» добавляет / в строку."

«Кнопка «\*» добавляет \* в строку."

«Кнопка «-» добавляет - в строку."

«Кнопка «+» добавляет + в строку."

«Кнопка «C» очищает строку."

«Кнопка «->» стирает последний символ строки."

«Кнопка «.» добавляет . в строку."

«Кнопка «+/-» изменяет знак в строке ответа на противоположный."

«Кнопка «=» делит строку на подстроки и выполняет действия, которые находятся между числами, после чего выводит результат."

Все процессы работают, программа не вылетает, учтены ситуации которые могут приостановить работу(такие как деление на ноль, или работа с большими числами)

Приложение умеет работать с дробными числами и выводить в ответ нецелые числа.

Тестирование:

| Описание теста          | Вводные данные       | Выходные данные              | Результат                                                  |
|-------------------------|----------------------|------------------------------|------------------------------------------------------------|
| Сложение                | 7+8                  | 15                           | Кнопка = дала сумму 7+8 равное 15                          |
| Вычитание               | 298 - 100            | 198                          | Кнопка = дала разность 298 - 100 равное 198                |
| Умножение               | 12*10                | 120                          | Кнопка = дала произведение 12*10 равное 120                |
| Деление                 | 35/5                 | 7                            | Кнопка= дала 35/5 равное 7                                 |
| Сложение больших чисел  | 12974912 + 128471290 | 141446202                    | Программа работает исправно с целыми числами при нажатии = |
| Вычитание больших чисел | 1284012234-9123      | 1284003111                   | Программа работает исправно с целыми числами при нажатии = |
| Умножение больших чисел | 12129481*138301      | 1,6775194 × 10 <sup>12</sup> | Программа работает исправно с целыми числами при нажатии = |
| Деление больших чисел   | 9820237032/2         | 4910118516                   | Программа работает исправно с целыми числами при нажатии = |
| Работа                  | с                    | дробными                     | числами                                                    |
| Сложение                | 0.5+0.1              | 0.6                          | Исправно                                                   |
| Вычитание               | 4.2 – 0.8            | 3.4                          | Исправно                                                   |
| Умножение               | 1.2*45.9             | 55,08                        | Исправно                                                   |
| Деление                 | 8193.51 / 289.3      | 28,32184583                  | Исправно                                                   |
| Деление на ноль         | 7/0                  | Error:Division               | На ноль делить нельзя, поэтому программа работает исправно |
| Умножение на ноль       | 312423*0             | 0                            | Исправно                                                   |

Вывод:

Программа работает исправно во всех случаях, т.к. учтены все возможные ситуации.