Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана»

(национальный исследовательский университет)

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА _____КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)_____

Отчет

по лабораторной работе № 1

Название лабораторной работы: Создание графических приложений в среде Qt Creator.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-24Б **02.03.2024** **А.С. Воеводин**

(Подпись, дата) (И.О. Фамилия)

Преподаватель **02.03.2024** **О.А. Веселовская**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Ознакомиться с механизмом работы с Qt и QtDesigner в языке C++

Задание

Заменить в программе схему выравнивания QHBoxLayout на QVBoxLayout и зафиксировать результат.

Изменить тип разделителя с QSplitter(Qt::Horizontal); на QSplitter(Qt::Vertical); и зафиксировать полученный результат.

Добавьте кнопки, выполняющие: бинарные операции x^y, $log_y x$ (по аналогии с операциями +,-,/,*), а также унарные sin(x) и cos(x) (по аналогии с операцией -/+) и разместите этот ряд кнопок вертикально, слева от цифровых кнопок с использованием нового объекта выравнивания
(Layout).

Разработать приложение, имеющее строку ввода данных, кнопку запуска преобразования и текстовое поле, предназначенное только для отображения информации. При
этом не использовать QtDesigner! Любой текст строки ввода должен отображаться в текстовом поле сразу после завершения ввода. В начале строки должна быть вставлена пометка «input:». При нажатии кнопки преобразования строка ввода должна быть преобразована либо в верхний регистр, либо в нижний противоположно тому, что производилось
при предыдущем нажатии кнопки.

Ход работы:

- Выполнение первого задания
- Выполнение второго задания
- Выполнение третьего задания и проведение тестов
- Выполнение четвертого задания и проведение тестов
- Вывод.

Для начала выполним первое задание и зафиксируем результат:

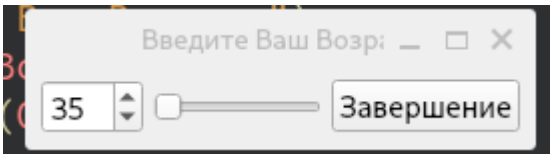


Рисунок 1 – Окно до изменения

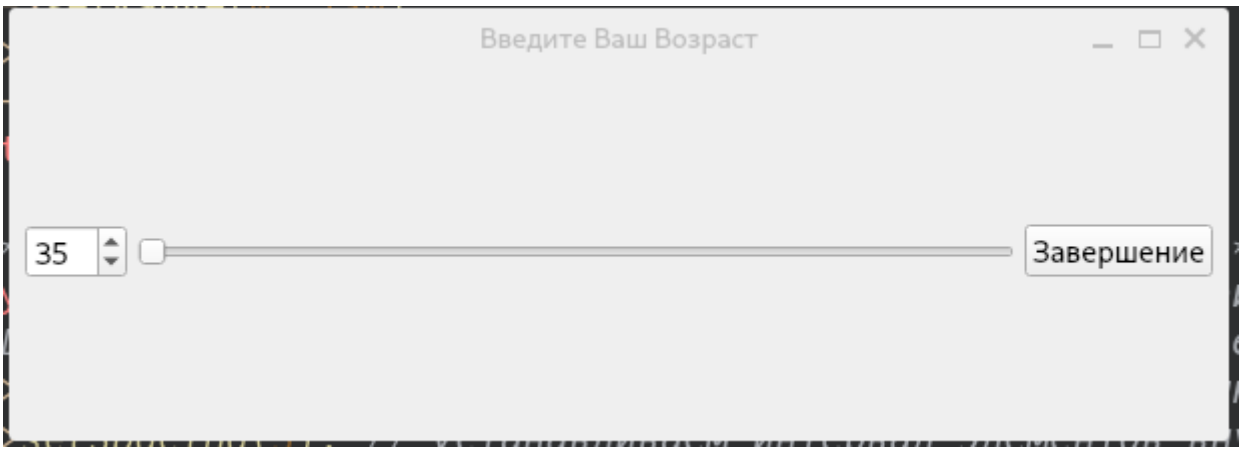


Рисунок 2 – Окно до изменения

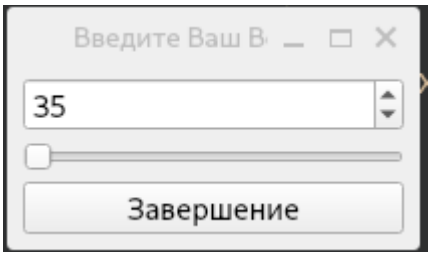


Рисунок 3 – Окно после изменения

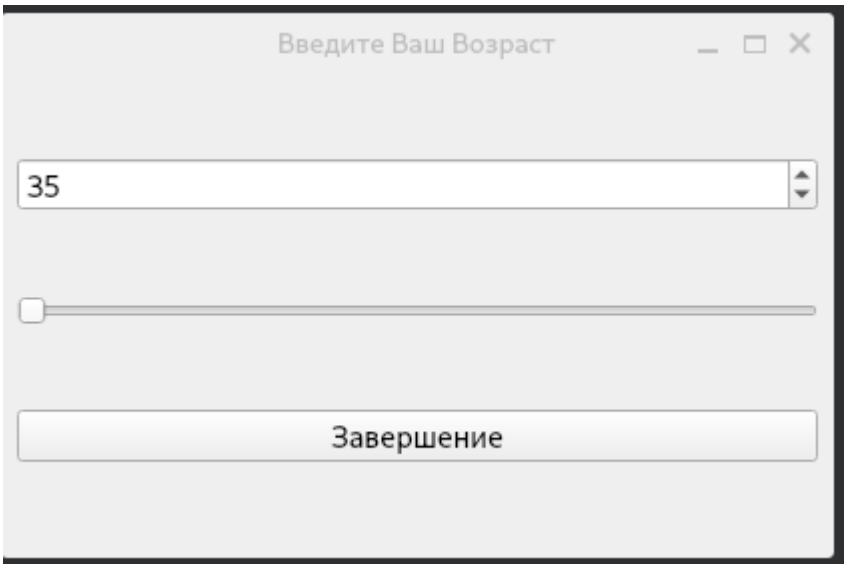


Рисунок 4 – Окно после изменения

Как видно из рисунков, выравнивание теперь не по горизонтали, а по вертикали.

Изменённая часть программы: QVBoxLayout *layout = new QVBoxLayout;

Теперь приступим к второму заданию:

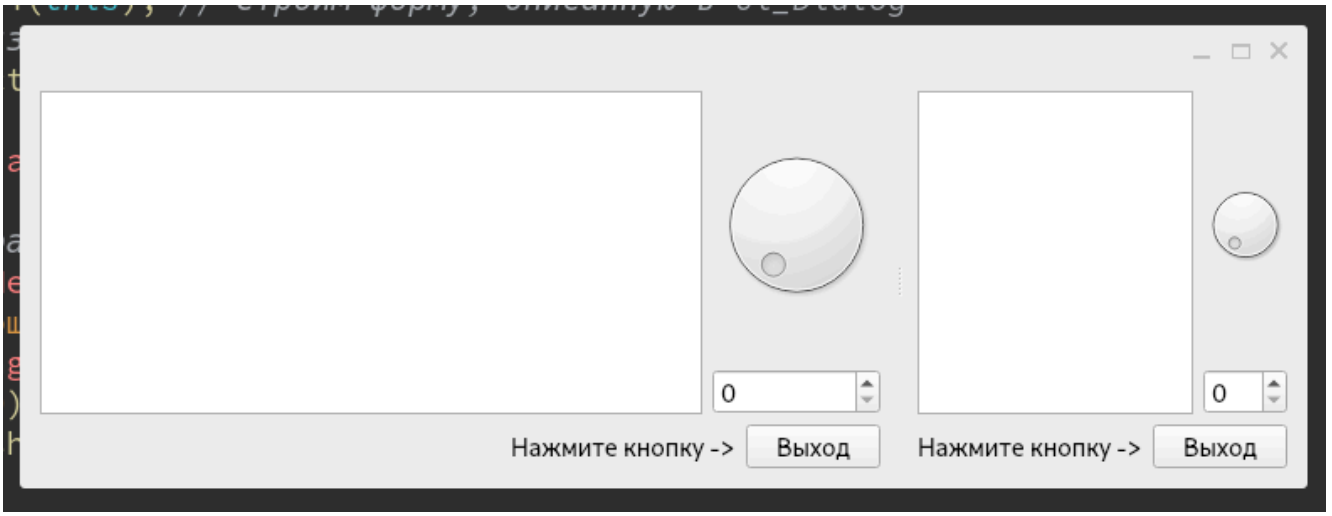


Рисунок 5 – Окно до изменения

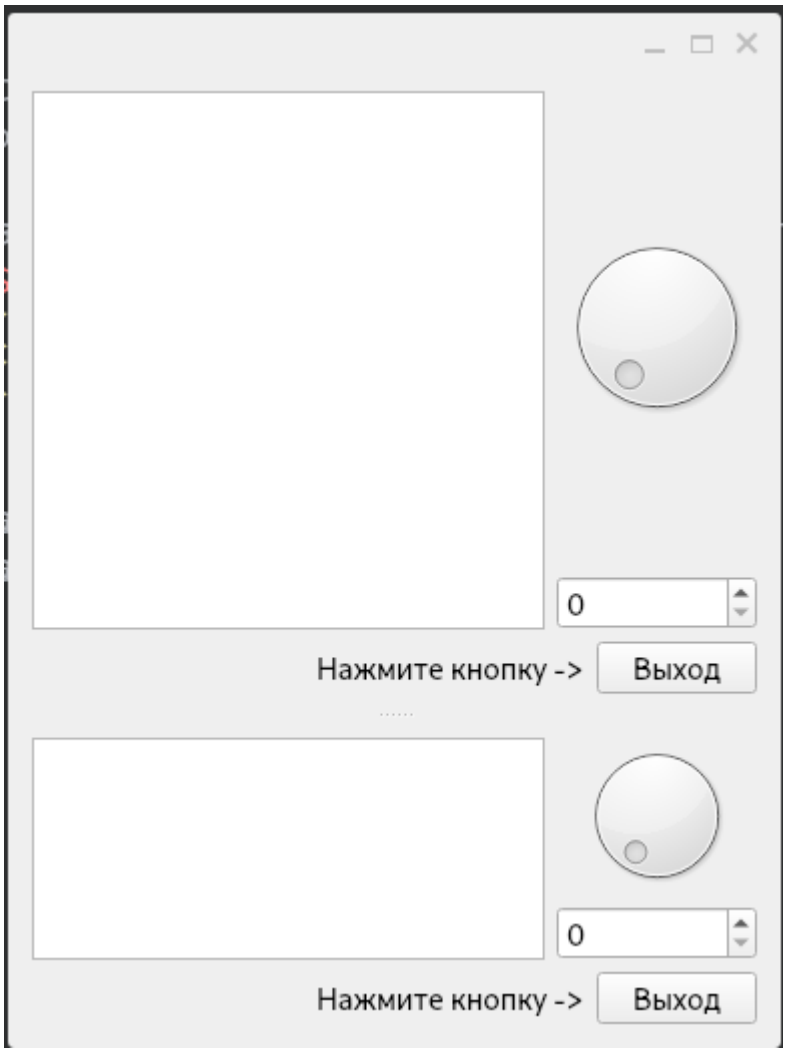


Рисунок 6 – Окно после изменения

Изменённая часть программы: QSplitter * splitter = new QSplitter(Qt::Vertical);

Теперь изменим код в третьем задании по условию и получим:

Рисунок 7 – Окно до изменения

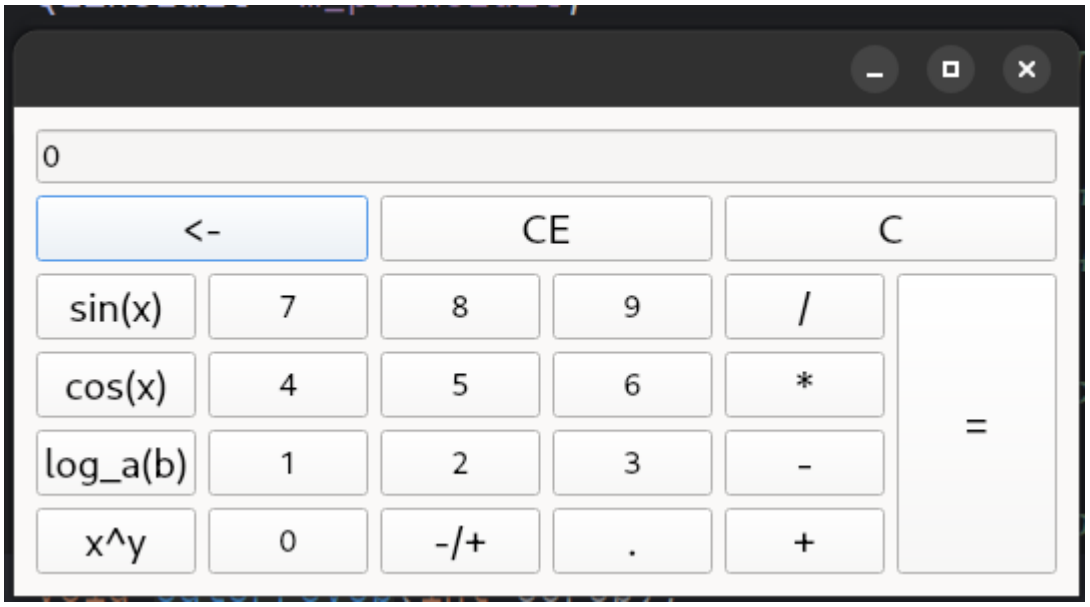


Рисунок 8 – Окно после изменения

Теперь посчитаем $log_5 7$ и получим:

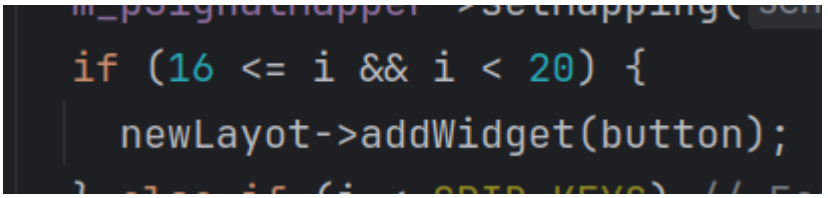
Рисунок 9 – Проверка корректности работы

```
#define SIN 16
#define COS 17
#define LOG 18
#define EXP 19
#define EQ 20
#define BKSP 30
#define CLR 31
#define CLR_ALL 32
// количество кнопок в
#define GRID_KEYS 20
```

Рисунок 10 – Изменённый фрагмент программы

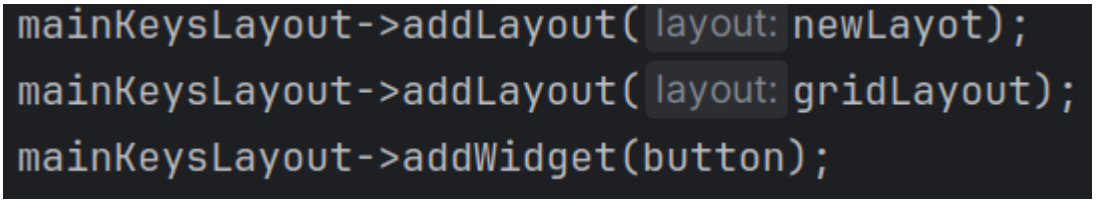
```
_btnDescr.push_back( t: BtnDescr( str: "sin(x)", i: SIN));
_btnDescr.push_back( t: BtnDescr( str: "cos(x)", i: COS));
_btnDescr.push_back( t: BtnDescr( str: "log_a(b)", i: LOG));
_btnDescr.push_back( t: BtnDescr( str: "x^y", i: EXP));
```

Рисунок 11 – Изменённый фрагмент программы



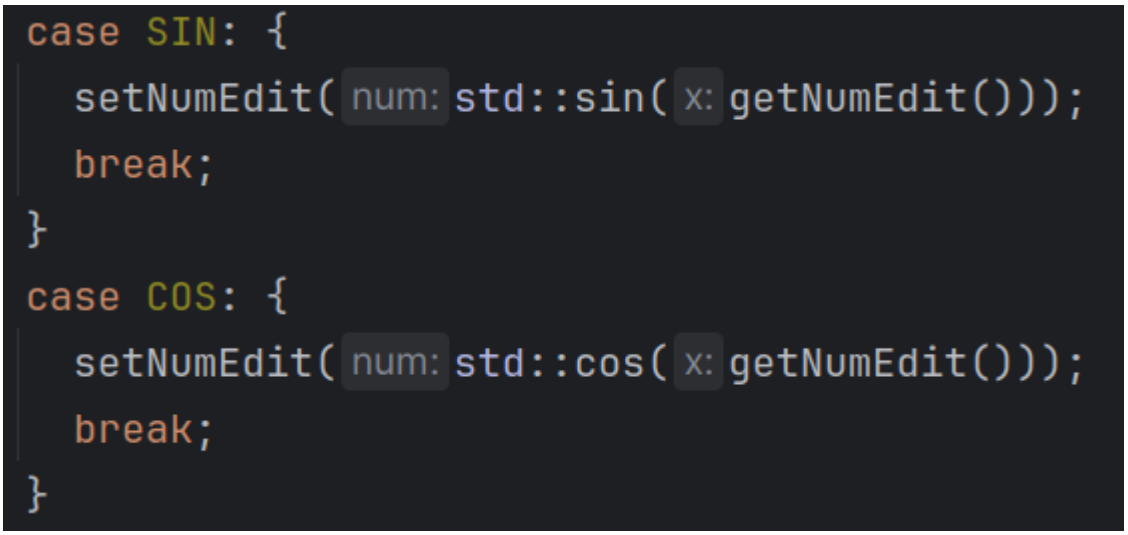
```
if (16 <= i && i < 20) {
    newLayot->addWidget(button);
```

Рисунок 12 – Изменённый фрагмент программы



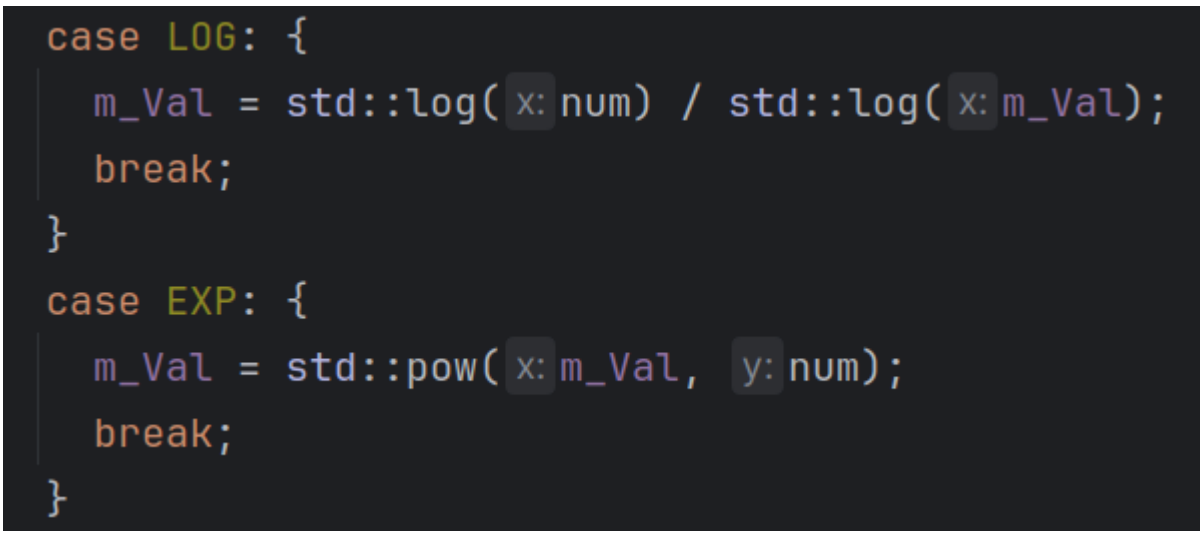
```
mainKeysLayout->addLayout( layout: newLayot);
mainKeysLayout->addLayout( layout: gridLayout);
mainKeysLayout->addWidget(button);
```

Рисунок 13 – Изменённый фрагмент программы



```
case SIN: {
    setNumEdit( num: std::sin( x: getNumEdit()));
    break;
}
case COS: {
    setNumEdit( num: std::cos( x: getNumEdit()));
    break;
}
```

Рисунок 14 – Изменённый фрагмент программы



```
case LOG: {
    m_Val = std::log( x: num) / std::log( x: m_Val);
    break;
}
case EXP: {
    m_Val = std::pow( x: m_Val,  y: num);
    break;
}
```

Рисунок 15 – Изменённый фрагмент программы

Как видно, калькулятор работает верно. Теперь приступим к последнему заданию:

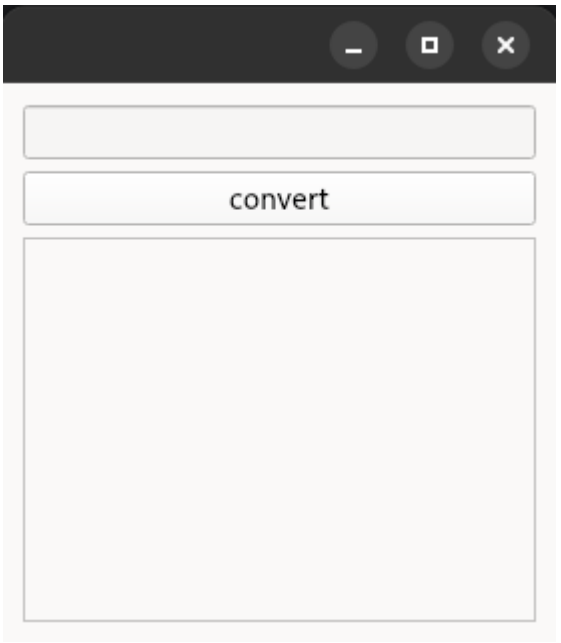


Рисунок 16 – Начальное окно

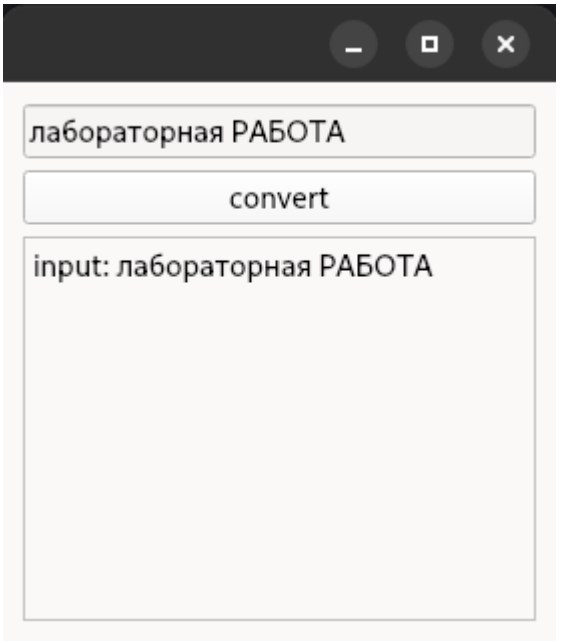


Рисунок 17 – После нажатии кнопки Enter

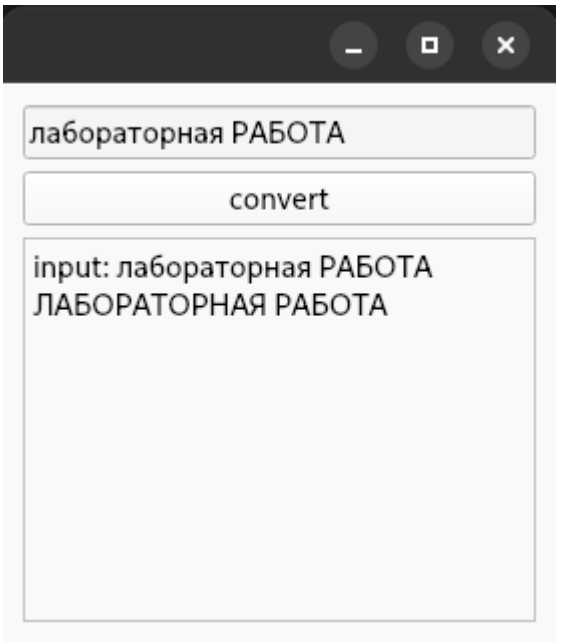


Рисунок 18 – После первого нажатия на кнопку

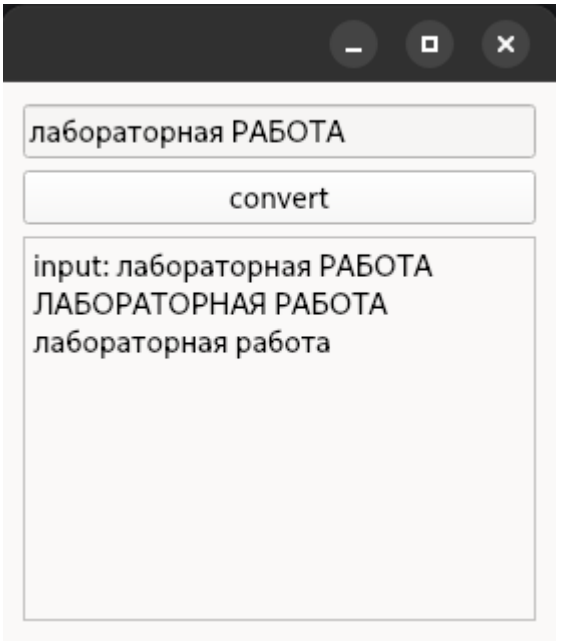


Рисунок 19 – После второго нажатия на кнопку

```cpp
#include <QApplication>
#include "Win.h++"

int main(int argc, char* argv[]) {
  QApplication app( &argc, argv);
  Win win;
  win.show();

  return app.exec(); // запускаем цикл обработки сообщений
}
```

Рисунок 20– main.cpp

```cpp
#include <QWidget>
#include <QLineEdit>
#include <QPushButton>
#include <QTextEdit>
#include <QVBoxLayout>
#include <QValidator>
#include <iostream>
#include <QString>

class Win : public QWidget {
 Q_OBJECT
 private slots:
  void add();
  void convert();
 protected:
  QLineEdit* lineEdit;
  QPushButton* button;
  QTextEdit* textEdit;
  QVBoxLayout* layout;
  int prevConvert;
 public:
  explicit Win(QWidget* parent = nullptr);
  ~Win() override = default;
};
```

Рисунок 21 – Win.h++

```cpp
#include "Win.h++"

Win::Win(QWidget* parent) : QWidget(parent), prevConvert(0) {
  lineEdit = new QLineEdit("", parent: this);
  button = new QPushButton( text: "convert", parent: this);
  textEdit = new QTextEdit( parent: this);
  layout = new QVBoxLayout( parent: this);
  textEdit->setReadOnly( ro: true);
  layout->addWidget(lineEdit);
  layout->addWidget(button);
  layout->addWidget(textEdit);
  connect( sender: lineEdit, signal: SIGNAL(editingFinished()), receiver: this, member: SLOT(add()));
  connect( sender: button, signal: SIGNAL(clicked()), receiver: this, member: SLOT(convert()));
}

void Win::add() {
  textEdit->setText(QString("input: " + lineEdit->text()));
  prevConvert = 0;
}

void Win::convert() {
  if (prevConvert == 1) {
    textEdit->append( text: lineEdit->text().toLower());
  } else if (prevConvert == 0) {
    textEdit->append( text: lineEdit->text().toUpper());
  }
  ++prevConvert;
}
```

Рисунок 22 – Win.cpp

Также приложение сделано так, чтобы после второго нажатия на кнопку ничего не происходило, а после редактирования поля ввода и нажатия кнопки Enter очищось поле вывода.

Вывод

В ходе лабораторной работы были получены навыки построения приложений, имеющих графический интерфейс пользователя с использованием кроссплатформенной библиотеки Qt, также навыки работы с QtDesigner и без него.