

**Лабораторная работа №2**  
**«ИССЛЕДОВАНИЕ СПОСОБОВ ИСПОЛЬЗОВАНИЯ**  
**СИГНАЛОВ И СЛОТОВ В QT-ПРИЛОЖЕНИЯХ»**

**Цель работы**

Исследовать принцип работы механизма сигналов и слотов фреймворка Qt. Приобрести практические навыки применения сигналов и слотов при разработке Qt-приложений.

**Постановка задачи**

1. Изучить принципы работы механизма сигналов и слотов в Qt, способы соединения сигналов и слотов (выполняется в ходе домашней подготовки к лабораторной работе).
2. Создать проект Qt Gui Application.
3. Создать класс-наследник класса QLabel, добавив собственный сигнал, который будет посылаться, когда значение QLabel равно числу, большему десяти.
4. Разместить на форме виджеты QLineEdit, QPushButton, два виджета QPlainTextEdit и виджет созданного на шаге 3 наследника QLabel.
5. Обеспечить изменение названия заголовка окна приложения на значение, введенное в QLineEdit при нажатии на кнопку.
6. Создать собственный слот для MainWindow, который будет копировать текст из первого QPlainTextEdit во второй, заменяя все символы 'а' на '\*'.
7. Подключить слот, созданный на предыдущем этапе к textChanged сигналу первого QPlainTextEdit, таким образом обеспечив автоматическое копирование.

8. Создать собственный слот для MainWindow, который будет выводить количество '\*' во втором QPlainTextEdit в QLabel.

9. Подключить слот, созданный на предыдущем этапе к textChanged сигналу второго QPlainTextEdit, таким образом обеспечив автоматическое подсчет количества символов '\*'.

10. Подключить слот setDisabled первого QPlainTextEdit к сигналу, созданному на шаге 3.3, тем самым обеспечив запрет на дальнейший ввод (setDisabled слот) при вводе более десяти символов 'a'.

11. Выполнить экспериментальное исследование полученного приложения, выполняя ввод тестовых последовательностей с различным количеством символов 'a' и различным их положением во вводимой строке: в начале, в середине и в конце.

## Ход работы

1. Был создан класс-наследник Label класса QLabel, в котором был добавлен сигнал isLimit(), который будет посылаться, когда значение QLabel равно числу, большему десяти (Приложение А – Листинг A1).

2. Далее были размещены на форме виджеты QLineEdit, QPushButton, два виджета QPlainTextEdit и виджет созданного на шаге 1 наследника QLabel.

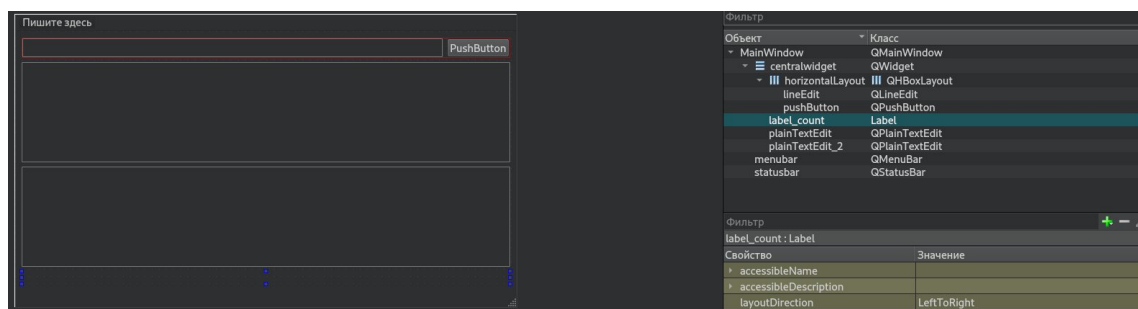


Рисунок 1 – Форма приложения

3. Затем была добавлена возможность изменения названия заголовка окна приложения на значение, введенное в QLineEdit при нажатии на кнопку

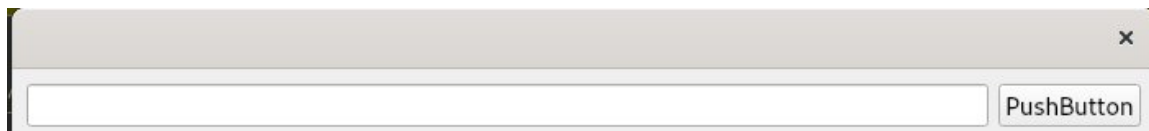


Рисунок 2 – Изменение заголовка окна при пустом QLineEdit



Рисунок 3 – Изменение заголовка окна на значение QLineEdit

4. После чего была реализована возможность копировать текст из первого QPlainTextEdit во второй, заменяя все символы 'а' на '\*', используя механизм сигналов и слотов.

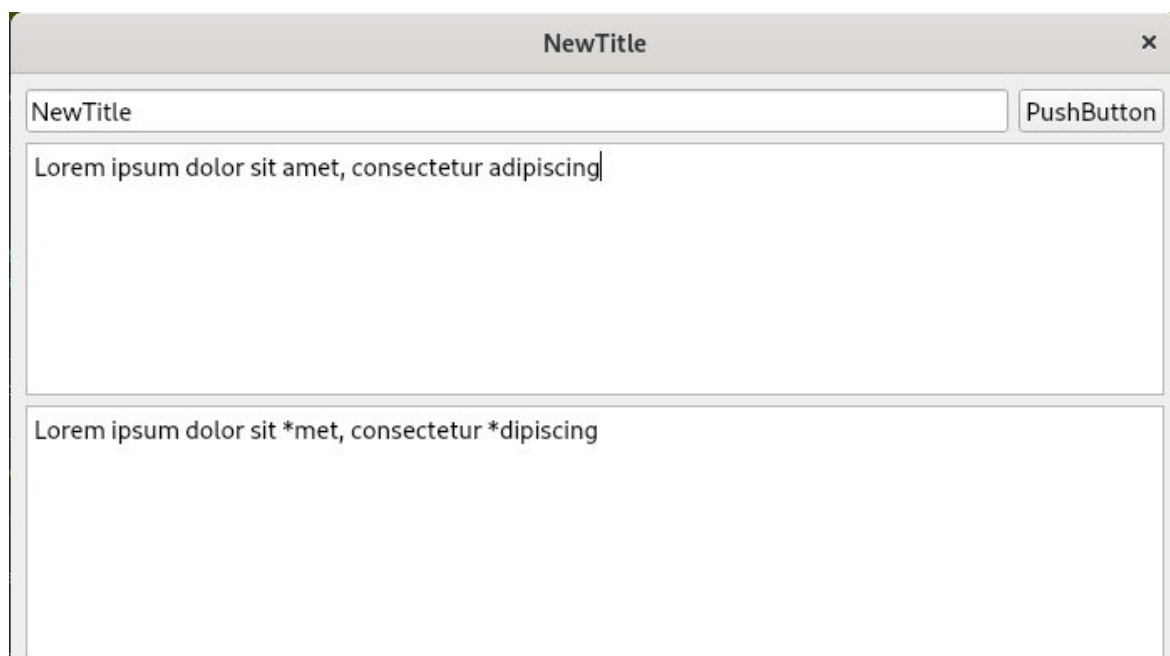


Рисунок 4 – Копирование из первого QPlainTextEdit во второй

5. Затем была добавлена возможность подсчёта символов ‘\*’ во втором QPlainTextEdit и блокировки дальнейшего ввода при достижении количества символов ‘\*’ равным 10.

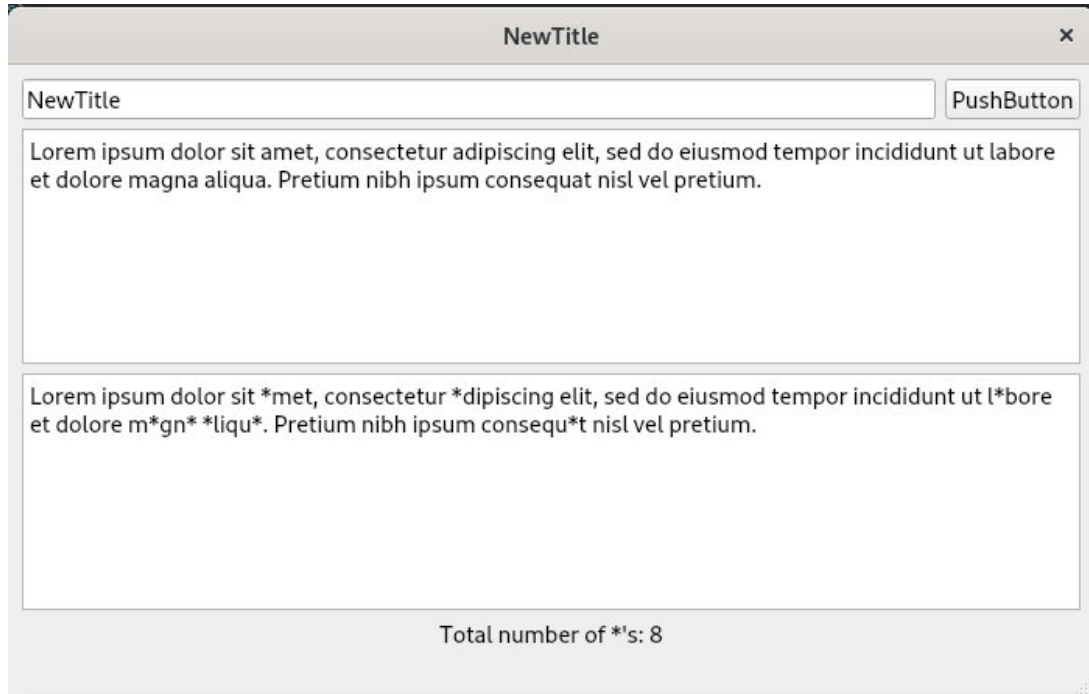


Рисунок 5 – Подсчёт количества символов ‘\*’

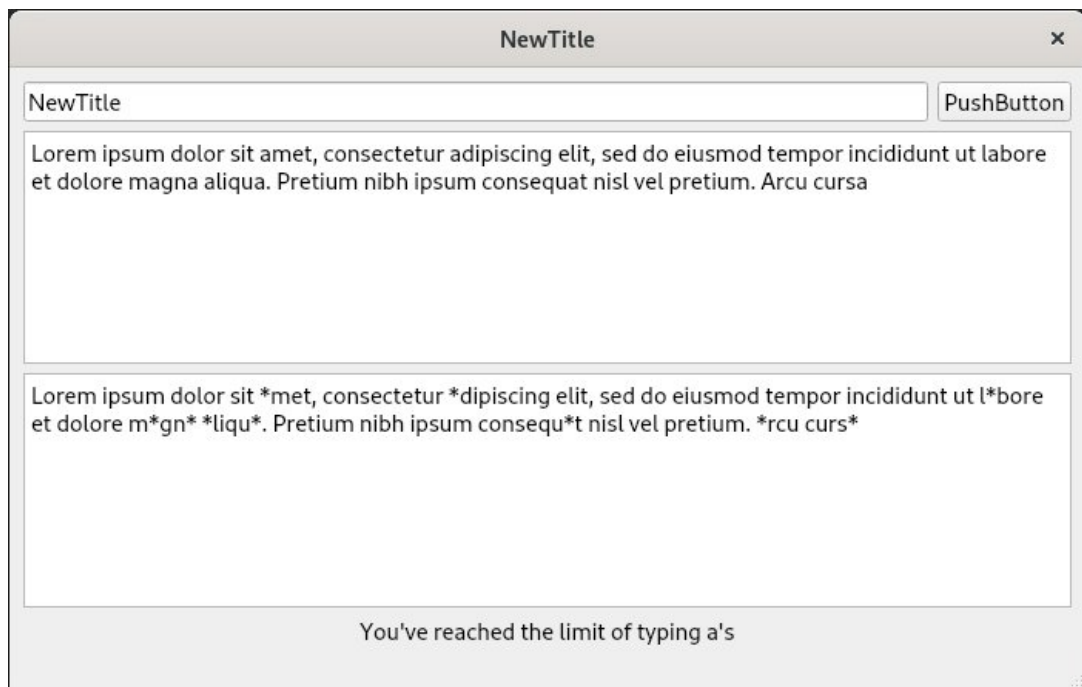


Рисунок 6 – Блокировка ввода при достижении максимального количества символов ‘\*’

## ВЫВОДЫ

В ходе выполнения лабораторной работы были исследованы принципы работы механизма сигналов и слотов фреймворка Qt. Механизм сигналов и слотов является главной особенностью фреймворка Qt и используется для коммуникации между объектами. Он отличается от механизма callback тем, что не требует жесткой связи между функцией, которая вызывает обратный вызов, и функцией-обработчиком, а также сигналов и слотов типобезопасен.

## ПРИЛОЖЕНИЕ А

### «ИСХОДНЫЙ КОД ПРОГРАММЫ»

#### Листинг А1 – Файл label.h

```
#ifndef LABEL_H
#define LABEL_H

#include <QLabel>

class Label : public QLabel
{
    Q_OBJECT
public:
    Label(QWidget *parent);
signals:
    void isLimit();

};

#endif // LABEL_H
```

#### Листинг А2 – Файл label.cpp

```
#include "label.h"

Label::Label(QWidget *parent)
    : QLabel(parent)
{

}
```

#### Листинг А3 – Файл 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 changeTitle();
    void replaceALetters();
    void countStars();
    void setDisabled();
};
#endif // MAINWINDOW_H
```

#### Листинг А4 – Файл mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(changeTitle()));
    connect(ui->plainTextEdit, SIGNAL(textChanged()), this,
SLOT(replaceALetters()));
    connect(ui->plainTextEdit_2, SIGNAL(textChanged()), this,
SLOT(countStars()));
    connect(ui->label_count, SIGNAL(isLimit()), this, SLOT(setDisabled()));
}

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

void MainWindow::changeTitle()
{
    ui->lineEdit->text().size()
QWidget::setWindowTitle(ui->lineEdit->text()) : QWidget::setWindowTitle(" ");
}

void MainWindow::replaceALetters()
{
    u i - > p l a i n T e x t E d i t _ 2 - > s e t P l a
i n T e x t ( u i - > p l a i n T e x t E d i t - > t o P l a i n T e x t ( ) . r e p l
a c e ("a", "*").replace('A', '*'));
}

void MainWindow::countStars()
{
    ui->label_count->setText(QString("Total number of
*'s: %0").arg(ui->plainTextEdit_2->toPlainText().count('*')));
    if (ui->plainTextEdit_2->toPlainText().count('*') >= 10) {emit
ui->label_count->isLimit();};
```

```

}

void MainWindow::setDisabled()
{
    ui->plainTextEdit->setReadOnly(true);
    ui->label_count->setText("You've reached the limit of typing a's");
}

```

#### Листинг А5 – Файл mainwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>
    <widget class="QMainWindow" name="MainWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>698</width>
                <height>406</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>MainWindow</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <layout class="QVBoxLayout" name="verticalLayout">
                <item>
                    <layout class="QHBoxLayout" name="horizontalLayout">
                        <item>
                            <widget class="QLineEdit" name="lineEdit"/>
                        </item>
                        <item>
                            <widget class="QPushButton" name="pushButton">
                                <property name="text">
                                    <string>PushButton</string>
                                </property>
                            </widget>
                        </item>
                    </layout>
                </item>
                <item>
                    <widget class="QPlainTextEdit" name="plainTextEdit"/>
                </item>
                <item>
                    <widget class="QPlainTextEdit" name="plainTextEdit_2">
                        <property name="enabled">
                            <bool>true</bool>
                        </property>
                        <property name="textInteractionFlags">
                            <set>Qt::NoTextInteraction</set>
                        </property>
                    </widget>
                </item>
                <item>

```



```

    <widget class="Label" name="label_count">
      <property name="text">
        <string/>
      </property>
      <property name="alignment">
        <set>Qt::AlignCenter</set>
      </property>
    </widget>
  </item>
</layout>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>698</width>
      <height>23</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<customwidgets>
  <customwidget>
    <class>Label</class>
    <extends>QLabel</extends>
    <header>label.h</header>
  </customwidget>
</customwidgets>
<resources/>
<connections/>
</ui>

```