**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Отчет

По лабораторной работе №1-2

На тему: “Исследование принципов работы платформы QT и среды разработки QT Creator. Исследование способов использования сигналов и слотов в QT-приложениях”

По дисциплине: “Кроссплатформенное программирование”

Выполнил:

ст.гр. ИС/б-17-2

Черняев Н.Г.

Проверил:

Строганов В.А.

Севастополь

2021

1 ЦЕЛЬ РАБОТЫ

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

2 ПОСТАНОВКА ЗАДАЧИ

2.1. Создать проект Qt Gui Application.

2.2. Создать класс-наследник класса QLabel, добавив собственный сигнал, который будет посылаться, когда значение QLabel равно числу, большему десяти.

2.3. Разместить на форме виджеты QLineEdit, QPushButton, два виджета QPlainTextEdit и виджет созданного на шаге 2.2 наследника QLabel.

2.4. Обеспечить изменение названия заголовка окна приложения на значение, введенное в QLineEdit при нажатии на кнопку.

2.5. Создать собственный слот для MainWindow, который будет копировать текст из первого QPlainTextEdit во второй, заменяя все символы ‘a’ на ‘\*’.

2.6. Подключить слот, созданный на предыдущем этапе к textChanged сигналу первого QPlainTextEdit, таким образом обеспечив автоматическое копирование.

2.7. Создать собственный слот для MainWindow, который будет выводить количество ‘\*’ во втором QPlainTextEdit в QLabel.

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

2.9. Подключить слот setDisabled первого QPlainTextEdit к сигналу, созданному на шаге 2.2, тем самым обеспечив запрет на дальнейший ввод (setDisabled cлот) при вводе более десяти символов ‘a’.

2.10. 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.

3 ТЕКСТ ПРОГРАММЫ

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 **copyTextToSecondTextBox**();

void **getCountOfSpecialCharacters**();

void **disablePlainTextEdit**();

};

#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);

connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(changeTitle()));

connect(ui->plainTextEdit, SIGNAL(textChanged()), this, SLOT(copyTextToSecondTextBox()));

connect(ui->plainTextEdit\_2, SIGNAL(textChanged()), this, SLOT(getCountOfSpecialCharacters()));

connect(ui->label, SIGNAL(ownSignal()), this, SLOT(disablePlainTextEdit()));

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**changeTitle**()

{

QString newTitle= ui->lineEdit->text();

this->setWindowTitle(newTitle);

ui->lineEdit->clear();

}

void MainWindow::**copyTextToSecondTextBox**()

{

QString text = ui->plainTextEdit->toPlainText();

text.replace("a", "\*");

ui->plainTextEdit\_2->setPlainText(text);

}

void MainWindow::**getCountOfSpecialCharacters**()

{

int count = ui->plainTextEdit\_2->toPlainText().count("\*");

ui->label->setCount(count);

}

void MainWindow::**disablePlainTextEdit**()

{

ui->plainTextEdit->setDisabled(true);

}

mylabel.h:

#ifndef MYLABEL\_H

#define MYLABEL\_H

#include <QLabel>

class **MyLabel** : public QLabel

{

Q\_OBJECT

private:

const int limit = 10;

public:

**MyLabel**(QWidget \*parent = 0);

public slots:

void **setCount**(int);

signals:

void **ownSignal**();

};

#endif // MYLABEL\_H

mylabel.cpp:

#include "mylabel.h"

MyLabel::**MyLabel**(QWidget \*parent) : QLabel(*parent*)

{

}

void MyLabel::**setCount**(int value)

{

if (value > limit)

{

emit ownSignal();

}

this->setText(QString::number(value));

}

4 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Запустив программу первый раз, мы видим интерфейс, представленный на рисунке 1.

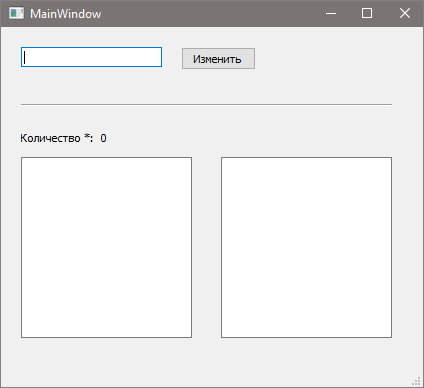


Рисунок 1 – Старт программы

Введем новое название окна в строку ввода и нажмем кнопку “Изменить” (Рисунок 2). Результат изменения окна приложения представлен на рисунке 3.

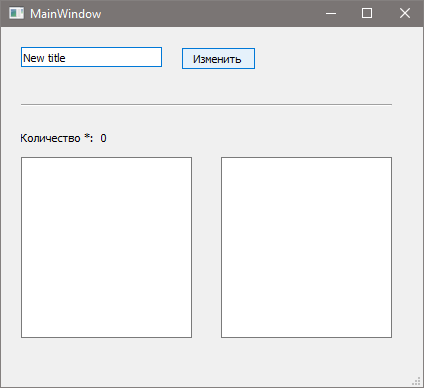


Рисунок 2 – Процесс изменения названия окна

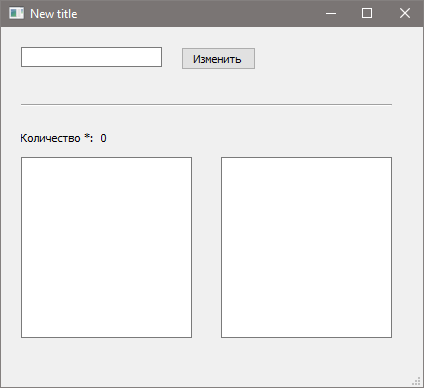


Рисунок 3 – Результат изменения окна приложения

Проверим замену символов “а” на “\*” путем ввода различных последовательностей символов в поле. Данный процесс представлен на рисунке 4.

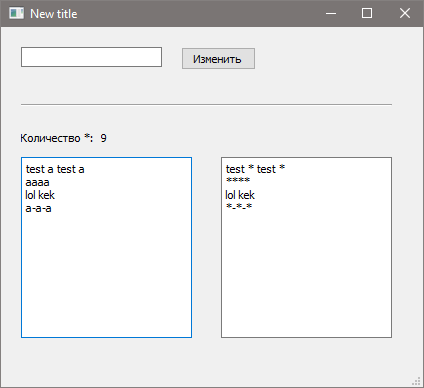


Рисунок 4 – Процесс замены символа “a” на “\*” и блокировка текстового поля

Проверим блокировку первого пол при достижении количества символов “\*” больше 10 во втором поле. Результат проверки изображен на рисунке 5.

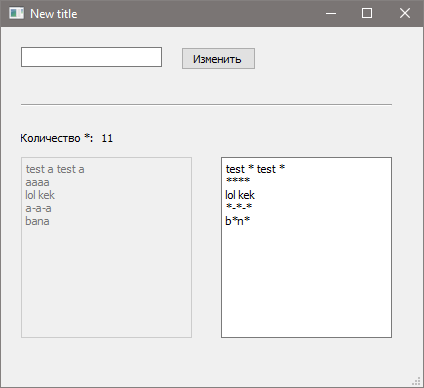


Рисунок 5 – Результат блокировки поля ввода при достижении количества \* больше 10

ВЫВОДЫ

В ходе выполнения лабораторной работы был изучен фундаментальный механизм Qt, позволяющий связывать объекты друг с другом – сигналы и слоты. Сигналы и слоты гораздо удобнее функций обратного вызова, т.к. привязка слотов происходит при помощи функции connect, что позволяет избавиться от жесткой привязки обратного вызова с вызывающей его функций и быть уверенным, что функция вызывается с корректными аргументами.