

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Кафедра «Компьютерная безопасность»

**ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ №25**

по дисциплине

«Языки программирования»

Работу выполнил
студент группы СКБ-222

подпись, дата

Д.А. Спиридонов

Работу проверил

подпись, дата

С.А. Булгаков

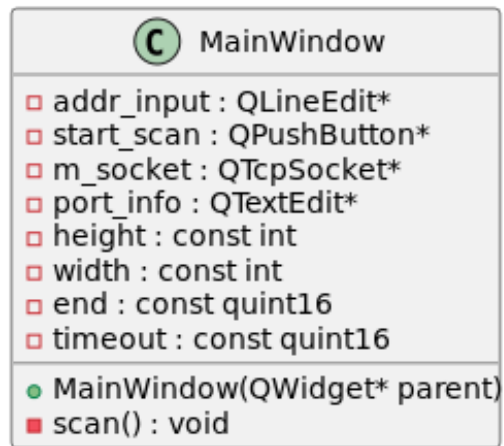
Москва 2023

Содержание

Приложение А	3
А.1 UML-диаграммы класса MainWindow	3
Приложение В	4
В.1 Исходный код MainWindow.h	4
В.2 Исходный код MainWindow.cpp	5
В.3 Исходный код main.cpp	6

Приложение А

А.1 UML-диаграммы класса MainWindow



Приложение В

В.1 Исходный код MainWindow.h

```
#ifndef __MAINWINDOW_H__
#define __MAINWINDOW_H__

#include <QtGlobal>

#include <QTcpSocket>
#include <QApplication>

#include <QVBoxLayout>
#include <QPushButton>
#include <QLineEdit>
#include <QTextEdit>
#include <QWidget>

class MainWindow : public QWidget
{
    Q_OBJECT
public:
    MainWindow(QWidget* parent=nullptr);
private slots:
    void scan();
private:
    const quint16 end = 1024; // Upper bound of scanning range
    const quint16 timeout = 50; // timeout for scanning

    const int height = 480;
    const int width = 640;

    QPushButton* start_scan;
    QLineEdit* addr_input;
    QTextEdit* port_info;

    QTcpSocket* m_socket;
};

#endif // __MAINWINDOW_H__
```

B.2 Исходный код MainWindow.cpp

```
#include "MainWindow.h"

MainWindow::MainWindow(QWidget* parent)
    : QWidget(parent)
{
    this->setWindowTitle(tr("Port scanner"));
    this->setFixedSize(width, height);

    m_socket = new QTcpSocket(this);

    QVBoxLayout* main_layout = new QVBoxLayout(this);

    addr_input = new QLineEdit;
    addr_input->setPlaceholderText(tr("Введите сюда адрес"));
    main_layout->addWidget(addr_input);

    start_scan = new QPushButton(tr("Начать сканирование"));
    connect(start_scan, &QPushButton::clicked, this, &MainWindow::scan);
    main_layout->addWidget(start_scan);

    port_info = new QTextEdit;
    main_layout->addWidget(port_info);
    port_info->setPlaceholderText(tr("Здесь будет информация о портах..."));
    port_info->setReadOnly(true);
}

void MainWindow::scan()
{
    if(addr_input->text().isEmpty())
    {
        return;
    }

    port_info->clear();
    addr_input->setReadOnly(true);
    start_scan->setEnabled(false);

    port_info->append("Starting scan...");
    QString host = addr_input->text();
    for(quint16 port = 0; port < end; port++)
    {
        m_socket->connectToHost(host, port);
        bool is_connected = m_socket->waitForConnected(timeout);

        QString port_num = QString::number(port);
        if(is_connected)
        {
            port_info->append("Порт " + port_num + ":\t открыт");
            m_socket->disconnectFromHost();
        }
        else
        {
            port_info->append("Порт " + port_num + ":\t закрыт");
        }
        QApplication::processEvents(); // For responding of application
    }
    port_info->append("Сканирование завершено");

    addr_input->setReadOnly(false);
    start_scan->setEnabled(true);
}
```

В.3 Исходный код main.cpp

```
#include <QApplication>
#include "MainWindow.h"

int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    MainWindow main_window;
    main_window.show();
    return app.exec();
}
```