**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Расчетно-графическое задание

Дисциплина: Объектно-ориентированное программирование

по теме “Домофон”

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверил: Буханов Д.Г.

**Белгород 2019**

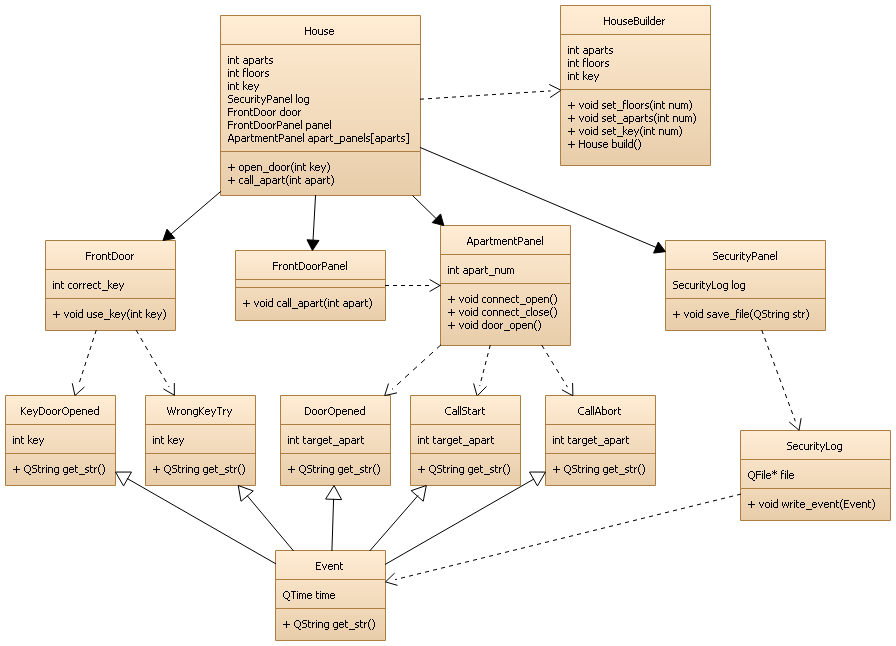
**Цель работы:** Смоделировать работу домофона в многоквартирном доме.

**Постановка задачи:** Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена панель с кнопками на каждую квартиру, микрофон и динамик. В каждой квартире имеется кнопка «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, в квартире имеется микрофон и динамик.

Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры. В квартире раздается звонок. При нажатии кнопки «СВЯЗЬ» внутри квартиры устанавливается звуковое сообщение между квартирой и посетителем. Звуки, произносимые посетителем в микрофон, установленный в подъезде, должны быть слышны в динамике, установленном в квартире. При нажатии на кнопку «ОТКРЫТЬ» дверь в подъезд открывается. Нажатие на кнопку «БЛОКИРОВКА» отключает подачу звонка в квартиру при нажатии на кнопку квартиры в подъезде. Повторное нажатие на кнопку «БЛОКИРОВКА» вновь включает подачу звонка.

Сведения о посещении дома (когда и из какой квартиры открывали замок домофона) сохраняются в журнале домофона. По прошествии суток журнал домофона пересылается на пульт начальника охраны. После пересылки данных журнал обнуляется.

**Диаграмма классов**



**Листинг программы**

**Файл ApartmentSelect.hpp(cpp)**

#pragma once

#include <QtWidgets>

#include "SignalButton.hpp"

#include "ApartmentsPanel.hpp"

class ApartmentSelect:public QWidget{

Q\_OBJECT

int floors;

int apart\_per\_floor;

QList<SignalButton\*> apartment\_buttons;

QGridLayout\* grid\_layout;

QVBoxLayout\* layout;

public:

ApartmentSelect(int floors,int apart):floors(floors),apart\_per\_floor(apart),QWidget(){

grid\_layout = new QGridLayout();

QLabel\* lbl = new QLabel("Выберите номер квартиры для связи.");

layout = new QVBoxLayout();

layout->addWidget(lbl);

for(int i=0,count=1;i<floors;i++){

for(int j=0;j<apart\_per\_floor;j++,count++){

apartment\_buttons.push\_back(new SignalButton(count,QString::number(count)));

grid\_layout->addWidget(apartment\_buttons.last(),i,j);

connect(apartment\_buttons[count-1],SIGNAL(b\_clicked(int)),this,SLOT(show\_apart\_window(int)));

}

};

layout->addLayout(grid\_layout);

this->setLayout(layout);

};

public slots:

void show\_apart\_window(int num);

void apart\_open\_door(int apart);

void apart\_open\_conn(int apart);

void apart\_close\_conn(int apart);

signals:

void open\_door(int apart);

void open\_conn(int apart);

void close\_conn(int apart);

};

#include "ApartmentSelect.hpp"

void ApartmentSelect::show\_apart\_window(int num){

ApartmentWindow\* window = new ApartmentWindow(num);

connect(window,SIGNAL(door\_open(int)),this,SLOT(apart\_open\_door(int)));

connect(window,SIGNAL(connect\_open(int)),this,SLOT(apart\_open\_conn(int)));

connect(window,SIGNAL(connect\_block(int)),this,SLOT(apart\_close\_conn(int)));

window->show();

};

void ApartmentSelect::apart\_open\_door(int apart){

emit open\_door(apart);

};

void ApartmentSelect::apart\_open\_conn(int apart){

emit open\_conn(apart);

};

void ApartmentSelect::apart\_close\_conn(int apart){

emit close\_conn(apart);

};

**Файл ApartmentsPanel.hpp(cpp)**

#pragma once

#include <QtWidgets>

class ApartmentWindow: public QWidget{

Q\_OBJECT

private:

int apart\_num;

QPushButton \*b\_open,\*b\_connect,\*b\_close;

QHBoxLayout\* h\_layout;

QVBoxLayout\* layout;

public:

ApartmentWindow(int num):QWidget(),apart\_num(num){

this->setWindowFlags(Qt::CustomizeWindowHint | Qt::WindowTitleHint);

this->setWindowTitle(QStringLiteral("Квартира %1").arg(apart\_num));

QLabel\* lbl = new QLabel(QStringLiteral("Панель квартиры номер %1").arg(apart\_num));

lbl->setAlignment(Qt::AlignCenter);

h\_layout = new QHBoxLayout();

layout = new QVBoxLayout();

b\_open = new QPushButton("Открыть");

b\_open->setEnabled(false);

b\_close = new QPushButton("Блокировка");

b\_close->setEnabled(false);

b\_connect = new QPushButton("Связь");

connect(this->b\_open,SIGNAL(clicked()),this,SLOT(door\_button\_click()));

connect(this->b\_connect,SIGNAL(clicked()),this,SLOT(connect\_button\_click()));

connect(this->b\_close,SIGNAL(clicked()),this,SLOT(close\_button\_click()));

h\_layout->addWidget(b\_connect);

h\_layout->addWidget(b\_open);

h\_layout->addWidget(b\_close);

layout->addWidget(lbl);

layout->addLayout(h\_layout);

this->setLayout(layout);

};

signals:

void door\_open(int apart\_num);

void connect\_open(int apart\_num);

void connect\_block(int apart\_num);

private slots:

void door\_button\_click();

void connect\_button\_click();

void close\_button\_click();

};

#include "ApartmentsPanel.hpp"

void ApartmentWindow::door\_button\_click(){

emit door\_open(this->apart\_num);

close();

};

void ApartmentWindow::connect\_button\_click(){

emit connect\_open(this->apart\_num);

this->b\_connect->setEnabled(false);

this->b\_open->setEnabled(true);

this->b\_close->setEnabled(true);

};

void ApartmentWindow::close\_button\_click(){

emit connect\_block(this->apart\_num);

close();

};

**Файл FrontDoor.hpp(cpp)**

#include <QtWidgets>

class FrontDoorPanel: public QWidget{

Q\_OBJECT

int correct\_key;

QHBoxLayout\* layout;

QPushButton\* key\_open,\*call\_open;

public:

FrontDoorPanel(int key):QWidget(),correct\_key(key){

layout = new QHBoxLayout();

key\_open = new QPushButton("Открыть ключем");

call\_open = new QPushButton("Позвонить в домофон");

layout->addWidget(key\_open);

layout->addWidget(call\_open);

this->setLayout(layout);

connect(key\_open,SIGNAL(clicked()),this,SLOT(b\_key\_slot()));

connect(call\_open,SIGNAL(clicked()),this,SLOT(b\_call\_slot()));

};

signals:

void b\_key\_click();

void b\_call\_click();

void record\_open\_door(int key);

void record\_wrong\_key(int key);

private slots:

void b\_key\_slot();

void b\_call\_slot();

};

#include "FrontDoor.hpp"

void FrontDoorPanel::b\_key\_slot(){

int input\_key = QInputDialog::

getInt(this, tr("QInputDialog::getText()"),

tr("Введите ключ:"), QLineEdit::Password);

if(input\_key == correct\_key){

emit record\_open\_door(correct\_key);

}

else{

QMessageBox msgBox;

msgBox.setText("Неверный ключ.");

msgBox.exec();

emit record\_wrong\_key(input\_key);

}

};

void FrontDoorPanel::b\_call\_slot(){

emit b\_call\_click();

};

**Файл HousePhone.hpp**

#pragma once

#include <QtWidgets>

#include "ApartmentsPanel.hpp"

#include "ApartmentSelect.hpp"

#include "FrontDoor.hpp"

#include "LogWindow.hpp"

class House{

int apart\_per\_floor,floors;

int correct\_key;

ApartmentSelect\* house;

FrontDoorPanel\* panel;

LogWindow\* log;

public:

House(int apf,int f, int key):apart\_per\_floor(apf),floors(f),correct\_key(key){

house = new ApartmentSelect(7,3);

panel = new FrontDoorPanel(1323);

log = new LogWindow();

QObject::connect(house,SIGNAL(open\_door(int)),log,SLOT(record\_open\_door(int)));

QObject::connect(house,SIGNAL(open\_conn(int)),log,SLOT(record\_connect\_open(int)));

QObject::connect(house,SIGNAL(close\_conn(int)),log,SLOT(record\_connect\_close(int)));

QObject::connect(panel,SIGNAL(record\_open\_door(int)),log,SLOT(record\_open\_with\_key(int)));

QObject::connect(panel,SIGNAL(record\_wrong\_key(int)),log,SLOT(record\_wrong\_key(int)));

QObject::connect(panel,SIGNAL(b\_call\_click()),house,SLOT(show()));

panel->show();

log->show();

};

};

class HouseBuilder{

int aparts = 2;

int floors = 2;

int key = 1234;

public:

void set\_floors(int num){

this->floors = num;

};

void set\_aparts(int num){

this->aparts = num;

}

void set\_key(int num){

this->key = num;

}

House build(){

return House(aparts,floors,key);

};

};

class HouseConfig{

QFile\* config;

QTextStream cfg;

int aparts,floors,key;

HouseBuilder builder;

public:

HouseConfig(){

config = new QFile("config.cfg");

config->open(QIODevice::ReadOnly);

cfg.setDevice(config);

cfg>>aparts>>floors>>key;

builder.set\_aparts(aparts);

builder.set\_floors(floors);

builder.set\_key(key);

};

House get\_result(){

return builder.build();

};

};

**Файл LogWindow.hpp(cpp)**

#pragma once

#include <QtWidgets>

const QString temp\_filename = "temp.log";

class LogWindow: public QWidget{

Q\_OBJECT

QListWidget\* list;

QPushButton\* save\_button;

QFile\* log;

QTextStream log\_stream;

QHBoxLayout\* layout;

public:

LogWindow(){

this->setWindowFlags(Qt::CustomizeWindowHint | Qt::WindowTitleHint);

this->setWindowTitle(QString("Журнал событий"));

list = new QListWidget();

save\_button = new QPushButton("Сохранить лог");

log = new QFile(temp\_filename);

log->remove();

log->open(QIODevice::WriteOnly);

log\_stream.setDevice(log);

layout = new QHBoxLayout();

layout->addWidget(list);

layout->addWidget(save\_button);

this->setLayout(layout);

connect(this->save\_button,SIGNAL(clicked()),this,SLOT(save\_log()));

connect(this->save\_button,SIGNAL(clicked()),this->list,SLOT(clear()));

};

public slots:

void save\_log();

void record\_open\_door(int apart\_num);

void record\_connect\_open(int apart\_num);

void record\_connect\_close(int apart\_num);

void record\_open\_with\_key(int key);

void record\_wrong\_key(int key);

};

#include "LogWindow.hpp"

void LogWindow::record\_open\_door(int apart\_num){

QString time\_str = QTime::currentTime().toString();

QString result\_str = "["+time\_str+"]. Открыта дверь. Квартира: "

+QString::number(apart\_num);

this->list->addItem(result\_str);

this->log\_stream<<result\_str<<endl;

};

void LogWindow::record\_connect\_open(int apart\_num){

QString time\_str = QTime::currentTime().toString();

QString result\_str = "["+time\_str+"]. Звонок в дверь. Квартира: "

+QString::number(apart\_num);

this->list->addItem(result\_str);

this->log\_stream<<result\_str<<endl;

};

void LogWindow::record\_connect\_close(int apart\_num){

QString time\_str = QTime::currentTime().toString();

QString result\_str = "["+time\_str+"]. Звонок сброшен. Квартира: "

+QString::number(apart\_num);

this->list->addItem(result\_str);

this->log\_stream<<result\_str<<endl;

};

void LogWindow::record\_open\_with\_key(int key){

QString time\_str = QTime::currentTime().toString();

QString result\_str = "["+time\_str+"]. Дверь открыта ключем. Ключ: "

+QString::number(key);

this->list->addItem(result\_str);

this->log\_stream<<result\_str<<endl;

};

void LogWindow::record\_wrong\_key(int key){

QString time\_str = QTime::currentTime().toString();

QString result\_str = "["+time\_str+"]. Попытка открыть дверь, ключ неверен. Ключ: "

+QString::number(key);

this->list->addItem(result\_str);

this->log\_stream<<result\_str<<endl;

};

void LogWindow::save\_log(){

QString fileName = QFileDialog::getSaveFileName(this,tr("Сохранить лог"),"/home",tr("log (\*.log)"));

this->log->rename(fileName);

this->log->close();

this->log->setFileName(temp\_filename);

this->log->open(QIODevice::WriteOnly);

};

**Файл main.cpp**

#include <QtWidgets>

#include "HousePhone.hpp"

int main(int argc, char \*argv[]){

QApplication app(argc,argv);

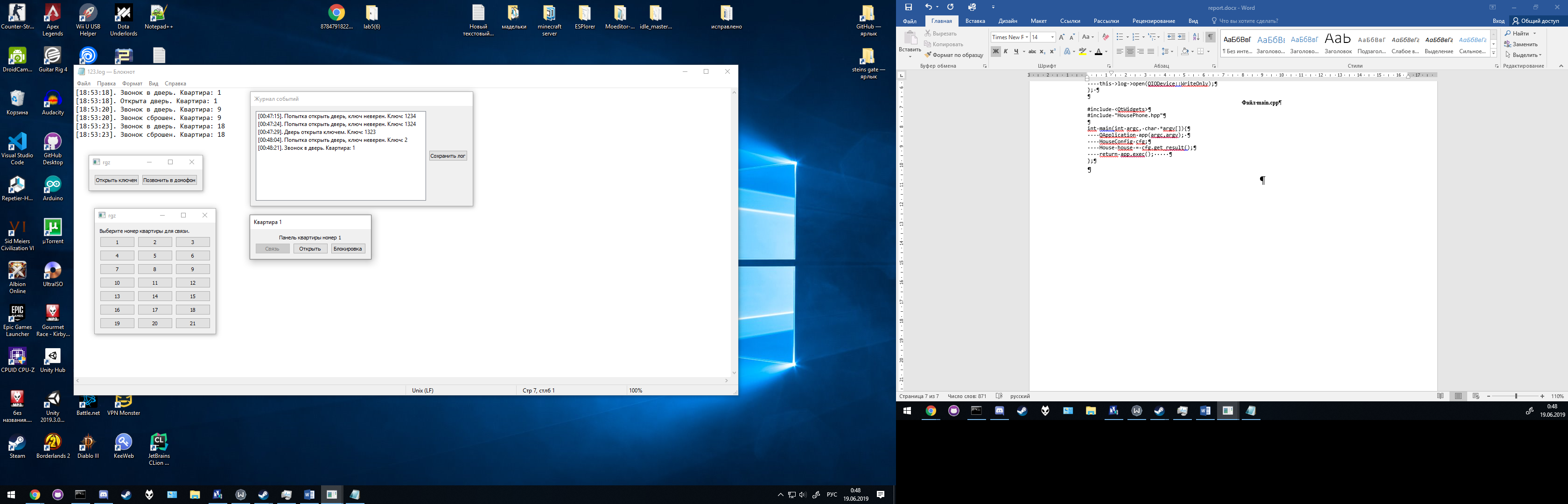
HouseConfig cfg;

House house = cfg.get\_result();

return app.exec();

};

**Результат работы программы**



**Вывод:** выполняя лабораторную работу, я получил навыки работы со фреймворком QT, научился использовать шаблоны проектирования и закрепил навыки создания диаграмм классов.

**Список литературы**

1. Документация к фреймворку QT - <https://doc.qt.io/>
2. Шаблоны проектирования. Строитель.- https://refactoring.guru/ru/design-patterns/builder
3. Шаблоны проектирования. Фабричный метод. - https://refactoring.guru/ru/design-patterns/factory-method