Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Воронежский государственный лесотехнический университет

имени Г.Ф. Морозова»

Кафедра Вычислительной техники и информационных систем

(название кафедры)

**Пояснительная записка**

КУРСОВОЙ РАБОТЫ

(вид работы)

Разработка программного продукта на платформе QT Creator.

(тема)

09.03.02 Информационные системы и технологии

(код и наименование направления подготовки)

По дисциплине: Технологии программирования

|  |  |  |
| --- | --- | --- |
| Студент группы ИС2-191-ОБ  (номер группы)  Руководитель, к.т.н. доцент  (ученая степень, ученое звание) | \_\_\_\_\_\_\_\_\_\_  (подпись)  \_\_\_\_\_\_\_\_\_\_  (подпись) | Величко В. А.  (инициалы и фамилия)  Юров А.Н.  (инициалы и фамилия) |

Воронеж 2021

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный лесотехнический университет   
имени Г.Ф. Морозова»

Кафедра Вычислительной техники и информационных систем

**З А Д А Н И Е**

на курсовую работу по дисциплине

«Технологии программирования»

Студенту 2 курса гр. ИС2-191-ОБ Величко В. А.

(Фамилия И.О.)

09.03.02 Информационные системы и технологии

Срок представления к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

**Тема работы:** Разработка программного продукта на платформе QT Creator.

**Исходные данные для проектирования:** для своей предметной области выполнить проектирование программного продукта для работы с базами данных.

**Перечень вопросов, подлежащих разработке:**

Введение.

Теоретический материал.

Анализ зачади, выбор структур данных, описание используемых переменных.

Проектирование структуры базы данных.

Проектирование программного продукта.

Пример реализации программного продукта.

Системные требования.

Заключение.

Библиографический список.

Руководитель, канд. техн. наук \_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Юров

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ Величко В. А.

(подпись) (число, месяц, год) (инициалы и фамилия)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc74591372)

[1. Теоретический материал 6](#_Toc74591373)

[2. Анализ задачи, выбор структур данных, описание используемых элементов 9](#_Toc74591374)

[3. Проектирование структуры базы данных 11](#_Toc74591375)

[4. Проектирование программного продукта 12](#_Toc74591376)

[5. Пример реализации программного продукта 16](#_Toc74591377)

[6. Системные требования 20](#_Toc74591378)

[ЗАКЛЮЧЕНИЕ 21](#_Toc74591379)

[Список используемых источников 22](#_Toc74591380)

[Приложение. Листинг программы 23](#_Toc74591381)

# **ВВЕДЕНИЕ**

**C++** - компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции.

Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником - языком C, - наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#. Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C.

Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

Целью курсовой работы является, применение полученных знаний по программированию на C++ в QtCreator.

Задачи курсовой работы:

1. Разработать базу данных
2. Создать интерфейс программного продукта
3. Разработать программный продукт

# **1. Теоретический материал**

Бьерн Страуструп высвободил объектно-ориентированный потенциал С путем перенесения возможностей классов Simula 67 в С. Первоначально новый язык носил имя "С с классами" и только потом стал называться C++. Язык C++ достиг популярности, будучи разработанным в Bell Labs, позже он был перенесен в другие индустрии и корпорации. Сегодня это один из наиболее популярных языков программирования в мире. C++ наследует как хорошие, так и плохие стороны С.

Бьерн Страуструп: "Я придумал C++, записал его первоначальное определение и выполнил первую реализацию. Я выбрал и сформулировал критерии проектирования C++, разработал его основные возможности и отвечал за судьбу предложений по расширению языка в комитете по стандартизации C++, - пишет автор самого популярного языка программирования. - Язык C++ многим обязан языку C, и язык C остается подмножеством языка C++ (но в C++ устранены несколько серьезных брешей системы типов C). Я также сохранил средства C, которые являются достаточно низкоуровневыми, чтобы справляться с самыми критическими системными задачами. Язык C, в свою очередь многим обязан своему предшественнику, BCPL; кстати, стиль комментариев // был взят в C++ из BCPL. Другим основным источником вдохновения был язык Simula67. Концепция классов (с производными классами и виртуальными функциями) была позаимствована из него. Средства перегрузки операторов и возможность помещения объявлений в любом месте, где может быть записана инструкция, напоминает Algol68.

Название C++ придумал Рик Масситти. Название указывает на эволюционную природу перехода к нему от C. "++" - это операция приращения в C. Чуть более короткое имя C+ является синтаксической ошибкой; кроме того, оно уже было использовано как имя совсем другого языка. Знатоки семантики C находят, что C++ хуже, чем ++C. Названия D язык не получил, поскольку он является расширением C и в нем не делается попыток исцеляться от проблем путем выбрасывания различных особенностей... Изначально C++ был разработан, чтобы автору и его друзьям не приходилось программировать на ассемблере, C или других современных языках высокого уровня. Основным его предназначением было сделать написание хороших программ более простым и приятным для отдельного программиста. Плана разработки C++ на бумаге никогда не было; проект, документация и реализация двигались одновременно. Разумеется, внешний интерфейс C++ был написан на C++. Никогда не существовало "Проекта C++" и "Комитета по разработке C++". Поэтому C++ развивался и продолжает развиваться во всех направлениях, чтобы справляться со сложностями, с которыми сталкиваются пользователи, а также в процессе дискуссий автора с его друзьями и коллегами".

В языке С++ полностью поддерживаются принципы объектно-ориентированного программирования, включая три кита, на которых оно стоит: инкапсуляцию, наследование и полиморфизм. Инкапсуляция в С++ поддерживается посредством создания нестандартных (пользовательских) типов данных, называемых классами. Язык С++ поддерживает наследование. Это значит, что можно объявить новый тип данных (класс), который является расширением существующего.

Хотя язык С++ справедливо называют продолжением С и любая работоспособная программа на языке С будет поддерживаться компилятором С++, при переходе от С к С++ был сделан весьма существенный скачок. Язык С++ выигрывал от своего родства с языком С в течение многих лет, поскольку многие программисты обнаружили, что для того, чтобы в полной мере воспользоваться преимуществами языка С++, им нужно отказаться от некоторых своих прежних знаний и приобрести новые, а именно: изучить новый способ концептуальности и решения проблем программирования. Перед тем как начинать осваивать С++, Страуструп и большинство других программистов, использующих С++ считают изучение языка С необязательным.

C++ в настоящее время считается господствующим языком, используемым для разработки коммерческих продуктов, 90% игр пишутся на С++ с применением DirectX.

# **2. Анализ задачи, выбор структур данных, описание используемых элементов**

Для реализации поставленной задачи было необходимо разработать программный продукт, которое считывает базу данных.

При создании программного продукта, были использованы следующие классы:

QAction

Класс QAction предоставляет абстрактное действие пользовательского интерфейса, которое может быть вставлено в виджеты. В приложениях множество общих команд может быть вызвано через меню, панели инструментов и сочетания клавиш. Поскольку пользователь ожидает, что каждая команда будет выполняться одним и тем же способом, независимо от использования пользовательского интерфейса, то полезно представить каждую команду как действие.

QTableView

Класс QTableView предоставляет реализацию модели / представления по умолчанию для табличного представления. QTableView реализует табличное представление, которое отображает элементы из модели. Этот класс используется для предоставления стандартных таблиц, которые ранее были предоставлены классом QTable, но с использованием более гибкого подхода, предоставляемого архитектурой модель / представление Qt.

QLable

Виджет QLable обеспечивает отображение текста или изображение. Функциональность взаимодействия с пользователем не предоставляется. Внешний вид метки можно настроить различными способами, и его можно использовать для указания мнемонического ключа фокуса для другого виджета.

QString

Класс QString предоставляет строку символов Unicode. QString хранит строку 16-битных QChar, где каждому QChar соответствует один символ Unicode 4.0. Unicode – это международный стандарт, который поддерживает большинство использующихся сегодня систем письменности.

QSqlDatabase

Класс QSqlDatabase предоставляет интерфейс для подключения к базе данных через соединение. Экземпляр класса QSqlDatabase представляет соединение. Создать соединение можно с помощью одной из статических функций addDatabase() и имя соединения. Соединение известно под своим собственным именем, а не по имени базы данных с которой оно соединяет. Вы можете иметь множество соединений с одной и той же базой данных.

QSqlQuery

Для использования команд SQL, после установления соединения, можно использовать класс QSqlQuery. Запросы оформляются в виде обычной строки, которая передается в конструктор.

# **3. Проектирование структуры базы данных**

Перед тем, как приступить к проектированию приложения, была спроектирована структура базы данных (Рисунок 1). Была создана база данных, включающая в себя 2 таблицы.

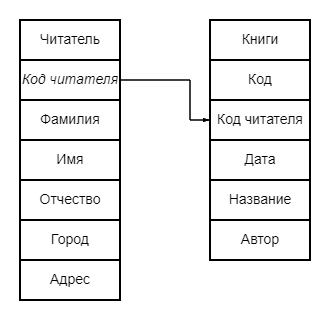
****

Рисунок 1 – Модель базы данных

# **4. Проектирование программного продукта**

Перед тем, как приступить к созданию приложения, была спроектирована схема (Рисунок 2). Был определен нужный функционал и в соответствии с ним, были спроектированы кнопки приложения.

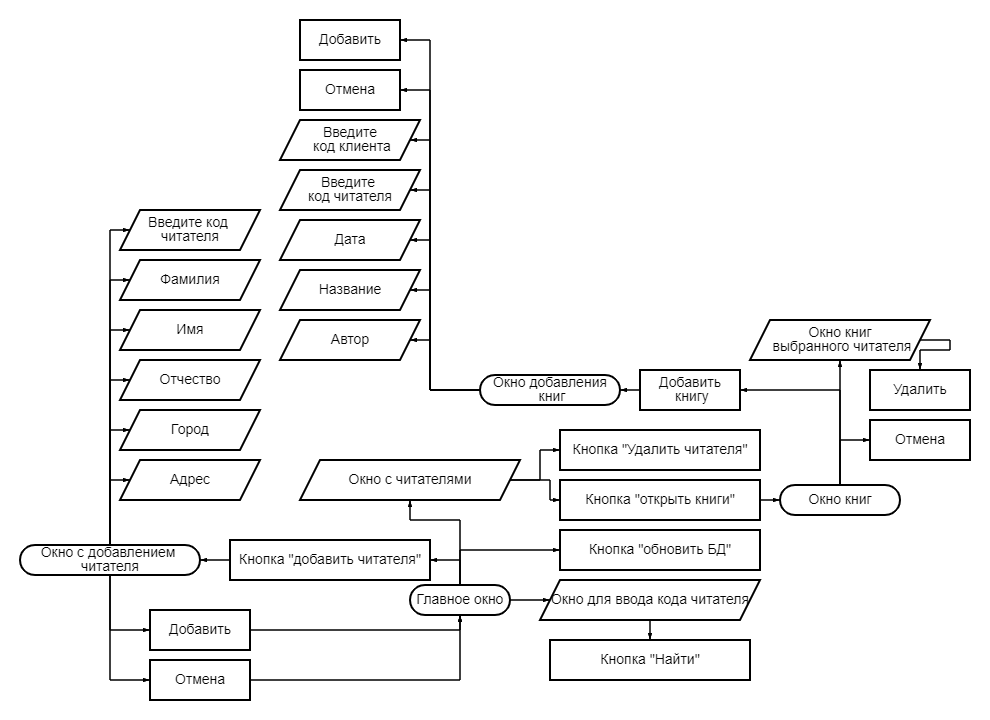


Рисунок 2 – Схема приложения

Также были созданы алгоритмы работы кнопок:

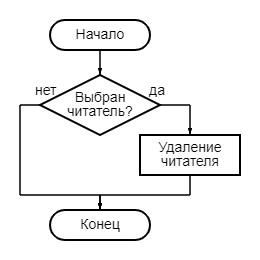


Рисунок 3 – Алгоритм кнопки «Удалить»

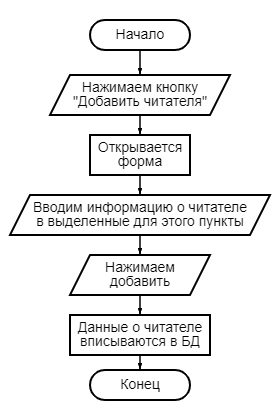


Рисунок 4 – Алгоритм добавления читателя.

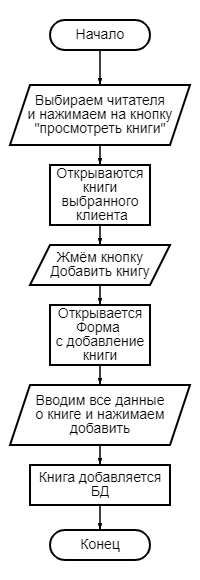


Рисунок 5 – Алгоритм добавления и просмотра книг

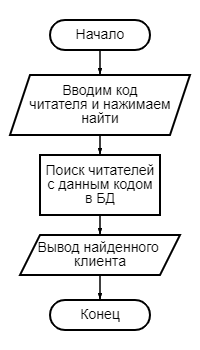


Рисунок 6 – Алгоритм поиска читателей.

# **5. Пример реализации программного продукта**

При запуске программы, выводится главное окно с читателями и редактированием их.

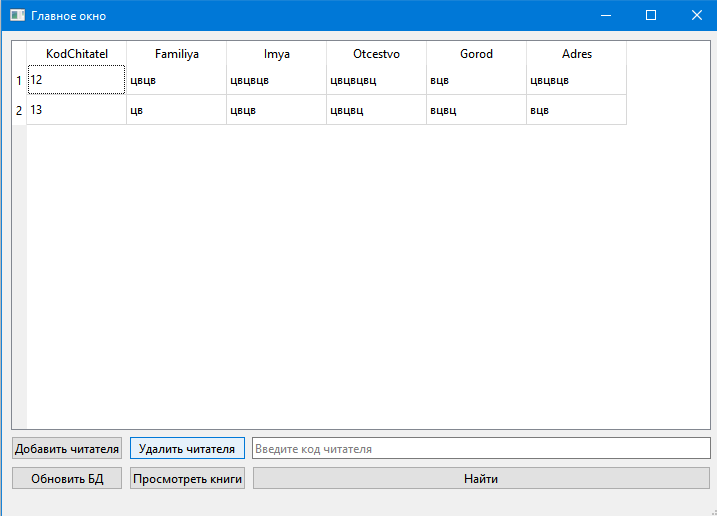


Рисунок 7 – Запуск программы

Далее при первом запуске стоит добавить двух новых читателей. Для этого нажимаем на кнопку «Добавить читателя» и откроется окно.

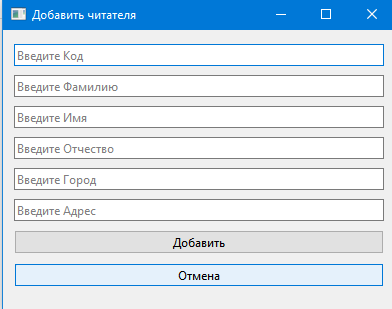


Рисунок 8 – Добавление клиента

Вводим нужные нам данные для читателя. После ввода данных нужно нажать кнопку «Добавить», если окно было открыть случайно можно нажать кнопку «Отмена» и вы будете возвращены в предыдущее окно с читателями.

После добавления клиентов можно каждому клиенту добавить несколько заказов. Находим нужного нам клиента с помощью кода клиента. Для этого вводим код клиента в окно над кнопкой «Найти» и жмём эту кнопку. Рисунок 11.

Выбираем клиента и нажимаем кнопку «Открыть заказы», после чего нам откроется окно с заказами данного клиента.

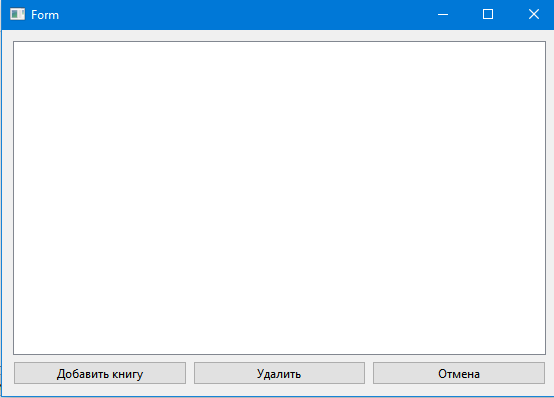


Рисунок 9 – Окно программы с заказами клиента

В данном окне будут отображаться заказы выбранного ранее читателя. В этом окне жмём «Добавить книгу». Выведется окно с вводом информации о заказе.

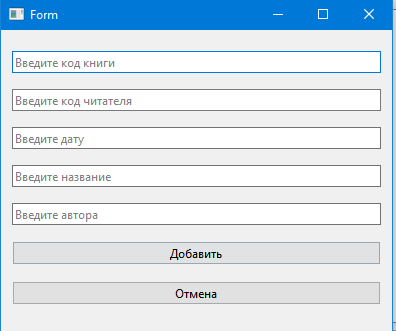


Рисунок 10 – Окно добавления книги

Кнопки удалить в Заказах и Клиентах работают одинаково:

Нужно выбрать нужную нам строчку и нажать кнопку удалить, после чего информация будет удалена.

Кнопки отмена работают так же одинаково, закрывая окно, в котором была нажата кнопка.

Информацию о товаре или клиентах можно редактировать двойным нажатием по нужной для нас ячейки.

Так как БД была создана с форматов sqlite. Её можно редактировать через SQLiteStudio.

# **6. Системные требования**

* 32-разрядный (х86) или 64-разрядный (х64) процессор с тактовой частотой 1 ГГц или выше.
* 1 ГБ (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ.
* 1 ГБ (для 32-разрядной системы) или 2 ГБ (64-разрядной системы) свободного места на жестком диске.
* Графическое устройство DirectX 9 с драйвером WDDM 1.0 или более поздней версии.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы были приобретены навыки работы с языком C++, а также средой разработки Qt. Было создано приложение в соответствии с поставленной задачей. В процессе написания программы были изучены классы Qt.

# **Список используемых источников**

1. Jasmin Blanchette, Mark Summerfield, перевод Андрей Киселёв «Разработка графического интерфейса с помощью библиотеки Qt3» книга распространяется на условиях Open Publication License, v1.0.

2. Герберт Шилдт, «Полный справочник по C++», 4-е издание, издательство Диалектика, Москва 2003г, 800стр.

3. Макс Шлее. Qt4. Профессиональное программирование на C++. —

СПб.: БХВ-Петербург, 2007. — 880 с.

4. М.Кузнецов. Самоучитель MYSQL5. : «БХВ-Петербург» - 2007 г.

5. Александр Чиртик, «Программирование на С++», издательство Питер, 2010 год, 352стр.

6. Юрий Земсков. Программирование на C++ с использованием библиотеки

Qt 4. г. Волгоград 2007 г.

# **Приложение. Листинг программы**

**DOBAVITCHIT.CPP**

#include "dobavitchit.h"

#include "ui\_dobavitchit.h"

QSqlDatabase db;

DobavitChit::**DobavitChit**(QWidget \*parent) :

QWidget(*parent*),

ui(new Ui::DobavitChit)

{

ui->setupUi(this);

}

DobavitChit::~***DobavitChit***()

{

delete ui;

}

void DobavitChit::**on\_pushButton\_clicked**()

{

QString KodChit = ui->lineEdit->text(); // присваиваются значения к переменным из окон

QString Familiya = ui->lineEdit\_2->text();

QString Imya = ui->lineEdit\_3->text();

QString Otcestvo = ui->lineEdit\_4->text();

QString Gorod = ui->lineEdit\_5->text();

QString Adres = ui->lineEdit\_6->text();

QString s = QString("INSERT INTO Chit VALUES('%1', '%2', '%3', '%4', '%5', '%6')").arg(KodChit).arg(Familiya).arg(Imya).arg(Otcestvo).arg(Gorod).arg(Adres);

QSqlQuery q (s,db); //q новая переменная, s выше,

DobavitChit::close(); //Закрываем форму

MainWindow\* f =new MainWindow; //присвваиваем к f \*ссылка на форму

f->show(); // Открываем другу форму f

}

void DobavitChit::**on\_pushButton\_2\_clicked**()

{

DobavitChit::close();

MainWindow\* f =new MainWindow;

f->show();

}

**DOBAVITKNIGU.CPP**

#include "dobavitknigu.h"

#include "ui\_dobavitknigu.h"

#include "mainwindow.h"

#include "dobavitchit.h"

#include "openchit.h"

DobavitKnigu::DobavitKnigu(QWidget \*parent,QSqlDatabase \*db) :

QWidget(parent),

ui(new Ui::DobavitKnigu)

{

ui->setupUi(this);

this->db = db;

}

DobavitKnigu::~DobavitKnigu()

{

delete ui;

}

void DobavitKnigu::on\_pushButton\_clicked()

{

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

QString KodChitatel = ui->lineEdit\_2->text();

QString Data = ui->lineEdit\_3->text();

QString Name = ui->lineEdit\_4->text();

QString Avtor = ui->lineEdit\_6->text();

// QString s = QString("INSERT INTO Kniga VALUES('%1', '%2', '%3', '%4', '%5')").arg(KodKnigi).arg(KodChitatel).arg(Data).arg(Name).arg(Avtor);

// QSqlQuery q (s,\*db);

DobavitKnigu::close();

}

void DobavitKnigu::on\_pushButton\_2\_clicked()

{

DobavitKnigu::close();

}

**MAINWINDOW.CPP**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

db = QSqlDatabase::addDatabase("QSQLITE");

db.setDatabaseName("KursDB.sqlite");

db.open();

db.exec("CREATE TABLE Chit (KodChitatel INTEGER PRIMARY KEY , Familiya TEXT, Imya TEXT, Otcestvo TEXT, Gorod TEXT, Adres TEXT);");

db.exec("CREATE TABLE Kniga (KodKnigi INTEGER, KodChitatel INTEGER, Data DATE, Name TEXT, Avtor TEXT);");

quvery = new QSqlQuery(db);

model = new QSqlTableModel(this, db);

model->setTable("Chit");

model->select();

ui->tableView->setModel(model);}

MainWindow::~MainWindow()

{

delete ui;}

void MainWindow::on\_pushButton\_clicked()

{

DobavitChit\* f =new DobavitChit;

f->show();

MainWindow::close();}

void MainWindow::on\_pushButton\_5\_clicked() //Обновление БД

{

model->setTable("Chit"); //Присвоение к переменной таблицы

model->select(); //Запрос

ui->tableView->setModel(model); // вывод}

void MainWindow::on\_tableView\_clicked(const QModelIndex &index) //Сохранение индекса

{

row=index.row();

Index=index;}

void MainWindow::on\_pushButton\_2\_clicked() //Удаление

{

model->removeRow(row);

model->setTable("Chit");

model->select();}

void MainWindow::on\_pushButton\_4\_clicked() //Найти

{

QSqlQueryModel \*query = new QSqlQueryModel();

QString s = QString("SELECT\* FROM Chit WHERE KodChitatel ='" + ui->lineEdit->text() + "'");

query->setQuery(s);

ui->tableView->setModel(query);}

void MainWindow::on\_pushButton\_3\_clicked()

{

QString KodChitatel = ((QSqlQueryModel\*)Index.model())->record(Index.row()).value("KodChitatel").toString();

OpenChit\* k = new OpenChit (nullptr, KodChitatel, &db);

// OpenChit (nullptr,KodChitatel,&db);

k->show();}

}

**OPENCHIT.CPP**

#include "openchit.h"

#include "ui\_openchit.h"

QString l;

OpenChit::**OpenChit**(QWidget \*parent,QString s, QSqlDatabase \*db) :

QWidget(*parent*),

ui(new Ui::OpenChit)

{

ui->setupUi(this);

}

OpenChit::~***OpenChit***()

{

delete ui;

}

void OpenChit::**on\_pushButton\_clicked**()

{

DobavitKnigu\* f =new DobavitKnigu(nullptr,*db*);

f->show();

}

void OpenChit::**on\_pushButton\_2\_clicked**()

{

db->exec("DELETE FROM Kniga WHERE KodKnigi = " + QString::number(row));

model->setQuery(QSqlQuery(l,\*db));

}

void OpenChit::**on\_tableView\_clicked**(const QModelIndex &index)

{

row = ((QSqlQueryModel\*)index.model())->record(index.row()).value("KodKnigi").toInt();

}

void OpenChit::**on\_pushButton\_4\_clicked**()

{

OpenChit::close();

}