МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гомельский государственный университет

имени Франциска Скорины»

Факультет математики и технологий программирования

Кафедра математических проблем управления и информатики

Разработка базы данных для библиотеки

Курсовая работа

Исполнитель

студент группы ПИ-31           \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.М. Обухов

Научный руководитель

старший преподаватель кафедры МПУИ \_\_\_\_\_\_\_\_\_ Л.И. Короткевич

Гомель 2022

**Реферат**

Курсовая работа 25 страниц, 15 таблиц, 2 рисунка, 2 приложения, 7 источников

**Ключевые слова:** база данных, язык программирования, приложение, визуализация данных, данные, средства разработки.

**Объект исследования:** Язык программирования С++, SQL, средства разработки.

**Предмет исследования:** возможности взаимодействия языков программирования при разработке приложений.

**Цель курсовой работы:** Изучение языков программирования C++ и SQL, их взаимодействие, и составление приложения для выполнения, а так же получение различных навыков при использовании этих языков программирования.

**Задачами курсовой работы являются:** Составление приложения, выполняющего подключение к базе данных, описывающее опись книг и выдача их читателям в библиотеке

**Выводы:** Созданные функциональности приложения позволяют упрощать работу при взаимодействии пользователя и кода.

**Содержание**

Введение 3

1. Назначение разработанного приложения 4

2 Описание средств разработки приложения 5

2.1 Среда разработки Visual Studio 2022 5

2.2 Система Управления Базами Данных Microsoft SQL Server 6

2.3 Среда управления SQL Server Management Studio 7

2.4 Язык программирования SQL 8

2.5 Язык программирования С++ 8

3 Структура базы данных разработанного приложения 10

3.1 Состав таблиц базы данных 10

3.2 Структура таблиц базы данных 11

4 Взаимодействие пользователя с разработанным приложением 15

Заключение 16

Список использованных источников 17

Приложение А Скрипт для создания структуры таблиц базы данных 18

Приложение Б Код C++, выполняющий подключение к СУБД, и выполняющий ряд заданных действий для взаимодействия с СУБД 23

# Введение

В библиотеках хранится великое число различного рода литературы, которую необходимо каким-либо оптимальным образом хранить, и знать, где её искать. Также библиотеку посещают читатели – они могут брать оттуда книги, при этом идет учет того, что они взяли.

Целью проекта является упрощение работы библиотеки, а именно: выдача книг клиентам(читателям), их хранение, каталогизирование, получение и списание; учет клиентов, сотрудников, помещений; контроль корректности данных.  
Данные возможности актуальны в нынешнее время, учитывая тот факт, что в библиотеке, архиве и читальных залах хранится множество едениц различного рода литературы, информацию о которых необходимо собрать воедино и упорядочить для последующего решения проблем по поиску, выдаче и учету книг, а так же книг, которые выдали читателю.

Задачей курсовой работы является создание приложения на базе языка C++ в интегрированной среде разработки Visual Studio, где работа идет с созданием GUI формы с подключением базы данных на языке SQL в MS SQL Server для хранения информации (см. приложение А).

Для достижения вышеописанных целей необходимо решить следующие задачи:

* проанализировать требования;
* реализовать серверное подключение с базой данных;
* разработать клиентскую часть, которая будет интерфейсом пользователя, или сотрудника;
* сделать приложение простым и доступным в использовании, а также настроить ограничения на доступ к данным;
* протестировать разработанное приложение.

Разработка визуальной части приложения будет вестись на MS Visual Studio 2022. Для хранения и получения данных будет подключена СУБД на MS SQL Server

# Назначение разработанного приложения

Приложение представляет собой возможность быстрого и удобного взаимодействия с хранимыми данными. Создано для сотрудников библиотеки с целью ведения учета хранимых книг, их поставок и списаний, для быстрого доступа к необходимым книгам, а также хранения информации о выдаче книг читателям с целью несения ответственности читателем за полученную им литературу, принадлежащую учреждению (библиотеке), выдавшего книгу.

Подробная информация о книгах, выдачах и читателях станет доступна пользователю, как только он выберет соответствующий раздел.

Подробная информация, которая будет упомянута выше, будет включать:

* личные данные читателя;
* личные данные сотрудника(-ков);
* полная информация о книгах: их название , жанр, издательство, инвертарный номер и т.п.;
* выдача книг читателям.

Примечание: информация будет доступна только пользователям, имеющим необходимый уровень доступа.

# Описание средств разработки приложения

## **Среда разработки Visual Studio 2022**

Microsoft Visual Studio  - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

## **Система Управления Базами Данных Microsoft SQL Server**

Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Преимущества СУБД:

* создание клиент-серверных приложений (Многопользовательская работа (совместное обновление данных), уменьшение сетевого трафика);
* создание программ, не чувствительных к изменению структуры данных;
* Наличие в СУБД стандартных, качественно реализованных, алгоритмов обработки данных (сортировка, поиск и др.);
* Автоматическое поддержание целостности данных (автоматическое резервное копирование, контроль ссылочной целостности, поддержка механизма транзакций с возможным откатом транзакций);
* разграничение доступа к данным;
* возможность получения данных по запросам без разработки специальных программ.

Центральным аспектом MS SQL является база данных, которая представляет из себя хранилище данных, организованных определенным способом. Для организации баз данных MS SQL использует реляционную модель.

Реляционная модель – подход к управлению данными с использованием языка и структуры, согласованных с логикой первого порядка, где все данные представлены в виде кортежей, сгруппированных в отношения.

Реляционная модель базы данных:

* каждому типу объекта базы данных соответствует таблица. Строка — экземпляр объекта, столбец – атрибут (поле);
* у столбца уникальное имя в таблице. Возможны одинаковые имена столбцов, расположенных в разных таблицах;
* в таблице может содержаться любое количество строк (возможно, нулевое);
* строки таблицы не имеют определенного порядка. Нельзя выбрать строку по номеру в таблице. Нет “первой”, “последней” и т.д. строк;
* в каждой таблице есть один или несколько столбцов, значения или совокупность значений, которых во всех строках различаются. Этот столбец (столбцы) называется первичным ключом таблицы, и выбор конкретной записи таблицы можно осуществить по значению её первичного ключа;
* столбец (совокупность столбцов) одной таблицы, значения в котором совпадают со значениями столбца, являющегося первичным ключом другой таблицы, называется внешним столбцом. Совместно первичный и внешний ключи создают между таблицами отношения предок/потомок (родительская и дочерняя таблицы).

## **Среда управления SQL Server Management Studio**

SQL Server Management server – интегрированная среда для управления любой инфраструктурой SQL. Используется для настройки, доступа, администрирования, разработки всех компонентов SQL. Является стандартным инструментом для работы с MS SQL. Предоставляет удобный интерфейс пользователю для настройки, мониторинга, администрирования экземпляров SQL server и БД. Также предоставляет возможность создавать запросы и скрипты.

Компоненты среды:

* обозреватель объектов – нужен для просмотра всех объектов и управления ими;
* обозреватель шаблонов – нужен для создания файлов со стандартным текстом, которые используются для ускорения разработки скриптов, запросов, и управления ими;
* средства визуального проектирования – нужны для создания запросов, таблиц и схем баз данных;
* редакторы запросов и текста – нужны для интерактивного написания и откладки скриптов, запросов.

## **Язык программирования SQL**

SQL (Structured Query Language - «язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. Является, прежде всего, информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. В общем случае SQL (без ряда современных расширений) считается языком программирования неполным по Тьюрингу, но вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений.

На языке SQL описываются наборы данных, помогающие получать ответы на вопросы. При использовании SQL необходимо применять правильный синтаксис. Синтаксис - это набор правил, позволяющих правильно сочетать элементы языка. Синтаксис SQL основан на синтаксисе английского языка и имеет много общих элементов с синтаксисом языка Visual Basic для приложений (VBA).

## **Язык программирования С++**

**C++** - компилируемый, статически типизированный язык программирования общего назначения.

Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков

Своими корнями он уходит в язык Си, который был разработан в 1969-1973 годах в компании Bell Labs программистом Деннисом Ритчи (Dennis Ritchie). В начале 1980-х годов датский программист Бьерн Страуструп (Bjarne Stroustrup), который в то время работал в компании Bell Labs, разработал С++ как расширение к языку Си. Фактически вначале C++ просто дополнял язык Си некоторыми возможностями объектно-ориентированного программирования. И поэтому сам Страуструп вначале называл его как "C with classes" ("Си с классами").

С++ является мощным языком, унаследовав от Си богатые возможности по работе с памятью. Поэтому нередко С++ находит свое применение в системном программировании, в частности, при создании операционных систем, драйверов, различных утилит, антивирусов и т.д. К слову сказать, ОС Windows большей частью написана на С++. Но только системным программированием применение данного языка не ограничивается. С++ можно использовать в программах любого уровня, где важны скорость работы и производительность. Нередко он применяется для создания графических приложений, различных прикладных программ. Также особенно часто его используют для создания игр с богатой насыщенной визуализацией. Кроме того, в последнее время набирает ход мобильное направление, где С++ тоже нашел свое применение. И даже в веб-разработке также можно использовать С++ для создания веб-приложений или каких-то вспомогательных сервисов, которые обслуживают веб-приложения. В общем С++ - язык широкого пользования, на котором можно создавать практически любые виды программ.

С++ является компилируемым языком, а это значит, что компилятор транслирует исходный код на С++ в исполняемый файл, который содержит набор машинных инструкций. Но разные платформы имеют свои особенности, поэтому скомпилированные программы нельзя просто перенести с одной платформы на другую и там уже запустить. Однако на уровне исходного кода программы на С++ по большей степени обладают переносимостью, если не используются какие-то специфичные для текущей ос функции. А наличие компиляторов, библиотек и инструментов разработки почти под все распространенные платформы позволяет компилировать один и тот же исходный код на С++ в приложения под эти платформы.

В отличие от Си язык C++ позволяет писать приложения в объектно-ориентированном стиле, представляя программу как совокупность взаимодействующих между собой классов и объектов. Что упрощает создание крупных приложений.

# Структура базы данных разработанного приложения

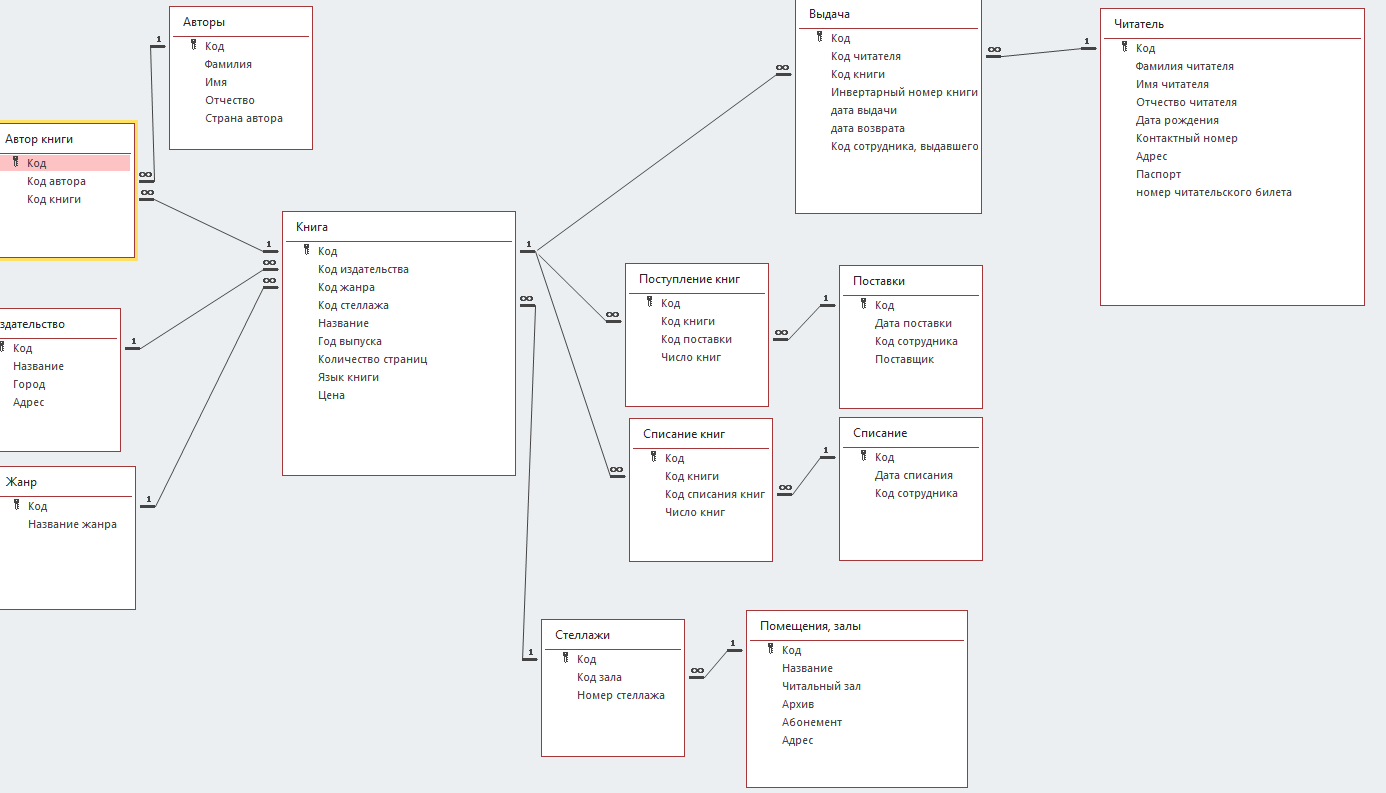
## **Состав таблиц базы данных**

Состав таблиц для базы данных библиотеки представлен в таблице 1.

**Таблица 1 – Состав таблиц базы данных**

|  |  |
| --- | --- |
| Наименование | Общие сведения о содержимом |
| 1. Издательство | издательство, выпустившее книгу |
| 1. Жанр | жанр книги |
| 1. Авторы | список всех авторов с информацией о них |
| 1. Автор книги | служит для связи между таблицами «Авторы», «Книга» |
| 1. Книга | содержит доступную информацию о книге |
| 1. Поставки | информация о поставках книг в библиотеку |
| 1. Списание | информация о списании книг |
| 1. Поступление книг | служит для связи между таблицами «Поставки», «Книга» |
| 1. Списание книг | служит для связи между таблицами «Списание», «Книга» |
| 1. Выдача | информация о выданной литературе читателю |
| 1. Стеллажи | описывает расположение книг на стеллажах |
| 1. Помещения | помещения библиотеки |
| 1. Сотрудники | сотрудники, работающие в библиотеке |
| 1. Читатель | информация о читателях библиотеки |

Взаимосвязь между таблицами указана на рисунке 2.



**Рисунок 1 – Схема базы данных**

## **Структура таблиц базы данных**

В таблицах 2-15 представлена структура таблиц разработанной базы данных.

**Таблица 2 – Издательство**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | код издательства |
| Название | varchar(50) | Название издательства |
| Город | varchar(40) | Город издательства |
| Адрес | varchar(80) | Адрес издательства |

**Таблица 3 – Жанр**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Код жанра |
| Название жанра | varchar(40) | Название жанра книги |

**Таблица 4 – Автор книги**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Код автора книги |
| Код книги | int | Код книги |
| Код автора | int | Код автора, написавшего книгу |

**Таблица 5 – Книга**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Код книги |
| Код издательства | int | Код издательства |
| Код жанра | int | Код жанра |
| Код стеллажа | int | Код стеллажа |
| Название | varchar(50) | Название книги |
| Год выпуска | int | Год выпуска книги |
| Число страниц | int | Число страниц в книге |
| Язык книги | varchar(20) | Язык, на котором издана книга |
| Цена | money | Цена книги при её закупке (в формате BUN) |

**Таблица 6 – Авторы**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код Автора |
| Фамилия | varchar(40) | Фамилия автора |
| Имя | varchar(40) | Имя автора |
| Отчество | varchar(40) | Отчество автора |
| Страна автора | varchar(40) | Страна, где родился автор |

**Таблица 7 – Поступление книг**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код поступления книг |
| Код книги | int | Код книги |
| Код поставки | int | Код поставки |
| Число книг | int | Число книг в партии |

**Таблица 8 – Поставки**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код Поставки |
| Дата поставки | date | Дата, когда были поставлены книги |
| Код сотрудника | int | Код сотрудника, ответственного за получение партии книг |
| Поставщик | varchar(40) | Код поставщика |

**Таблица 9 – Списание книг**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код списания книг |
| Код книги | int | Код книги |

**Окончание таблицы 9**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код списания книг | int | Код списания книг |
| Число книг | int | Число книг, которые списаны за определенный период |

**Таблица 10 – Списание**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код Списания |
| Дата списания | date | Дата списания книг |
| Код сотрудника | int | Код сотрудника, ответственного за списание книг |

**Таблица 11 – Выдача**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код выдачи |
| Код читателя | int | Код читателя |
| Код книги | int | Код книги |
| Инвентарный номер | varchar(40) | Инвертарный номер книги |
| Дата выдачи | date | Дата выдачи книги читателю |
| Дата возврата | date | Дата возврата книги читателю |
| Код сотрудника, выдавшего книгу | int | Код сотрудника, ответственного за процесс выдачи книги |

**Таблица 12 – Стеллажи**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код стеллажа |
| Код зала | int | Код зала, где находится стеллаж |
| Номер стеллажа | varchar(40) | Номер стеллажа (для идентификации) |

**Таблица 13 - Помещения**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код помещения |
| Название | varchar(40) | Название помещения |
| Читальный зал | bit | Поле для отметки, чем является данное помещение |
| Архив | bit | Поле для отметки, чем является данное помещение |
| Абонемент | bit | Поле для отметки, чем является данное помещение |
| Адрес помещения | varchar(40) | Адрес помещения |

**Таблица 14 – Читатель**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код читателя |
| Фамилия | varchar(40) | Фамилия читателя |
| Имя | varchar(40) | Имя читателя |
| Отчество | varchar(40) | Отчество читателя |
| Дата рождения | date | Дата рождения читателя |
| Контактный номер | varchar(40) | Номер телефона для связи с читателем |
| Адрес проживания | varchar(40) | Адрес, где живет читатель |
| Данные паспорта | char(60) | Идентификационный номер паспорта |
| Номер читательского билета | varchar(40) | Номер читательского билета |

**Таблица 15 - Сотрудники**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Код | int | Уникальный код сотрудника |
| Код помещения | int | Код помещения, где работает сотрудник |
| Фамилия | varchar(40) | Фамилия сотрудника |
| Имя | varchar(40) | Имя сотрудника |
| Отчество | varchar(40) | Отчество сотрудника |

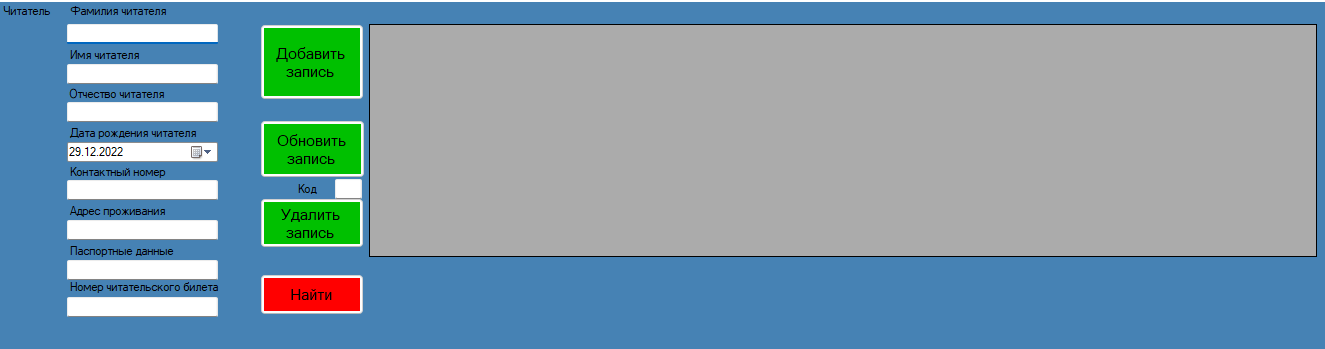
**Окончание таблицы 15**

|  |  |  |
| --- | --- | --- |
| Наименование поля | Формат поля | Содержимое поля |
| Занимаемая должность | varchar(40) | Занимаемая сотрудником должность |
| Стаж | int | Стаж сотрудника (в годах) |

# Взаимодействие пользователя с разработанным приложением

Для управления приложением с подключенной базой данных пользователю предоставляется окно GUI, на котором расположены различные элемента для отображения информации и управлением ею: текстовые поля, поля для ввода текста, выбор даты, списки (в т.ч. выпадающие), и кнопки

Пример работы приложения представлен на рисунке 2



**Рисунок 2 – Рабочая область пользовательского интерфейса для взаимодействия с пользователем**

# Заключение

Для упрощения работы с людьми и ресурсами библиотеки было создано приложение на языке С++, с подключением СУБД, в которой содержатся различные данные касательно структуры библиотеки, её содержимого, а также посещающих её клиентов (читателей). Пользователям с определёнными правами, таких как сотрудники, разрешено не только просматривать содержимое БД, но и вносить изменения, добавлять, удалять, редактировать информацию в БД.

Для успешного выполнения данной работы были изучены навыки работы с созданием оконных приложений в среде Visual Studio, язык С++, а также реляционные базы данных, в том числе язык запросов SQL.

Для достижения вышеописанных целей были выполнены следующие шаги:

* проанализировать требования;
* реализовать серверное подключение с базой данных;
* разработать клиентскую часть, которая будет интерфейсом пользователя, или сотрудника;
* сделать приложение простым и доступным в использовании, а также настроить ограничения на доступ к данным;
* протестировать разработанное приложение.

# Список использованных источников

1. Форум программистов [Электронный ресурс] // URL: https://stackoverflow.com
2. Форум программистов [Электронный ресурс] // URL: https://www.cyberforum.ru
3. Пархомов, Б. C/C++ и MS Visual C++ 2010 для начинающих/ Б. Пархомов – BHV.: 2011. – 729 с.
4. Сидорина, Т. Самоучитель Microsoft Visual Studio C++ и MFC. — СПб.: БХВ-Петербург, 2009. — 848 с.
5. Официальная документация по SQL от компании Microsoft https://learn.microsoft.com/ru-ru/sql
6. Форум программистов [Электронный ресурс] // URL: https://stackoverflow.com
7. Платформа видеостриминга YouTube [Электронный ресурс] // URL: https://www.youtube.com

# Приложение А

# Скрипт для создания структуры таблиц базы данных

alter table Книга

drop constraint FK\_Книга\_Издательство,

constraint FK\_Книга\_Жанр,

constraint FK\_Книга\_Стеллаж

GO

alter table [Автор книги]

drop constraint FK\_Книга\_Автор\_книги,

constraint FK\_Автор\_книги\_Авторы

GO

alter table [Поступление книг]

drop constraint FK\_Книга\_Поступления\_книг,

constraint FK\_Поступление\_книг\_Поставки

GO

alter table [Списание книг]

drop constraint FK\_Книга\_Списание\_книг,

constraint FK\_Списание\_книг\_Списание

GO

alter table [Выдача]

drop constraint FK\_Книга\_Выдача,

constraint FK\_Выдача\_Сотрудники,

constraint FK\_Выдача\_Читатель

GO

alter table [Стеллажи]

drop constraint FK\_Стеллажи\_Помещения

GO

alter table [Сотрудники]

drop constraint FK\_Сотрудники\_Помещения

go

alter table [Поставки]

drop constraint FK\_Поставки\_Сотрудники

go

alter table [Списание]

drop constraint FK\_Списание\_Сотрудники

go

drop table Издательство

go

drop table Жанр

go

drop table Книга

go

drop table [Автор книги]

go

drop table Авторы

go

drop table [Поступление книг]

go

drop table Поставки

go

drop table [Списание книг]

go

drop table Списание

go

drop table Выдача

go

drop table Стеллажи

go

drop table Читатель

go

drop table [Помещения]

go

drop table Сотрудники

go

Create table Издательство

(Код int not null primary key identity,

Название varchar(50),

Город varchar(40),

Адрес varchar(80) not null

)

go

create table Жанр

(Код int primary key not null identity,

[Название жанра] varchar(40) not null

)

GO

create table Книга

(Код int primary key not null identity,

[Код издательства] int,

[Код жанра] int,

[Код стеллажа] int,

[Название] varchar(50),

[Год выпуска] int,

[Число страниц] int,

[Язык книги] varchar(20) not null,

[Цена] money

)

GO

create table [Автор книги]

(Код int primary key not null identity,

[Код книги] int,

[Код автора] int

)

GO

create table Авторы

(Код int primary key not null identity,

[Фамилия] varchar(40),

[Имя] varchar(40),

[Отчество] varchar(40),

[Страна автора] varchar(40)

)

GO

create table [Поступление книг]

(Код int primary key not null identity,

[Код книги] int,

[Код поставки] Int,

[Число книг] int

)

GO

create table Поставки

(Код int primary key not null identity,

[Дата поставки] date,

[Код сотрудника] int,

[Поставщик] varchar(40)

)

GO

create table [Списание книг]

(Код int primary key not null identity,

[Код книги] int,

[Код списания книг] int,

[Число книг] int

)

GO

create table Списание

(Код int primary key not null identity,

[Дата списания] date,

[Код сотрудника] int

)

GO

create table Выдача

(Код int primary key not null identity,

[Код читателя] int,

[Код книги] int,

[Инвертарный номер] varchar(40),

[Дата выдачи] date not null,

[Дата возврата] date,

[Код сотрудника, выдавшего книгу] int

)

GO

create table Стеллажи

(Код int primary key not null identity,

[Код зала] int not null,

[Номер стеллажа] varchar(4)

)

GO

create table Помещения

(Код int primary key not null identity,

[Название] varchar(40),

[Читальный зал] bit,

[Архив] bit,

[Абонемент] bit,

[Адрес помещения] varchar(40)

)

GO

create table Читатель

(Код int primary key not null identity,

[Фамилия] varchar(40),

[Имя] varchar(40),

[Отчество] varchar(40),

[Дата рождения] date,

[Контактный номер] varchar(40),

[Адрес проживания] varchar(40),

[Данные паспорта] char(60),

[Номер читательского билета] varchar(40)

)

GO

create table Сотрудники

(Код int primary key not null identity,

[Код помещения] int,

[Фамилия] varchar(40),

[Имя] varchar(40),

[Отчество] varchar(40),

[Занимаемая должность] varchar(40),

[Стаж] int

)

GO

alter table Книга

add constraint FK\_Книга\_Издательство foreign key ([Код издательства]) references Издательство(Код),

constraint FK\_Книга\_Жанр foreign key ([Код жанра]) references Жанр(Код),

constraint FK\_Книга\_Стеллаж foreign key ([Код стеллажа]) references Стеллажи(Код)

GO

alter table [Автор книги]

add constraint FK\_Книга\_Автор\_книги foreign key ([Код книги]) references Книга(Код),

constraint FK\_Автор\_книги\_Авторы foreign key ([Код автора]) references Авторы(Код)

GO

alter table [Поступление книг]

add constraint FK\_Книга\_Поступления\_книг foreign key ([Код книги]) references Книга(Код),

constraint FK\_Поступление\_книг\_Поставки foreign key ([Код поставки]) references Авторы(Код)

GO

alter table [Списание книг]

add constraint FK\_Книга\_Списание\_книг foreign key ([Код книги]) references Поставки(Код),

constraint FK\_Списание\_книг\_Списание foreign key ([Код списания книг]) references Списание(Код)

GO

alter table [Выдача]

add constraint FK\_Книга\_Выдача foreign key ([Код книги]) references Книга(Код),

constraint FK\_Выдача\_Сотрудники foreign key ([Код сотрудника, выдавшего книгу]) references Сотрудники(Код),

constraint FK\_Выдача\_Читатель foreign key ([Код читателя]) references Читатель(Код)

GO

alter table [Стеллажи]

add constraint FK\_Стеллажи\_Помещения foreign key ([Код зала]) references Помещения(Код)

GO

alter table [Сотрудники]

add constraint FK\_Сотрудники\_Помещения foreign key ([Код помещения]) references Помещения(Код)

GO

alter table [Поставки]

add constraint FK\_Поставки\_Сотрудники foreign key ([Код сотрудника]) references Сотрудники(Код)

GO

alter table [Списание]

add constraint FK\_Списание\_Сотрудники foreign key ([Код сотрудника]) references Сотрудники(Код)

GO

insert into Жанр

values

('Практическое пособие')

insert into Издательство

values

('ГГУ имени Ф. Скорины','Гомель','Советская 104')

insert into Читатель

values

('Обухов','Никита','Михайлович','2002-06-21','+37529#######','Свиридова ##','HB#######','number\_bilet')

insert into Авторы

values

('Белокурский','Максим','Сергеевич','Беларусь'),

('Рябченко','Наталия','Валерьевна','Беларусь'),

('Атвиновский','Александр','Алексеевич','Беларусь')

insert into Помещения(Название,[Адрес помещения])

values

('1й читальный зал при ГГУ им. Ф. Скорины','ул. Советская 102'),

('2й читальный зал при ГГУ им. Ф. Скорины','ул. Советская 102')

insert into Стеллажи([Код зала],[Номер стеллажа])

values

(1,1)

insert into Книга

values

(1,1,1,'Дифференциальные уравнения', 2022, 47, 'русский', 30)

insert into [Автор книги]([Код книги],[Код автора])

values

(1,1),

(1,2),

(1,3)

insert into Сотрудники

values

(1,'Иванов','Петр','Ильич','библиотекарь',3),

(1,'Семченко','Алла','Сергеевна','уборщица',2)

insert into [Поступление книг]([Код книги],[Код поставки],[Число книг])

values

(1,1,10)

insert into [Поставки]

values

('2022-05-07',1,'ГГУ им. Ф. Скорины')

# Приложение Б

# Код C++, выполняющий подключение к СУБД, и выполняющий ряд заданных действий для взаимодействия с СУБД

public:

SqlConnection^ conn;

SqlConnectionStringBuilder^ connection;

void DoConnect() {

connection = gcnew SqlConnectionStringBuilder();

connection->DataSource = "NIKITPC";

connection->InitialCatalog = "Библиотека";

connection->IntegratedSecurity = "True";

conn = gcnew SqlConnection(Convert::ToString(connection));

}

String^ SendCommand(String^ squery) {

SqlCommand^ query = gcnew SqlCommand(squery, conn);

conn->Open();

String^ index = (query->ExecuteNonQuery()).ToString();

if (conn != nullptr)

conn->Close();

return index;

}

int^ SendCommandRes(String^ squery) {

SqlCommand^ query = gcnew SqlCommand(squery, conn);

conn->Open();

int^ index=0;

try {

Object^ o = query->ExecuteScalar();

index = Convert::ToInt32(o);

}

catch (Exception^ e) {};

if (conn != nullptr)

conn->Close();

return index;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

try {

DoConnect();

String^ surname = textBox1->Text;

String^ name = textBox2->Text;

String^ patronymic = textBox3->Text;

String^ bdate = dateTimePicker1->Text;

String^ num = textBox4->Text;

String^ addr = textBox5->Text;

String^ passportID = textBox6->Text;

String^ readersTicketID = textBox7->Text;

String^ squery =

"insert into Читатель([Фамилия],[Имя],[Отчество],[Дата рождения],[Контактный номер],[Адрес проживания],[Данные паспорта],[Номер читательского билета])"

"values('"+surname+"','"+name+"','"+patronymic+"','"+bdate+"','"+num+"','"+addr+"','"+passportID+"','"+readersTicketID+"')";

String^ changed = SendCommand(squery);

MessageBox::Show(changed);

}

catch (Exception^ e) {

MessageBox::Show(e->ToString(), "error", MessageBoxButtons::OK);

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

try {

DoConnect();

String^ surname = textBox1->Text;

String^ name = textBox2->Text;

String^ patronymic = textBox3->Text;

String^ bdate = dateTimePicker1->Text;

String^ num = textBox4->Text;

String^ addr = textBox5->Text;

String^ passportID = textBox6->Text;

String^ readersTicketID = textBox7->Text;

String^ squery =

"update Читатель set Фамилия= '"+surname+"',Имя='"+name+"',Отчество='"+patronymic+"',[Дата рождения]='"+bdate+"',[Контактный номер]='"+num+"',[Адрес проживания]='"+addr+"',[Данные паспорта]='"+passportID+"',[Номер читательского билета]='"+readersTicketID+"'"

"where Код = '"+Convert::ToInt32(textBox8->Text)+"'" ;

String^ changed = SendCommand(squery);

MessageBox::Show(changed);

}

catch (Exception^ e) {

MessageBox::Show(e->ToString(), "error", MessageBoxButtons::OK);

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

try {

DoConnect();

int^ getCode = Convert::ToInt32(textBox8->Text);

int^ checkIF = SendCommandRes("select Код from [Читатель] where Код = '" + getCode + "' ");

boolean f = false;

if ((int)checkIF>0) { f = true; }

if (f) {

String^ squery = "delete Читатель where Код = '" + Convert::ToInt32(textBox8->Text) + "'";

String^ changed = SendCommand(squery);

MessageBox::Show(changed);

} else{

MessageBox::Show("no data", "missing data",MessageBoxButtons::RetryCancel);

}

}

catch (Exception^ e) {

MessageBox::Show(e->ToString(), "error", MessageBoxButtons::OK);

}

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

try {

DoConnect();

conn->Open();

SqlCommand^ comms = gcnew SqlCommand("select \* from Читатель", conn);

SqlDataAdapter^ da = gcnew SqlDataAdapter(comms);

DataTable^ dt = gcnew DataTable();

da->Fill(dt);

dataGridView1->DataSource = dt;

}

catch (Exception^ e) {};}

