Министерство Образования Кыргызской Республики

«Кыргызский Государственный Технический Университет им. И. Раззакова»

Факультет «Информационных технологий»

Кафедра «Программное обеспечение компьютерных систем»

Утверждаю

Заведующий кафедрой ПОКС,

Д.ф.-м.н., проф. КГТУ \_\_\_\_\_\_\_\_\_\_\_Салиев А.Б.

“\_\_\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_\_ г.

**Отчет**

на тему:

«**Информационная система выдачи книг и читательского спроса**»

**РУКОВОДИТЕЛЬ**:

Турсалиева Э.Н.

**СТУДЕНТ:** гр. Пи-3-18

Аскербеков А.Н.

Оглавление

[**1. Введение** 3](#_Toc44666297)

[**2. Основание для разработки** 3](#_Toc44666298)

[**2.2. Наименование работы:** 3](#_Toc44666299)

[**3. Назначение** 3](#_Toc44666300)

[**3.1. Цели создания системы:** 3](#_Toc44666301)

[**3.2. Задачи** 3](#_Toc44666302)

[4. Спецификация проблемы. 4](#_Toc44666303)

[**5.Технические требования** 4](#_Toc44666304)

[**5.1.** **Требования к функциональным характеристикам.** 4](#_Toc44666305)

[**5.1.3. Организация входных и выходных данных** 5](#_Toc44666306)

[**5.2. Требования к надежности** 5](#_Toc44666307)

[**5.3. Требования к составу и параметрам технических средств.** 5](#_Toc44666308)

[5.5. Требования по безопасности 5](#_Toc44666309)

[**6. Требования к программной документации** 6](#_Toc44666310)

[**6.1. Входные** 6](#_Toc44666311)

[**6.2** **Выходные** 6](#_Toc44666312)

[**7.База данных программы** 7](#_Toc44666313)

[**8.Интерфейс программы:** 12](#_Toc44666314)

[**Заключение:** 30](#_Toc44666315)

[**Список литератур:** 31](#_Toc44666316)

# **1. Введение**

Работа выполняется в рамках проекта «Информационная система выдачи книг и читательского спроса» в библиотеке хранятся разные книги на разные темы по различным областям знаний, относящиеся к разным жанрам. Книги написаны разными авторами и изданы разными издательствами. Каждая книга хранится в библиотеке в нескольких экземплярах. Читатели берут определенные экземпляры книг на заранее определенный срок. Библиотекарь отслеживает закупку книг, их выдачу и возврат книг читателями, а также ведет учет читательского спроса

# **2. Основание для разработки**

2.1. Программа разрабатывается на основе учебного плана кафедры «Программное обеспечение компьютерной системы».

## **2.2. Наименование работы:**

«Информационная система выдачи книг и читательского спроса»

2.3. Исполнитель: Студент группы ПИ-3-18 Аскербеков Айдин Нурланбекович

2.4. Соисполнители: нет.

# **3. Назначение**

Разрабатываемая система предназначена для учета поступления книг и использования ресурсов (книг).

## **3.1. Цели создания системы:**

Информационная система для выдачи книг и читательского спроса.

## **3.2. Задачи**

Выполнять следующие бизнес-процессы:

Оформлять книги, поступившие в библиотеку.

Создавать и оформлять формуляры читателей (читательский билет)

Выдавать книги читателям

Вести учет выдачи книг и читательского спроса

Вести базу данных:

Осуществлять поиск данных по запросам пользователя

Проводить анализ читателей должников и рейтинга авторов и их книг.

Принимать решение на основе анализа читательского спроса

Формировать выходные документы

Получать информацию о:

Название, цена, число страниц, год издания

Название темы

Название жанра

ФИО, адрес, телефон автора

Название, адрес, телефон издательства

Номер экземпляра

ФИО, адрес, телефон читателя

1. Спецификация проблемы.
2. Отсутствует база данных книг;
3. Все документы заполняются вручную;
4. Многие сотрудники забывают о сроке возврата книг от читателей
5. Отсутствие хранение архивов документов и редких экземпляров книг.
6. Библиотеки испытывают большие трудности с комплектованием. То-есть новые книги поступают не систематически, в ограниченном ассортименте и в небольшом количестве, образовательные учреждения не в состоянии заказывать именно ту литературу, которая им необходима.

**5.Технические требования**

* 1. **Требования к функциональным характеристикам.**
* авторизация пользователей;
* распределение прав доступа пользователя;
* добавление номенклатуры;
* редактирование, изменение, удаление существующий данных номенклатуры;
* хранить данные о читателе, о сотруднике, о поставщике, книг и выдача книг;
* сортировка данных (по жанром, по автору, название книг, ФИО читателя и.д.р);
* Осуществлять поиск данных по запросам;
* Уведомление о сроке возврате книг;
* Выводить чек поставщику;
* Формировать выходные документы;
* архивировать старые данные;
* выводить отчеты о выдачи книг, о расчетах;

## **5.1.3. Организация входных и выходных данных**

Входные данные: закупаемая книга, паспорт читателя, приходная накладка

Выходные данные: выданные книги, формуляр читателя, чек от закупаемой книги

## **5.2. Требования к надежности**

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска исполняемого файла системы;

– при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;

## **5.3. Требования к составу и параметрам технических средств.**

Система должна работать на IBM-совместимых персональных компьютерах.

Минимальная конфигурация:

• тип процессора. core i3 и выше;

• объем оперативного запоминающего устройства 32 Мб и более;

• объем свободного места на жестком диске 40 Мб.

Рекомендуемая конфигурация:

• тип процессора. core i5;

• объем оперативного запоминающего устройства 128 Мб;

• объем свободного места на жестком

диске 60 Мб.

• ПК с наличием usb-входов, принтер.

5.5. Требования по безопасности

Система должна обеспечивать надежную безопасность данных, основываясь на стандарт ГОСТ 17799, такую как:

* Авторизация пользователей и контроль доступа;
* Защита обмена данными.

# **6. Требования к программной документации**

## **6.1. Входные**

6.1.1. Приходная накладная

6.1.2. Счет-фактура

6.1.3. Паспорт читателя

* + 1. Закупаемая книга
  1. **Выходные** 
     1. Выданная книга
     2. Формуляр читателя
     3. Отчеты о деятельности библиотеки
     4. Журнал учета

**7.База данных программы**

Цель работы - построить базу данных (БД) по проекту “ Информационная система выдачи книг и читательского спроса ”. Объектом исследования является библиотека, а предметом исследования – книги. Необходимо разработать базу данных, которая была бы максимально гибкой и удобной в использовании. База данных, разрабатываемая в данном проекте, предназначена эффективно осуществлять хранение, поиск и выдачу информации о книгах, темах, авторах, читателях, издательствах, экземплярах, жанрах. Обеспечить выдачу информации о взятых книгах и о читателях должниках. В качестве инструмента построения базы данных использован SQL Server. В данной работе при создании БД были использованы следующие таблицы, представлены они на Рис.1.

|  |  |
| --- | --- |
| **Наименование таблицы** | **Первичный ключ** |
|
| Выдача\_книги1 | Код\_выдачи\_книги |
| Книги | Код\_книги |
| Накладная | Код\_накладной |
| Поставщик | Код\_поставщика |
| Сотрудники | Код\_сотрудника |
| Читатель | Код\_читателя |

Рис.1. Таблица сущностей

**Описание таблиц.**

Таблица «Выдачи\_книги1» (Рис.2.) состоит из десяти полей: «Код\_выдачи\_книги» (тип данный bigint) , «Код\_читателя» (тип данный bigint), «ФИО\_читателя» (тип данный varchar(50)), «Код\_книги» (тип данный bigint), «Дата\_выдачи\_книги»( тип данный date), «Дата\_возврата\_книги» (тип данный date), «ФИО\_сотрудника» (тип данный varchar(50)), «Код\_сотрудника» (тип данный bigint), «Количество\_книг» (тип данный bigint).



Рис.2. Таблица «Выдачи\_книги1»

Таблица «Книги» (Рис.3.) состоит из десяти полей: «Код\_книги» (тип данный bigint), «Название\_книги» (тип данный varchar(50)), «Автор\_книги» (тип данный varchar(50)), «Жанр\_книги» (тип данный varchar(50)), «Год\_издание» (тип данный date), «Издательство» (тип данный varchar(50)), «Место\_издательство» (тип данный varchar(50)), «Количество\_страниц» (тип данный int), «Количество\_экземпляров» (тип данный int), «Код\_накладной» (тип данный bigint).



Рис.3. Таблица «Книги»

Таблица «Накладная» (Рис.4.) состоит из четырнадцати полей: «Код\_накладной» (тип данный bigint), «Дата\_накладной» (тип данный date), «Автор\_книги» (тип данный varchar(50)), «Год\_издание» (тип данный date), «Жанр\_Книги» (тип данный varchar), «Количество\_книг» (тип данный int), «Код\_поставщика» (тип данный bigint), «Цена», «Код\_книги» (тип данный bigint),

«Название\_книги» (тип данный varchar(50)), «Издательство» (тип данный varchar(50)), «Количество\_страниц» (тип данный int), «Место\_издание» (тип данный varchar(50)).

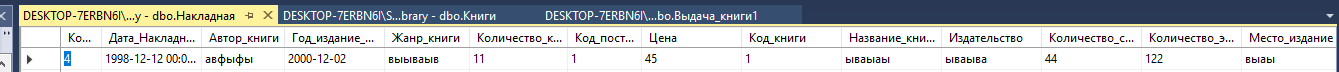


Рис.4. Таблица «Накладная»

Таблица «Препараты» (Рис.5.) состоит из пяти полей: «Код\_поставщика» (тип данный bigint), «Наименование» (тип данный varchar(50)), «Адрес» (тип данный varchar(50)), «Телефон» (тип данный varchar(50)), «E\_mail» (тип данный varchar(50)).

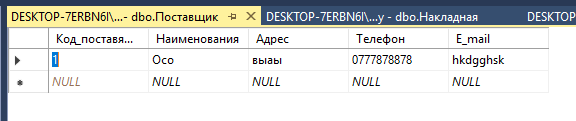


Рис.5. Таблица «Поставщик»

Таблица «Сотрудники» (Рис.6.) имеет четыре поля: «Код\_сотрудника» (тип данных числовой), «ФИО» (тип данный varchar(50)), «Дата\_рождения» () и «Телефон» (тип данный varchar(50)).

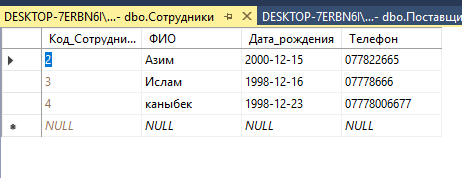


Рис.6. Таблица «Сотрудники»

Таблица «Сотрудники» (Рис.7.) состоит из шести полей: «Код\_Читателя» (тип данный bigint) , «ФИО» (тип данный varchar(50)), «Дата\_рождения» (тип данный date), «Адрес» (тип данный varchar(50)), «Телефон» (тип данный varchar(50)), «Предупреждения» (тип данный varchar(50)).

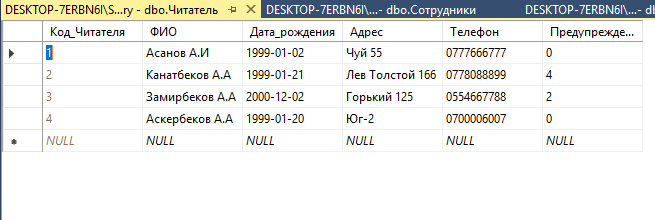


Рис.7. Таблица «Читатель»

После создания таблиц, связываем их в единую схему данных используя связи. Полученная схема данных представлена на Рис.8.

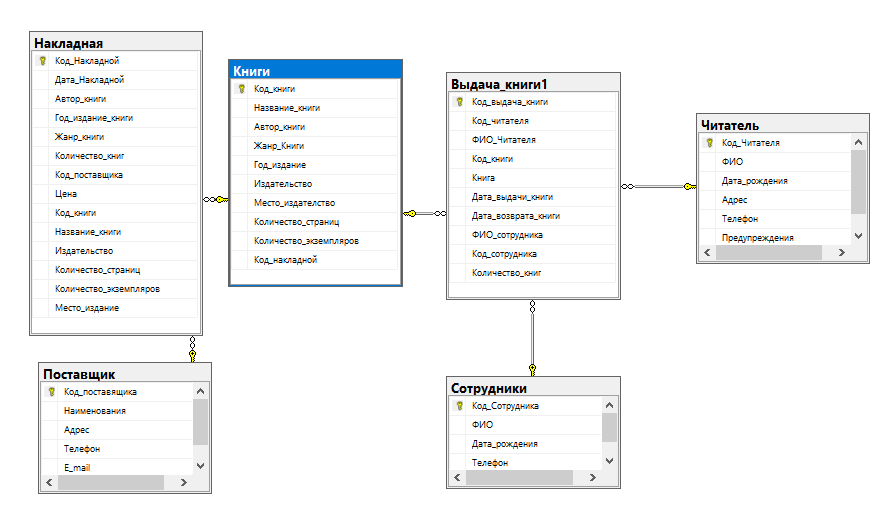


Рис.8. Схема данных

**8.Интерфейс программы:**

Всего в проекте: 10 форм.

1 форма – форма при запуске – .

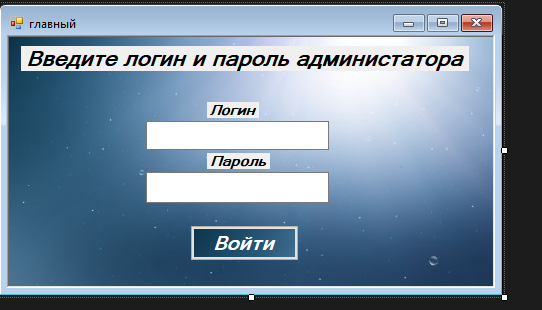


Рис.20

Данная форма Содержит два TextBox1 для ввода пароли и для логина и один кнопка для

входа в главный меню администратора или для главный меню сотрудника

2 форма – Главное меню администратора .

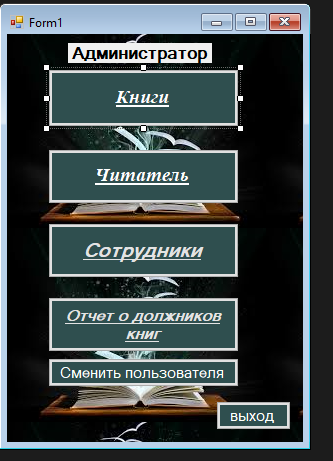


Рис.21.Форма главного меню администратора

Содержит 6 кнопок открытия форм с таблицами базы данных и 6 кнопка для выхода из приложения.

Код открытия форм: Form2.Show() …

Код для выхода: Close()

Также форма имеет определённые свойства:

Text – Главное меню (у каждой формы будет свой текст, соотв. содержанию).

BackColor, BackgroundImage, Font – изменяются для каждой формы индивидуально, это касается и всех элементов управления.

FormBorderStyle – Fixed3D

Icon – значок формата .ico

MaximizeBox, MinimizeBox = False

3 форма – Главное меню сотрудника .

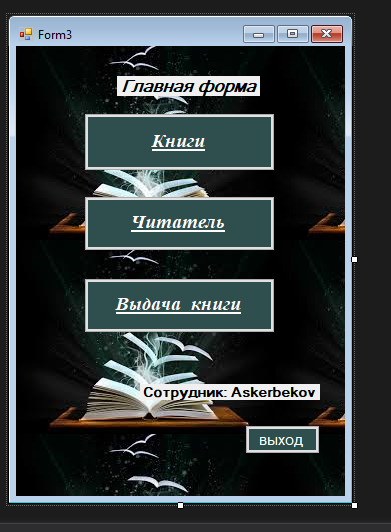


Рис.21.Форма главного меню администратора

Содержит 4 кнопок открытия форм с таблицами базы данных и 4 кнопка для выхода из приложения.

Код открытия форм: Form3.Show() …

Код для выхода: Close()

Также форма имеет определённые свойства:

Text – Главное меню (у каждой формы будет свой текст, соотв. содержанию).

BackColor, BackgroundImage, Font – изменяются для каждой формы индивидуально, это касается и всех элементов управления.

FormBorderStyle – Fixed3D

Icon – значок формата .ico

MaximizeBox, MinimizeBox = False

4 форма – форма с таблицей Сотрудники.

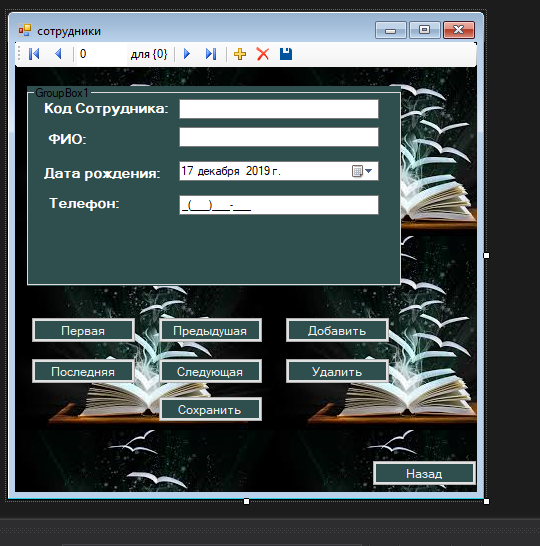


Рис.21.Главная форма информации о сотрудниках

Это первая форма с информацией из БД. Вся информация переносится на форму путём переноса необходимых данных с помощью мышки. Если TextBox, к примеру, необходимо заменить на MaskedTextBox или ComboBox, то данные БД, для данного столбца необходимо перенести заново.

Также, на выбор, предоставляется несколько вариантов переноса:

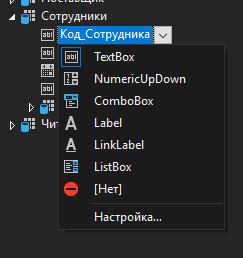


Рис.22.Варианты переноса данных из БД на форму

После переноса данных, создаётся BindingNavigation (или же ToolBar), его можно либо оставить, либо удалить.

На каждой форме с таблицами будут минимум 8 кнопок, эти 8 кнопок – обеспечивают работу с данными:

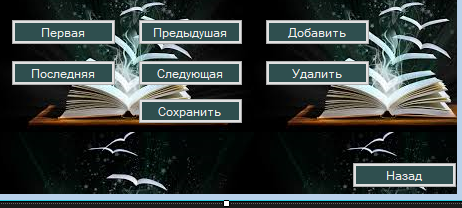


Рис.23.Работа с информацией (для всех главных форм)

Первая: СотрудникиBindingSource.MoveFirst()

Последняя: СотрудникиBindingSource.MoveLast()

Следующая: СотрудникиBindingSource.MoveNext()

Предыдущая: СотрудникиBindingSource.MovePrevious()

Добавить: СотрудникиBindingSource.AddNew()

Удалить: СотрудникиBindingSource.RemoveCurrent()

Сохранить: Me.Validate()

Me.СотрудникиBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.LibraryDataSet)

Назад: Close()

Работа с 4 формай окончена

5 форма – форма с таблицей читатели



Рис.24.Главная форма информации о читателя

После переноса данных, создаётся BindingNavigation (или же ToolBar), его можно либо оставить, либо удалить.

На каждой форме с таблицами будут минимум 8 кнопок, эти 8 кнопок – обеспечивают работу с данными:

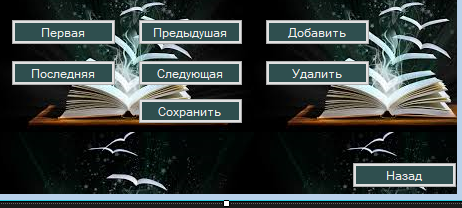


Рис.23.Работа с информацией (для всех главных форм)

Первая: ЧитательBindingSource.MoveFirst()

Последняя: ЧитательBindingSource.MoveLast()

Следующая: ЧитательBindingSource.MoveNext()

Предыдущая: ЧитательBindingSource.MovePrevious()

Добавить: ЧитательBindingSource.AddNew()

Удалить: ЧитательBindingSource.RemoveCurrent()

Сохранить: Me.Validate()

Me. ЧитательBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.LibraryDataSet)

Назад: Close()

Работа с 5 формой окончена

6 форма – форма с таблица книги.

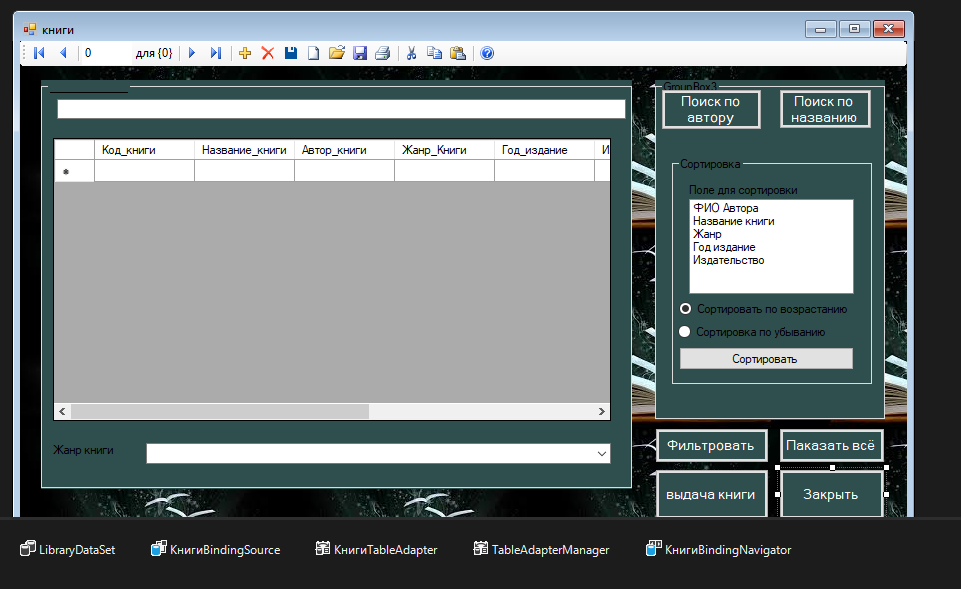


Рис.23.Таблица книги

Отличия форм заключаются только в наборах данных и оформлении. Функционал практически не имеет отличий.

Таблица переносится на форму из источника данных в виде полной таблицы. Первым делом на форме присутствует сортировка.

Данные записываются в ListBox (все существующие столбы в таблице). Сортировать можно по возрастанию и по убыванию. Выбор осуществляется с помощью элемента управления RadioButton. Если параметр сортировки не задан – кнопка для сортировки будет неактивна. Если задан параметр сортировки и выбран метод сортировки – кнопка будет активна и можно произвести сортировку, по её нажатию.

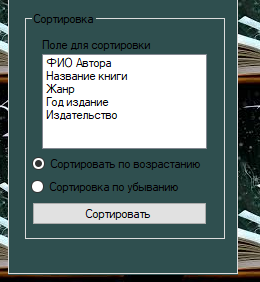


Рис.25.Сортировка в таблице

Ниже сортировки располагаются 4 кнопки.

Кнопка “Фильтровать” выводит жанры книг, которые мы выбрали в ComboBox. Стоит напомнить, что ComboBox использует элементы из БД, в выпадающем списке.

Здесь мы отфильтровали жанр книги «исторический»

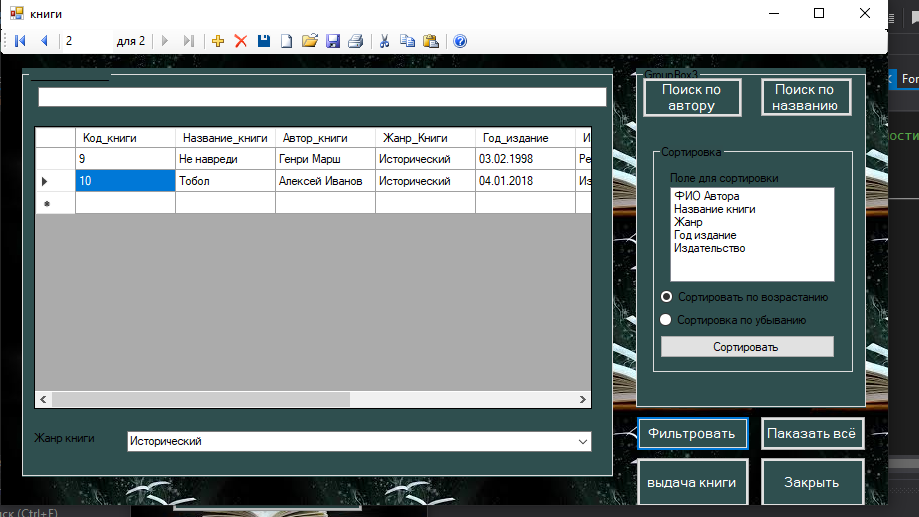


Рис.26.фильтр по жанру

При нажатии кнопки “Показать всё” данные фильтрации сбрасываются.

При нажатии кнопки “Выдачи книг ” данные фильтрации сбрасываются.

Открывается форма выдача книг

Код кнопки

выдача\_книги.Show()

Также есть кнопки “поиск по автору” и “поиск по названию”

Здесь в TextBox3 пишем название автора или название книги и зависимо от условии нажимаем кнопку поиск по автору или поиск по названию книги

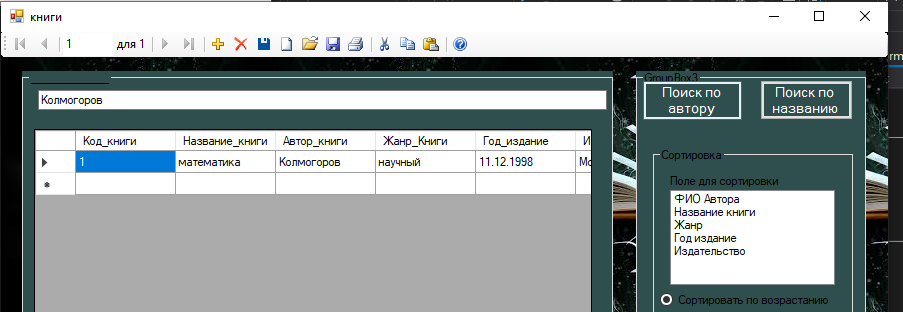


Рис.27.поиск по автору

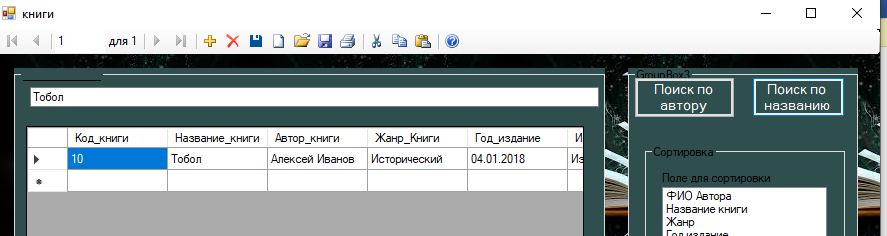


Рис.28.поиск по названию

При нажатие кнопки “закрыть” закрывается форма

7 форма – форма с таблицей выдачи книг

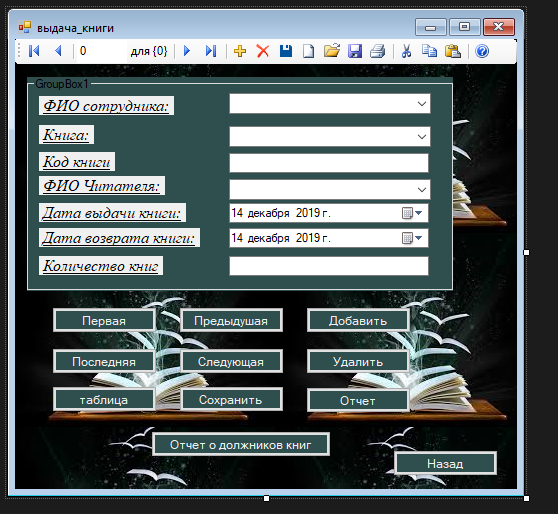


Рис.29.поиск по названию

Это первая форма с информацией из БД. Вся информация переносится на форму путём переноса необходимых данных с помощью мышки. Если TextBox, к примеру, необходимо заменить на MaskedTextBox или ComboBox, то данные БД, для данного столбца необходимо перенести заново.

Также, на выбор, предоставляется несколько вариантов переноса:

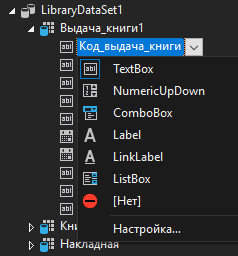


Рис.30 Варианты переноса данных из БД на форму

После переноса данных, создаётся BindingNavigation (или же ToolBar), его можно либо оставить, либо удалить.

На каждой форме с таблицами будут минимум 11 кнопок, эти 11 кнопок – обеспечивают работу с данными:

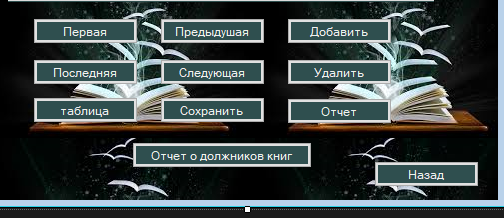


Рис.31.Работа с информацией

Первая: Выдача\_книги1BindingSource.MoveFirst()

Последняя: Выдача\_книги1BindingSource.MoveLast()

Следующая: Выдача\_книги1BindingSource.MoveNext()

Предыдущая: Выдача\_книги1BindingSource.MovePrevious()

Добавить: Выдача\_книги1BindingSource.AddNew()

Удалить: Выдача\_книги1BindingSource.RemoveCurrent()

Сохранить: Me.Validate()

Me. Выдача\_книги1BindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.LibraryDataSet)

Назад: Close()

Отчет : отчет.Show()

Таблица : таблица\_выдачи\_книг.Show()

Отчет о должников: отчет\_должников\_кник.Show()

8 форма – форма с таблицей выдачи книг в виде таблицы

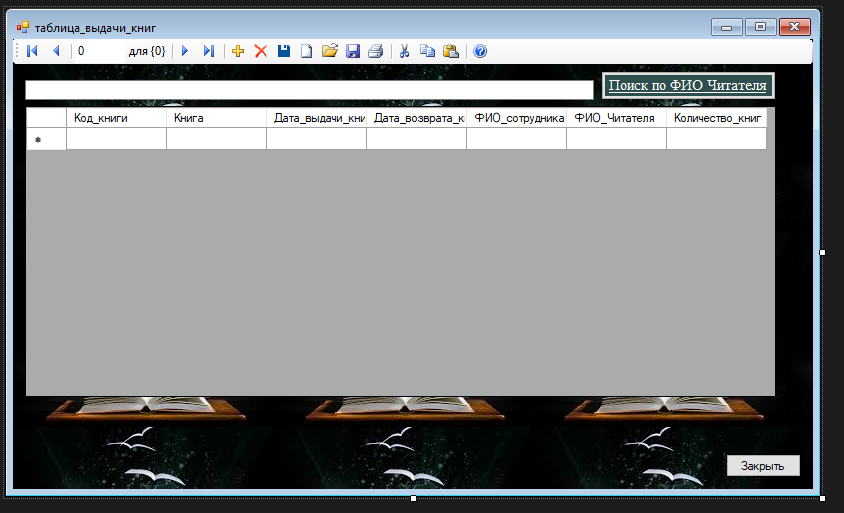


Рис.32.выдачи книг виде таблицы

Это форма с информацией из БД. Вся информация переносится на форму путём переноса необходимых данных с помощью мышки.

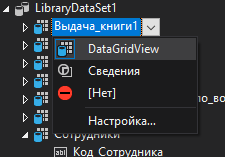


Рис.33. переноса данных из БД на форму

В данном форме есть 2 кнопка для поиска Читателя по ФИО и для закрытие формы

Код кнопки Поиск по ФИО Читателя

Выдача\_книги1BindingSource.Filter = "ФИО\_Читателя='" & TextBox1.Text & "'"

Код кнопки закрыть

Close()

9 форма – работа с отчетом

Для отображения отчётов используются обычные формы виндовс, об этом было сказано ранее. Сам отчёт создаётся иначе.

Для создания отчёта необходимо проделать следующие действия: Автосалон – добавить – создать элемент – отчёты (reporting) – отчёт.

Создаётся новый отчёт. При создании необходимо указать DataSet. Это данные (столбцы таблиц) которые будут использоваться в отчёте.

После чего появляется отчёт. Данные в отчёте вставляются (выбираются) справа; слева описывается, какие данные были использованы.

Отчёт можно всячески редактировать и изменять внешний вид.

Все отчёты строятся по одному и тому же принципу, подключение отчёта к форме происходит идентично.

Коды в отчётах не используются. Поэтому фактической разницы в отчётах нет. обычных отчётов по таблицам заключается в выборе данных из всех таблиц.

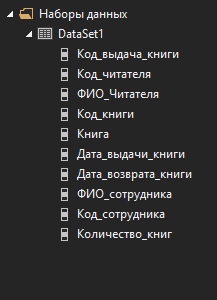


Рис.34.Пример набора данных для отчёта (DataSet) по шестому отчёту

Этот отчёт основан на заранее подготовленном представлении, в котором участвовала вся необходимая информация из всех таблиц.

Вид полного отчёта по заказам в Visual Studio:

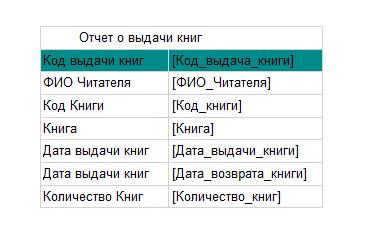


Рис.35.Отчёт в Visual Studio (не полностью, в связи с большим объёмом данных)

Подключение к форме осуществляется добавлением специального элемента управления ReportViewer на необходимую форму. Необходимо указатель параметр – Закрепить в родительском контейнере (чтобы он занимал всю форму). Затем выбрать необходимый отчёт:

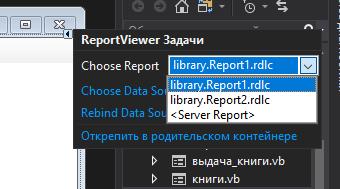


Рис.36.Пример ReportViewer для первого отчёта на форме.

При нажатии кнопки получаем следующую информацию:

Отчёт по сотрудникам при работе программы (представление в Form8):

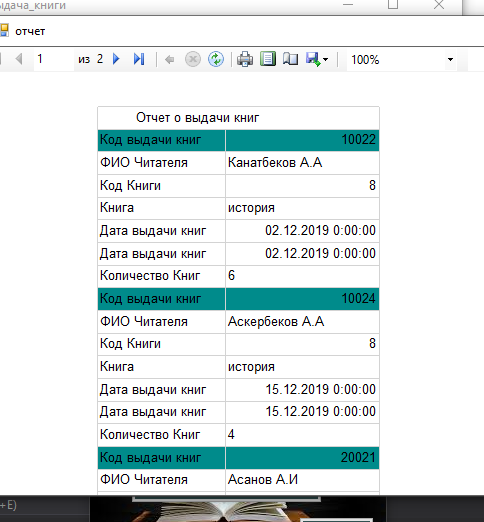


Рис.37.Отчёт о выдачи книг

Отчёт о должников читателей (представление в Form9):

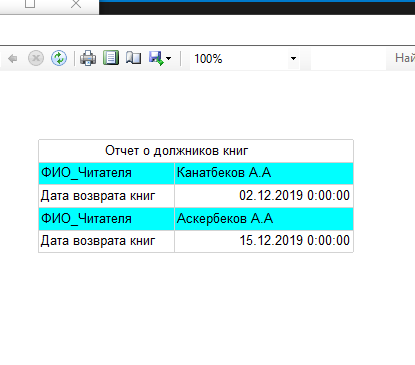


Рис.38.Отчёт о должников книг

# **Заключение:**

Одной из основных задач данной информационной системы является предоставление своевременного и полного обеспечения необходимой информацией о состоянии библиотеки, поступивших в библиотеку книгах и о читателях. В функции данной информационной системы также входит обеспечение надлежащей организации обработки и хранения полученной ранее информации, контроль за ее использованием и учет.

Данное информационная система позволяет информации сокращает до минимума ручные операции, повышает производительность труда сотрудников библиотеки, улучает качество и повышает эффективность управления в библиотеке.

# **Список литератур:**

1. Карпова Т.С. Базы данных. Модели, разработка, реализация/СПб.: Питер,2002. – 304 с.
2. <http://www.interface.ru/rational/rosemain.htm>
3. Воронько К.Л. Организация библиотечных фондов и каталогов. Учебник для библ. техникумов и библ. отд-ний культ.-просвет. училищ. М., «Книга», 1972. – 397 с.: ил.
4. А.М. Вендров “CASE-технологии. Современные методы и средства проектирования информационных систем”, cтатья c сайта <http://www.citforum.ru>
5. Маклаков С.В. Bpwin и Erwin. CASE-средства разработки информационных систем. М.: ДИАЛОГ-МИФИ, 2000 – 256 с.
6. Уэнди Боггс, Майкл Боггс. UML и Rational Rose. Лори, 2004 – 510 с.
7. Дейт К.Дж. Введение в системы баз данных, 6-е издание: Пер. с англ.- К.; М.;СПб: Издательский дом «Вильямс», 1999. - 848 c.