

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ФАКУЛЬТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ

**АВТОМАТИЗАЦИЯ УЧЁТА В ШКОЛЬНОЙ БИБЛИОТЕКЕ ( НА  
ПРИМЕРЕ РАЗУМЕНСКОЙ СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ  
ШКОЛЫ №2)**

Выпускная квалификационная работа  
обучающегося по направлению подготовки 09.03.03 «Прикладная  
информатика»  
очной формы обучения, группы 07001304  
Калласа Виталия Андреевича

Научный руководитель  
к.т.н., доцент  
Черноморец А. А.

БЕЛГОРОД 2017

## СОДЕРЖАНИЕ

Введение.....	4
1 Аналитическая часть.....	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.1.1 Характеристика предприятия, включая организационную структуру управления предприятием.....	7
1.1.2 Краткая характеристика подразделения.....	8
1.2 Постановка задачи.....	11
1.2.1 Цель и назначение автоматизированного варианта решения задачи.....	11
1.2.2 Общая характеристика организации решения задачи на ЭВМ.....	12
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	13
1.4 Анализ существующих разработок.....	16
2 Проектная часть.....	19
2.1 Обоснование проектных решений.....	19
2.1.1 Обоснование проектных решений по техническому обеспечению.....	19
2.1.2 Обоснование проектных решений по информационному обеспечению.....	21
2.1.3 Обоснование проектных решений по программному обеспечению.....	22
2.1.4 Обоснование выбора программных средств.....	23
2.2 Описание базы данных.....	25
3 Программная реализация проектных решений.....	32
3.1 Программное обеспечение задачи.....	32
3.2 Технологическое обеспечение задачи.....	33
3.3 Описание контрольного примера реализации проекта.....	36
3.4 Обоснование эффективности экономической разработки.....	41
3.4.1 Выбор и обоснование показателей эффективности.....	41
3.4.2 Расчет показателей эффективности.....	42

Заключение.....	46
Список использованных источников.....	48
Приложение А.....	52
Приложение Б.....	54
Приложение В.....	60

## ВВЕДЕНИЕ

Одним из важнейших направлений в работе библиотек является организация учёта книжного фонда и всех форм библиотечного обслуживания. Анализ полученных данных позволяет оценить качество и эффективность деятельности библиотеки, а также указать вектор её дальнейшего развития [13].

Учёт библиотечного фонда является важнейшим условием его сохранности и контроля за движением и объёмом [6]. Особенно остро этот вопрос стоит в последние годы, в связи с реформированием учёта библиотечных фондов в рамках национальной программы сохранности библиотечных фондов. Основной её задачей является создание единой системы государственного учёта библиотечных фондов страны.

Учёт отражает пополнение фонда, выдачу книг, величину библиотечного фонда и служит основой для государственного статистического учёта, планирования деятельности, отчётности, обеспечения сохранности фонда, контроля за наличием и движением книжных изданий в библиотеке [8].

Школьная библиотека – одно из важнейших структурных подразделений современной школы [15]. Библиотека в России обязательна в каждом образовательном учреждении. Обеспеченность библиотеки необходимыми информационными материалами учитывается при лицензировании общеобразовательного учреждения. На законодательном уровне также закреплён бесплатный доступ ко всем ресурсам и материалам библиотеки. При этом материалами могут быть не только печатные издания, но и различные информационные данные на любых носителях.

Библиотекарь — профессионал по обработке информации в библиотечной и информационной науках, который выполняет организацию и управление информационными услугами и материалами для пользователей библиотеки [18]. Как правило, библиотекари работают в государственных

библиотеках или библиотеках высших или средних учебных заведений, начальных или средних школ, медиа-центрах, библиотеках в рамках предприятий или компаний.

Учёт библиотечного фонда заключается в том, что процессы автоматизации затронули не только производственную деятельность человечества, но и информационное пространство, в частности библиотеки [6]. Невозможно представить жизнь человека без интернета и компьютера. Автоматизация учёта в библиотеке позволяет библиотекарям избавиться от большинства сложных процессов, что обеспечивает более быстрое, полное и качественное обслуживание посетителей.

Учет библиотечного фонда – совокупность правил и процедур, обеспечивающих регистрацию и обобщение сведений о величине, составе, движении фонда библиотеки и его стоимости. Учет позволяет отследить поступление документов в фонд, их выбытие, объём всего библиотечного фонда, состав документов по различным признакам. Учет библиотечного фонда независимо от способов его ведения - традиционных или с использованием компьютерных технологий, должен обеспечивать полноту и достоверность учетной информации, документированное оформление каждого поступления в фонд и выбытия из него, совместимость приемов и форм учета, их надежность в условиях применения традиционной и автоматизированной технологий [11].

Актуальность темы данной ВКР также обусловлена тем, что в библиотеке МОУ «Разуменская средняя общеобразовательная школа №2» в настоящее время отсутствует какая – либо информационная система. Когда учёт в библиотеке не автоматизирован, все данные о передвижении книг приходится записывать вручную, сортировать и хранить на каталожных карточках.

Объектом исследования в данной работе является деятельность школьной библиотеки.

Предметом исследования является процесс учёта книг.

Целью данной работы является автоматизация учёта книг в школьной библиотеке.

Задачами данной работы являются:

- Изучение теоретических и практических аспектов учёта книг в школьных библиотеках.
- Проектирование системы учёта книг в библиотеке.
- Разработка системы учёта книг в библиотеке.
- Расчёт эффективности системы.

Работа состоит из введения, трёх глав и заключения.

Введение раскрывает актуальность, определяет степень научной разработки темы, цель исследования.

В первой главе даётся общая характеристика предметной области, структура предприятия, изучается его функционирование, рассматриваются примеры существующие информационных систем.

Во второй главе рассматриваются проектные решения и процессы внутри области задачи. Также описывается используемая база данных.

В третьей главе показан этап разработки программы, на рисунках представлена работа программы, а также производится расчёт показателей экономической эффективности. В заключении подводятся итоги проделанной работы, формируются окончательные выводы по разработке автоматизированной системы.

## **1 Аналитическая часть**

### **1.1 Технико-экономическая характеристика предметной области**

#### **1.1.1 Характеристика предприятия, включая организационную структуру управления предприятием**

Библиотека в школе является главным информационным, образовательным, культурно - досуговым элементом образовательной среды.

Для каждой библиотеки вырабатывается своя стратегия развития[14].

Школа должна анализировать и раскрывать творческий интеллектуальный потенциал ребёнка.

Библиотека в школе является информационным центром, и оказывает помощь учителям и учащимся в воспитательном процессе. Школа не сможет без хорошей библиотеки выполнять на должном уровне воспитательные и образовательные функции.

Школьная библиотека обязана предоставлять идеи и информацию, столь нужные для успешного ориентирования и существования в современном информационном обществе, где важнейшую роль играют знания, а также развивает воображение и воспитывает гражданскую ответственность [16].

Библиотека занимает активную роль во взаимодействии библиотекаря и ученика, которая строится на принципах «педагогике сотворчества». Воспитание культуры чтения, формирование интереса к книге, развитие творческого мышления осуществляется библиотекой с помощью традиционных и инновационных методов читательского и личностного развития. Обычно это происходит в виде индивидуальной работы с учеником, построенной в виде диалога, включая внеурочную жизнь школьника.

Школьная библиотека можно имеет ряд факторов, способствующих формированию активной читательской деятельности, готовности ребенка реализовать свой творческий потенциал, творить. Ребёнок и библиотекарь находятся рядом в течение десяти лет, что позволяет следить за развитием ребёнка [24]. Так как библиотекарь тесно работает вместе с учителями, это позволяет видеть в ребёнке не только обычного пользователя информации, читателя, но также и ученика.

Школьная библиотека не только является звеном образовательной среды, она сама и есть та самая материальная и духовная среда, направленная на активизацию творческой деятельности детей и взрослых [22].

Библиотеки с древнейших времён были хранителями духовных богатств общества, помощниками пользователей. Со временем менялся человек, развивались наука и техника, добавлялись и усложнялись школьные программы, дети становились умнее, «продвинутое», но люди так и не изобрели лучшего способа для развития интеллекта и мышления, чем чтение. Но сегодня книгу постепенно вытесняют с её места монополиста в области источника знаний. Различные видео и аудиоматериалы, информационные ресурсы и программы обучения иногда оказываются удобнее, полезнее и предпочтительней книги.

### **1.1.2 Краткая характеристика подразделения**

Библиотеки, осуществляющие информационное обслуживание российской системы образования, находятся в подчинении двух министерств: культуры и массовых коммуникаций, образования и науки [7]. Опытные библиотекари выделяют в качестве особо важного направления работу с учениками и преподавателями.

До середины семидесятых годов пролого века школьная библиотека развивалась так же, как и обычная дошкольная, однако её фонды, функции и задачи были направлены только на поддержку учебного процесса [23].

Совокупный фонд школьных библиотек составляет около 1 миллиарда томов. Каждая из них имеет в среднем в своем фонде 15 тысяч экземпляров, почти половину которых составляет учебная литература. Также в среднем на одну библиотеку приходится 400 читателей-школьников.

Существенная разница также лежит между поселковой и среднестатистической городской библиотекой. Объём фонда в городе – 33 тыс. экземпляров, в посёлке – 13 тыс. экземпляров, в том числе 16 тыс. и 6 тыс. учебников соответственно.

Книгообеспеченность на одного школьника составила около 32 экземпляров литературы, в том числе 22 экземпляра учебников.

Зачастую библиотеки испытывают трудности с комплектованием. Новые книги поступают не систематически и в небольшом количестве. Образовательные учреждения не всегда в состоянии заказывать необходимую им литературу.

Фонды современных библиотек составляют не только традиционные печатные издания, но также и нетрадиционные пока что виды документов (кино-, фото-, и фоно- документы).

Большинство школьных библиотек имеют в наличии только самые основные учебники из Федерального перечня, что затрудняет выбор и вариативность учебных программ.

Библиотеки контролируют процесс работы пользователей с учебниками и обязаны постоянно отчитываться перед органами контроля за образованием [17].

Современные школьные библиотеки повсеместно отказываются от единовременной выдачи комплектов учебников в начале учебного года. Этот процесс теперь ведётся в течение всего года. Это связано с непостоянной поставкой изданий, использованием альтернативных учебников, необходимостью обращения к изданиям прошлых лет при изучении отдельных тем и подготовки рефератов и докладов.

Работа с учебным фондом объемна, разнообразна по содержанию и технологиям, и требует от школьных библиотек весомых трудозатрат.

В настоящее время происходит процесс адаптации библиотечного обслуживания к быстро меняющимся условиям общественной жизни, обновление его средств, форм и методов – это утверждение в полной мере справедливо и для школьных библиотек [12].

Технология работы - совокупность операций и процессов, отвечающих за стабильное функционирование школьной библиотеки [34].

Технология определяется на основе инструкций и методических материалов по следующим вопросам:

- формирование фондов;
- планирование и отчетность;
- комплектование и хранение фондов;
- ведение справочно-библиографического аппарата библиотеки;
- учет и обработка документов;
- организация справочно-библиографического обслуживания;
- работа с читателями.
- продвижение информационной продукции и услуг;

## 1.2 Постановка задачи

### 1.2.1 Цель и назначение автоматизированного варианта решения задачи

Разрабатываемая система предназначена для учета поступления и использования книг.

Объекты автоматизации:

- Журнал для записи и выдачи книг;
- Журнал новых поступлений;

При автоматизации учета выдачи, списания, обновления фонда необходимо получать, обрабатывать и хранить информацию в журналах.

Эксплуатация объекта автоматизации не требуют соблюдения каких-либо специальных мер. Система эксплуатируется в оборудованном помещении.

Разрабатываемая система не является сложной, и для её обслуживания не требуются специалисты высокой квалификации. Систему может обслуживать один пользователь, уверенно работающий с MS Office.

Для обеспечения необходимых расчетов, ввода и хранения информации будут использованы современные СУБД: реляционные или объектно-ориентированные, например, СУБД MySql [26], так как они широко распространены и запускаются на большинстве современных систем.

Требуемые технические средства: ПК с наличием usb-входов, принтер.

Данная система призвана сделать работу, связанную с учётом в школьной библиотеке, удобнее и быстрее.

## 1.2.2 Общая характеристика организации решения задачи на ЭВМ

Для достижения цели ВКР необходимо выявить основные потоки информации, проанализировать их представление и надобность периодичной обработки соответствующих документов [35].

При разработке базы данных «Школьная библиотека» было проведено обследование предметной области. В результате были проанализированы основные библиотечные процессы, выявлены информационные потребности работников библиотеки, изучены формы входных и выходных документов, отчетов.

В разрабатываемой автоматизированной информационной системе необходимо учесть, что в качестве входных потоков информации должны быть рассмотрены:

- информация о читателях;
- информация о книгах;
- информация о жанрах;
- информация о выданных книгах;
- информация о возврате книг.

Описание выходных документов представляет собой перечень вторичных документов, как например разного вида отчёты.

Автоматизация рутинных операций позволит работнику значительно ускорить обработку заявки на книгу, сильно уменьшить время поиска [39].

### 1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

В библиотеке учебного заведения «Разуменская средняя общеобразовательная школа №2» не было обособленной системы учёта. Анализ вёлся на основе данных, которые библиотекари сами вносили в журналы и карты. На рисунке 1 представлена контекстная диаграмма работы библиотеки «Как есть». Показаны основные информационные потоки.

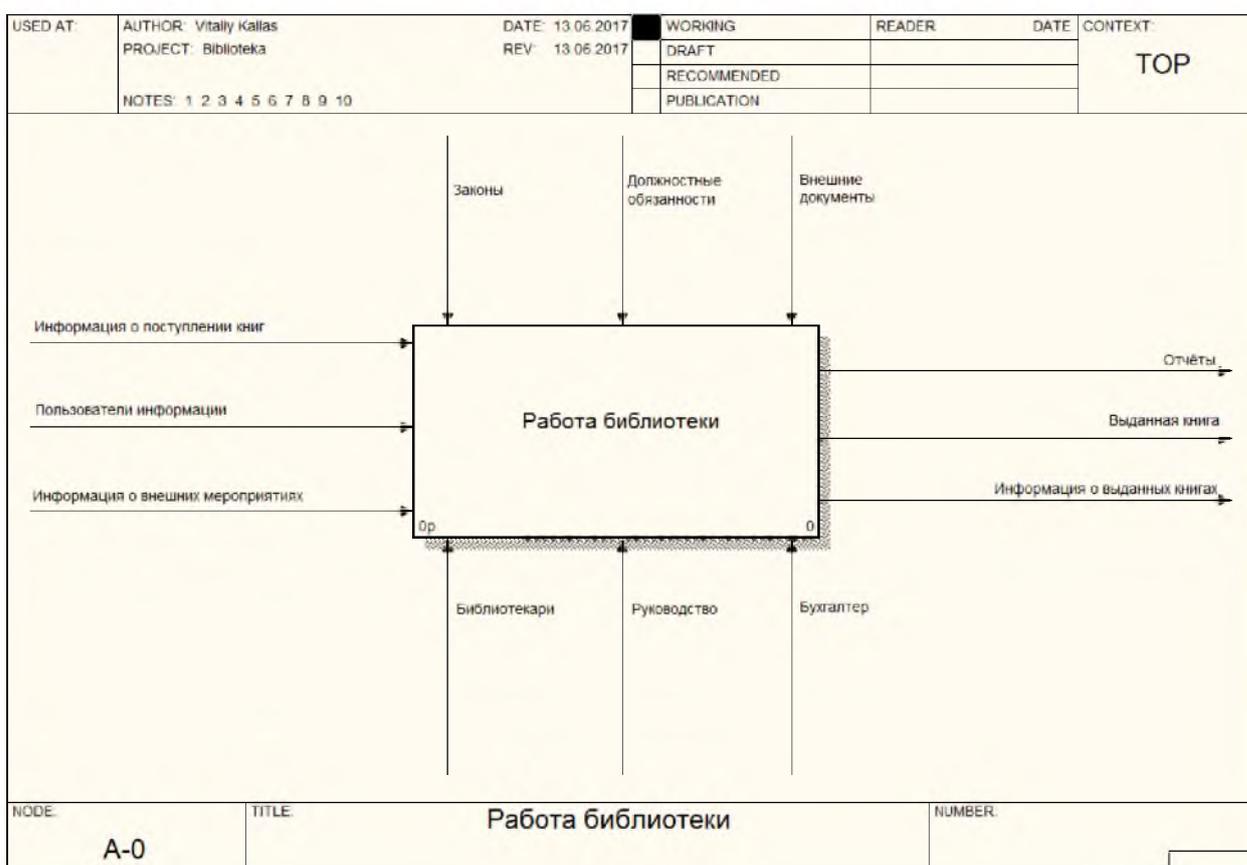


Рисунок 1 – Контекстная диаграмма «Как есть»

На рисунке 2 представлен в общем виде тот же процесс функционирования школьной библиотеки, но развернуто. Показаны основные информационные потоки как граничные, так и потоки, порождаемые соответствующими подпроцессами.

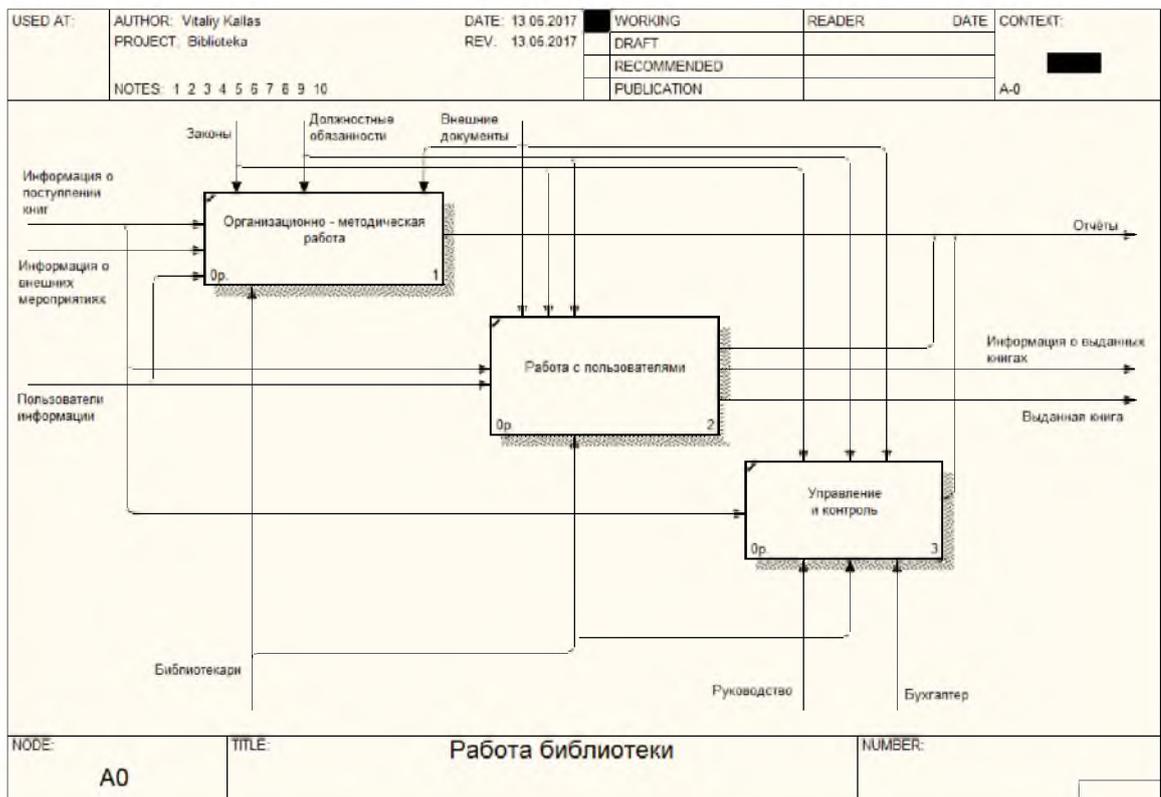


Рисунок 2 – Декомпозиция диаграммы «Как есть»

На рисунках 3, 4 и 5 показана деятельность предприятия с применением технологий данной работы.

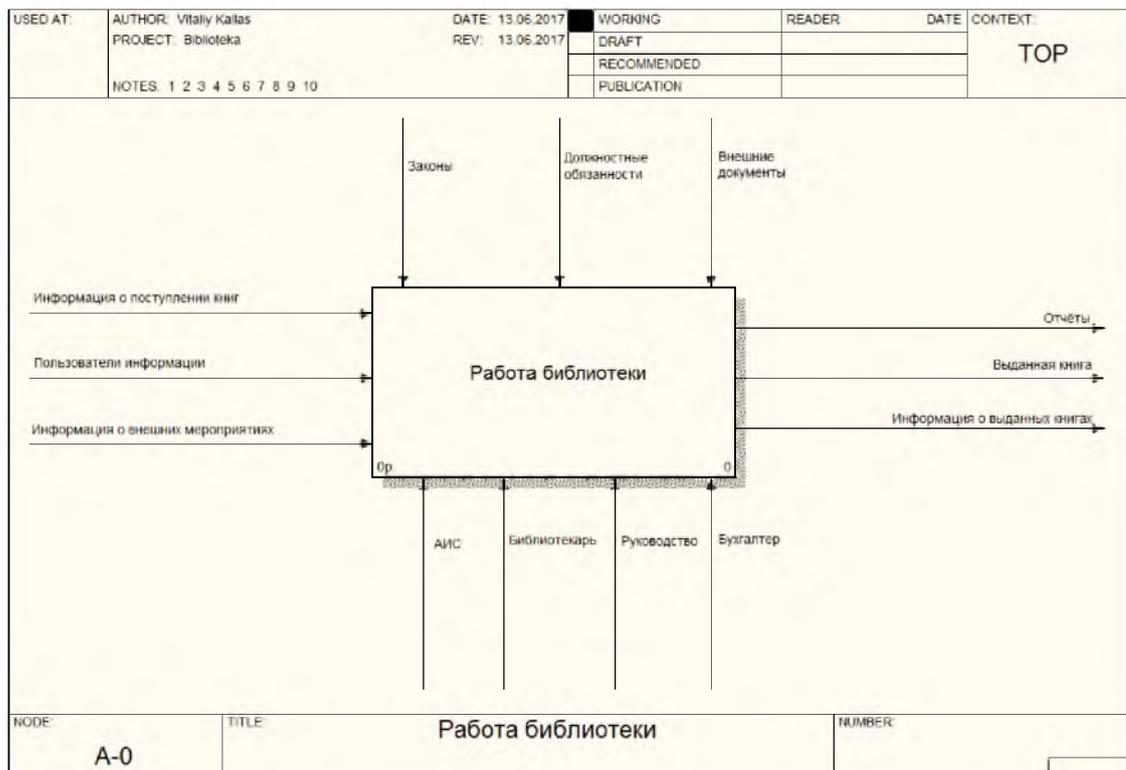


Рисунок 3 - Контекстная диаграмма «Как должно быть»

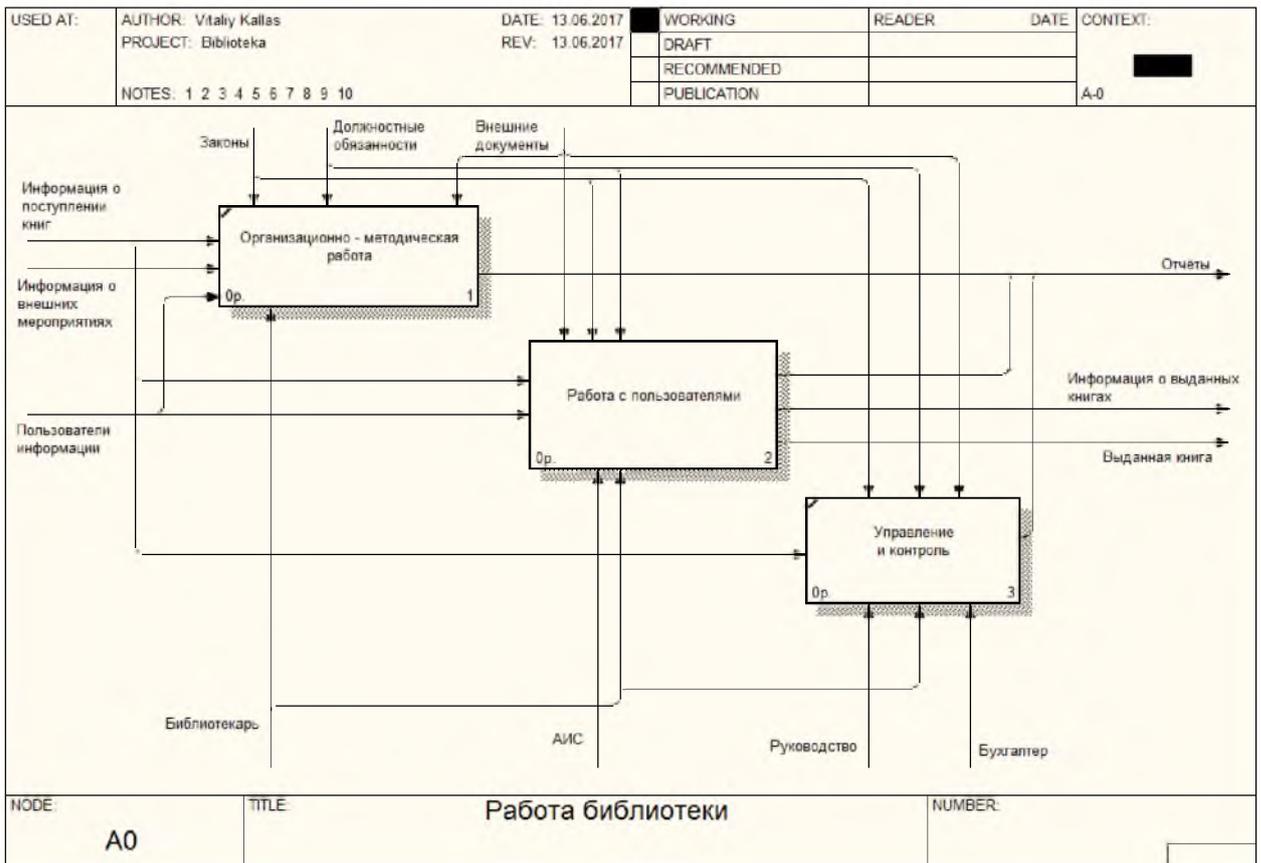


Рисунок 4 – Диаграмма декомпозиции «Как должно быть»

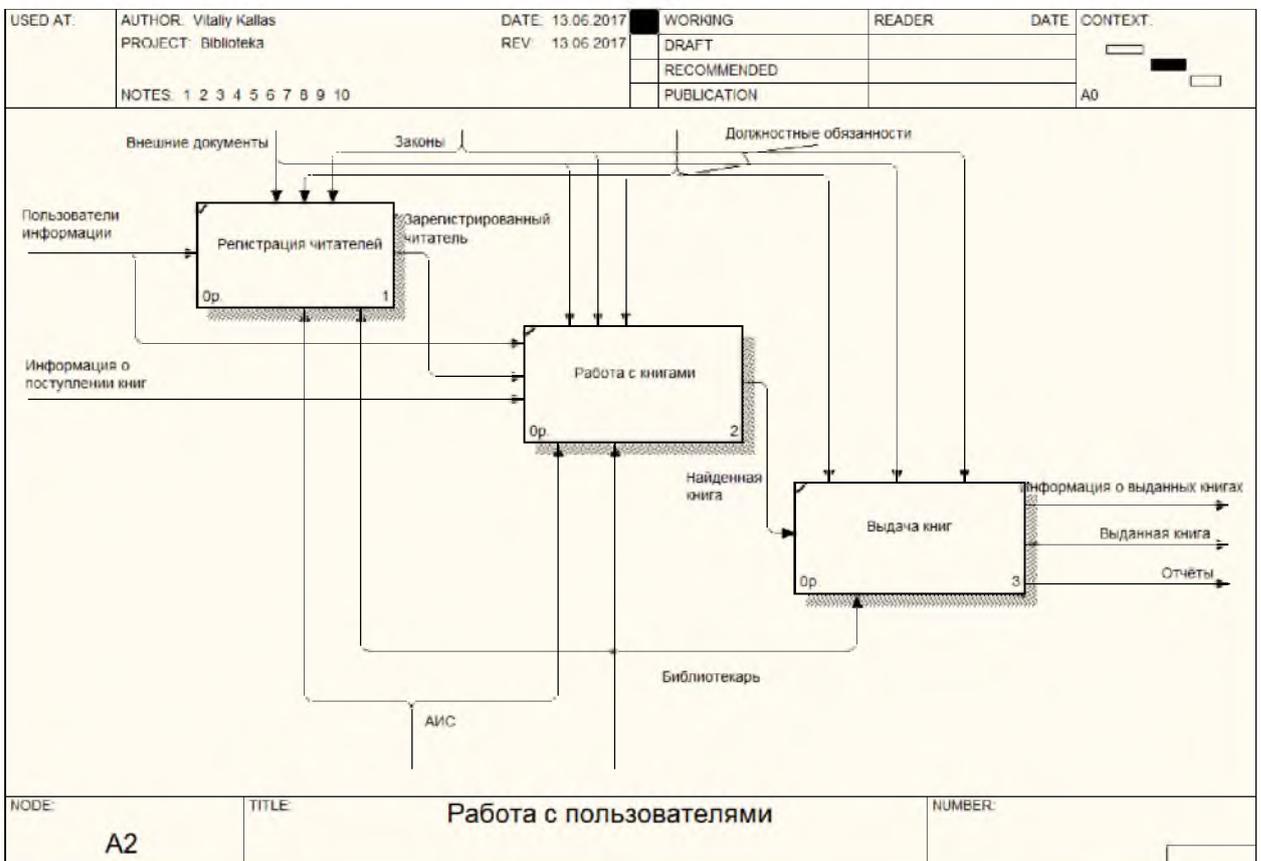


Рисунок 5 – Диаграмма декомпозиции «Работа с пользователями»

## 1.4 Анализ существующих разработок

На данный момент существует огромное количество информационных систем для автоматизации учёта библиотек. Самые известные из них:

- Система «Учёт книг»;
- Библиотека MD 4.3;
- 1С:Библиотека Колледжа;
- Автоматизированная информационно-библиотечная система OPAC-Global;
- АС-Библиотека-3;
- Система ИРБИС.

Все вышеперечисленные программы это очень большие системы, охватывающие каждый аспект библиотечной работы. Но для небольших учебных учреждений такие решения являются очень сложными, громоздкими и дорогими.

Именно в таких случаях встает вопрос о создании собственных специализированных приложений, обладающих только необходимым для конкретного учреждения функционалом, требующимся для данной деятельности.

Такие приложения обладают положительными аспектами:

- Они просты в эксплуатации;
- Стоимость их разработки намного ниже любого широко известного проекта;
- Не содержат лишнего функционала;
- Поддерживают возможность модернизации.

На рисунках 6 и 7 показаны интерфейсы ведущих программ в данной области.

Программа «Учёт книг» является одной из самых распространённых систем учёта книг в библиотеках, но в рассматриваемом учреждении она

будет не столь эффективна, ввиду того, что чрезмерно сложна в обращении, в неё трудно вносить информацию, и она является платной.

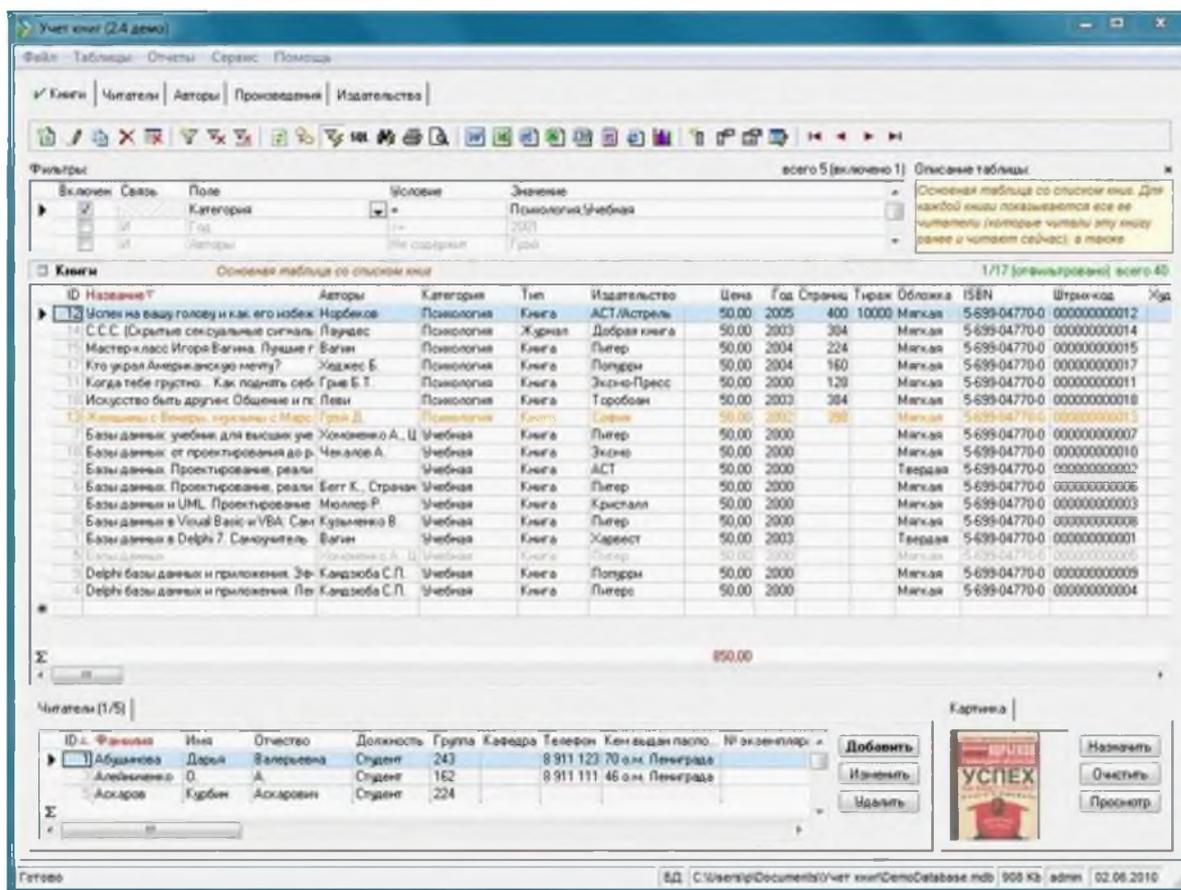


Рисунок 6 – Интерфейс программы «Учёт книг»

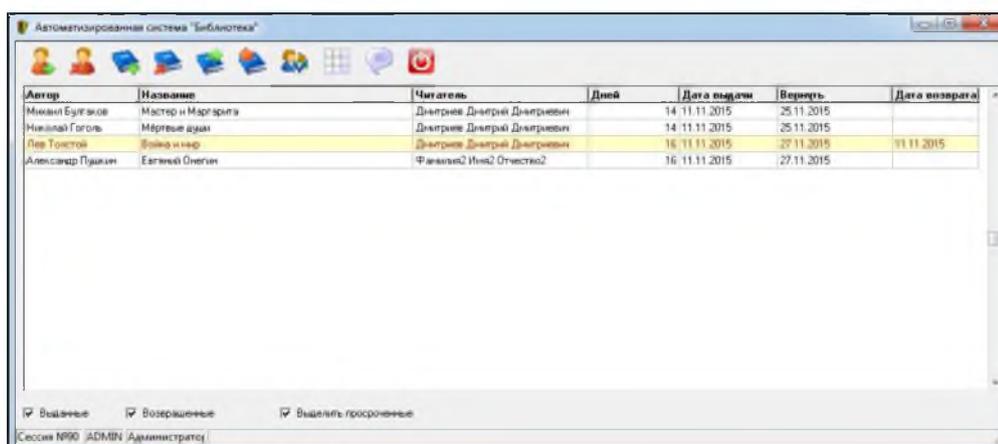


Рисунок 7 – Интерфейс программы «АИС «Библиотека»»

АИС «Библиотека» представляет собой не настолько популярный продукт, как система «Учёт книг», что сказывается на качестве исполнения и функциях.

**Вывод:** В первом разделе была изучена работа библиотеки, построены контекстные диаграммы, проанализированы существующие информационные системы, выявлены их сильные и слабые стороны. На основе полученных данных и будет выполняться дальнейшая работа.

## 2 Проектная часть

### 2.1 Обоснование проектных решений

#### 2.1.1 Обоснование проектных решений по техническому обеспечению

Для решения поставленной задачи необходимо соответствующее техническое обеспечение. Техническое обеспечение данного проекта включает в себя непосредственно ЭВМ (системный блок), монитор, клавиатуру, мышь и принтер.

При выборе ЭВМ обычно руководствуются рядом характеристик. К ним относятся надежность, стоимость, производительность, объем памяти и другие [37].

От значения вышеуказанных характеристик зависит возможность работы с требуемыми программными средствами.

При выборе ЭВМ для реализации комплекса поставленных задач учитываются такие характеристики:

- скорость обработки информации (тактовая частота процессора);
- объем оперативной памяти, этот фактор также влияет на скорость обработки информации;
- объем жесткого диска, который влияет на возможности хранения данных;
- наличие периферийных устройств;
- другие технические характеристики ПЭВМ.

Библиотека располагает следующим оборудованием:

- PC AT Pentium;
- оперативная память – 4 гб;
- жесткий диск - 50 Гб;
- дисковод 3.5”;

- клавиатура, мышь, принтер.

Объем оперативной и внешней памяти являются достаточными для широко используемых в настоящее время СУБД, таких как FoxPro , Clarion , Access , Paradox , Delphi, MySql и др. Характеристики быстродействия выбранного класса машин таковы, что позволят системе работать без видимых оператору задержек, а это благотворно скажется на психологическом состоянии работника и на качестве самой работы. Применять машины более низкого быстродействия не рекомендуется, так как программа будет работать медленно, что приведет к снижению производительности труда и отрицательно скажется на настроении сотрудников.

Для распечатки документов необходим принтер. Существует несколько типов принтеров: матричные, струйные, лазерные. Матричные принтеры имеют более низкую скорость печати, чем струйные и лазерные, но относительно низкую цену, и для них не требуются дорогостоящие чернила или порошок. Кроме того, матричные принтеры позволяют вручную управлять расположением текста на странице путем соответствующего размещения бумаги. Это очень удобно, особенно когда необходимо вывести на один лист разнородные данные, так как можно распечатывать информацию по частям.

## **2.1.2 Обоснование проектных решений по информационному обеспечению**

Информационное обеспечение библиотеки – это документный фонд на традиционных носителях информации и научно-образовательные электронные ресурсы [20].

Фонд библиотеки формируется в соответствии с задачами учебного и научно-исследовательского процессов и представлен учебниками, учебно-методическими, научными, справочными и периодическими издания, соответствующими по своему содержанию требованиям ФГОС и СУОС НИЯУ МИФИ.

Учебные дисциплины обеспечены основной и дополнительной литературой. Объем фонда на 01.10.2016 г. составляет 17561 экз., включая электронные ресурсы. В том числе:

- учебная литература 13347;
- учебно-методическая литература 1026;
- художественная литература 983;
- научно - популярная литература 1143;
- справочная литература 1062.

Комплектование фонда осуществляется в соответствии с задачами образовательного и научно-исследовательского процессов в учебном заведении. Фонды основной и дополнительной литературы формируются как за счёт учебной, так и учебно-методической литературы, методических пособий на бумажных носителях [23].

Библиотека оснащена современным, лицензионным программным обеспечением. Всего персональных компьютеров – 1. К услугам читателей небольшой читальный зал, в котором имеется возможность бесплатно пользоваться беспроводной сетью Интернет (Wi-Fi).

### **2.1.3 Обоснование проектных решений по программному обеспечению**

Программное обеспечение (ПО) включает совокупность компьютерных программ, описаний и инструкций по их применению на ЭВМ.

Все программное обеспечение принято делить на общее и специальное ПО.

В состав общего программного обеспечения входят операционные системы, оболочки, СУБД и т.д.

Специальное ПО включает совокупность прикладных программ, разработанных для решения конкретных задач.

Операционная система (ОС) в наибольшей степени определяет облик всей вычислительной системы в целом. ОС представляет собой самую важную часть программного обеспечения, так как образует среду выполнения для остальных программ [33].

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера, особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами.

От эффективности алгоритмов управления ресурсами компьютера во многом зависит эффективность ОС в целом. Поэтому, характеризуя ОС, часто приводят важнейшие особенности реализации функций ОС по управлению процессорами, памятью, внешними устройствами компьютера, процессами. Так, например, в зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на многозадачные и однозадачные, многопользовательские и однопользовательские, на многопроцессорные и однопроцессорные системы.

К факторам, определяющим выбор конкретной ОС относятся:

- требования к аппаратным средствам – IBM PC-совместимый компьютер;
- поддержка сетевой технологии, и в частности поддержка протоколов TCP/IP;
- надежность;
- быстродействие.

На автоматизируемом предприятии установлены следующие компоненты, необходимые для работы программы:

- ОС Microsoft Windows 7;
- Microsoft SQL Server Compact 4.0;
- Microsoft .NET Framework 4.

#### **2.1.4 Обоснование выбора программных средств**

В рамках данного проекта было принято решение разработать автоматизированную систему, включающую базу данных по учету книг и приложение по ее сопровождению. Учитывая возможности имеющегося на предприятии оборудования и программного обеспечения, необходимо создать современный программный продукт, избегая таких недостатков существующих коммерческих предложений, как высокая стоимость внедрения и сопровождения и слабая ориентированность на пользователя со слабой профессиональной подготовкой. Также необходимо уделить особое внимание надежности приложения и простоте его интерфейса.

Поэтому для разработки автоматизированной системы учета передвижения продукции была выбрана БД Microsoft SQL Server Compact 4.0.

Microsoft SQL Server Compact 4.0 — бесплатная встроенная БД, которую можно использовать для создания веб-сайтов ASP.NET и приложений для персональных компьютеров с ОС Windows [3]. SQL Server

Compact 4.0 занимает мало места и поддерживает частное развертывание исполняемого модуля в папке приложения, упрощает разработку в Visual Studio и WebMatrix и обеспечивает удобную миграцию схемы и данных в SQL Server [2].

Работать с Microsoft SQL Server Compact 4.0 можно не только в текстовом режиме, но и в графическом.

C# — объектно-ориентированный язык программирования.

C# относится к семье языков с C - подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java [27]. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Основные достоинства C#:

- C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности — как FCL, так и CLR;

- C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;

- C# является мощным объектным языком с возможностями наследования и универсализации;

- C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования [28]. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от C++ к C#;

- сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как указатели, адресация, разыменованное, адресная арифметика;

- благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;

- мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;

- реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

## 2.2 Описание базы данных

Разработанная программа может использоваться в любой библиотеке, она значительно облегчит работу ее сотрудников. Вся необходимая информация о книге, читателях, авторах, издательствах будет доступна всем библиотекарям. Они могут добавлять, удалять и изменять практически любые данные, а также узнать о наличии книги в библиотеке из базы данных [29].

В данной базе данных присутствуют следующие сущности: Книга, Автор, Жанр, Издательство, Изданная книга, Читатель, Журнал.

Одна книга может быть написана одним или несколькими авторами. Один автор может написать одну или несколько книг.

Изданная книга издается в одном издательстве. Издательство издает одну или несколько изданных книг.

Одна книга может относиться только к одной изданной книге. Одна изданная книга включает в себя одну или несколько книг.

Книга может быть одного или нескольких жанров. Один жанр может быть у одной или нескольких книг.

Один читатель может читать несколько книг. Одна книга может быть только у одного читателя.

Один автор может печататься в одном издательстве. Одно издательство может печатать работы нескольких авторов.

В базе данных присутствуют описанные ниже атрибуты:

Книга:

- Идентификационный код экземпляра;
- Название;
- Год издания;
- Количество страниц.

Автор:

- Фамилия;
- Имя;
- Отчество;
- Год рождения.

Издательство:

- Название;
- Контактные данные.

Изданная книга:

- Идентификационный код книги.

Жанр:

- Название;
- Короткое описание жанра.

Читатель:

- Фамилия;
- Имя;
- Отчество;
- Год рождения;

- Контактные данные.

Журнал:

- Время;

- ФИО;

- Книга;

- Операция.

Сущность – это некоторый объект, за которым пользователь хочет наблюдать и знать всю информацию о нем.

Сущность Author:

- Id – тип данных – int, наличие обязательно – идентификатор автора;

- Surname – тип данных – varchar, наличие обязательно – фамилия автора;

- Name – тип данных – varchar, наличие обязательно – имя автора;

- MidName – тип данных – varchar, наличие не обязательно – отчество автора;

- Year – тип данных – int, наличие не обязательно – год рождения автора.

Сущность Genre:

- Id – тип данных – int, наличие обязательно – идентификатор жанра;

- Title – тип данных – varchar, наличие обязательно – название жанра;

- Informatoin – тип данных – varchar, наличие не обязательно – информация о данном жанре.

Сущность Book:

- Code – тип данных – int, наличие обязательно – идентификатор книги;

- Title – тип данных – nchar, наличие обязательно – название книги;

- Year – тип данных – int, наличие не обязательно – год издания книги;

- PageCount – тип данных – int, наличие не обязательно – количество страниц в данной книге;

- IdReader – тип данных – int, наличие не обязательно – идентификатор читателя данной книги;

- ISBNBook – тип данных – int, наличие обязательно – ISBN данной книги.

#### Сущность Reader:

- Id – тип данных – int, наличие обязательно – идентификатор читателя;

- Surname – тип данных – varchar, наличие обязательно – фамилия читателя;

- Name – тип данных – varchar, наличие обязательно – имя читателя;

- MidName – тип данных – varchar, наличие не обязательно – отчество читателя;

- Year – тип данных – int, наличие не обязательно – год рождения читателя;

- Contact – тип данных varchar, наличие обязательно – контактные данные читателя.

#### Сущность PublishingHouse

- Id – тип данных – int, наличие обязательно – идентификатор издательства;

- Title – тип данных – varchar, наличие обязательно – название издательства;

- Contact – тип данных varchar, наличие не обязательно – контактные данные издательства.

#### Сущность PublishBook

- ISBN – тип данных – int, наличие обязательно – международный стандартный книжный номер;

- IdPublishingHouse – тип данных – int, наличие обязательно – идентификатор издательства.

### Сущность AuthorBook

- IdAuthor – тип данных – int, наличие обязательно – идентификатор автора;

- IdBook – тип данных – int, наличие обязательно – идентификатор книги.

### Сущность GenreBook

- IdGenre – тип данных – int, наличие обязательно – идентификатор жанра;

- IdBook – тип данных – int, наличие обязательно – идентификатор книги.

### Сущность Logs

- DateTime – тип данных – int, наличие обязательно – время на компьютере;

- IdReader – тип данных – int, наличие обязательно – идентификатор читателя;

- CodeBook – тип данных – int, наличие обязательно – идентификатор книги;

- Gave – тип данных – int, наличие обязательно – операция с книгой.

Реализация данной структуры описана в приложении А.

Логическая модель базы данных «Библиотека» представлена на рисунке 8.



Рисунок 8 – Логическая модель базы данных

Создание xml-файла включает описание каждого поля таблицы (свойства класса), а также связи, которые указываются в отображении связей, если есть связь «многие ко многим», необходимо указать дополнительную таблицу, через которую происходит эта связь.

Все классы представлены в приложении Б.

Для работы с базой данной необходимо использовать интерфейс `ISessionFactory` для получения интерфейса `ISession`, с помощью которого можно использовать методы `Get`, `Delete`, `Flush`, `BeginTransaction`, `SaveOrUpdate` и другие. Интерфейс `ISessionFactory` используется в начале работы с приложением, а `ISession` получаем тогда, когда необходимо. Создадим статистический класс `Helper`, с помощью которого будет получен `ISession` и в котором будет храниться `ISessionFactory` [25].

Для всех таблиц, кроме `PublishingBook`, имеется 4 кнопки – «Изменить», «Добавить», «Удалить» и «Дополнительно», для `PublishingBook` отсутствует кнопка «Дополнительно», т.к. для нее вся информация уже выведена.

XML — это легко читаемый язык иерархической текстовой разметки, который обычно применяется для обмена данными в системах и между системами.

Представление XSD-схемы находится в приложении В.

**Вывод:** В данном разделе была спроектирована информационная система, обоснованы все проектные решения, описана база данных. На основе полученных данных можно приступить к разработке автоматизированной системы.

### 3 Программная реализация проектных решений

#### 3.1 Программное обеспечение задачи

В разработанной программе предусмотрен только один пользователь – библиотекарь. Все исполняемые библиотекарем функции при работе с программой показаны в древе функций библиотекаря. Древо функций библиотекаря представлено на рисунке 9.

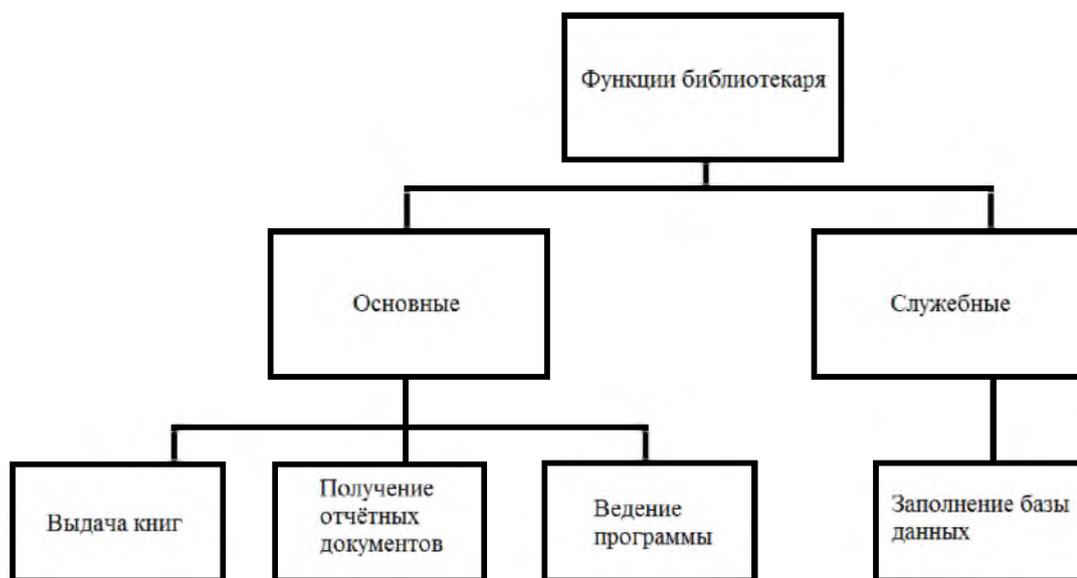


Рисунок 9 – Древо функций библиотекаря

На основании дерева функций разработан сценарий диалога, схематически представленный на рисунке 10.



Рисунок 10 – Сценарий диалога программы

### 3.2 Технологическое обеспечение задачи

Технологический процесс автоматизированной обработки информации представляет собой совокупность операций, выполняемых в определенной последовательности - от сбора исходной информации до получения результатной информации [19].

Под операцией понимается комплекс действий, выполняемых над информацией на рабочем месте.

Разработка технологического процесса должна обеспечить максимальную автоматизацию процессов обработки информации при использовании различных технологических средств и высокую достоверность получения результатной информации при минимальных трудовых и стоимостных затратах [40].

Состав операций и последовательность их выполнения зависят от характера решаемых задач и имеющегося комплекса технических средств на каждом уровне обработки.

Характер задач в основном определяется объемами обрабатываемой информации, периодичностью решения, а также сложностью алгоритмов ее преобразования. Решение задачи на каждом уровне обработки может выполняться разными способами с применением различных вычислительных средств и методов контроля [5].

При определении состава операций технологического процесса создается возможность выбора наиболее рационального способа обработки информации.

Технологический процесс начинается со сбора данных. Сбор данных осуществляется с бумажных носителей и текстового редактора Excel [1].

Входные данные заносят в программу вручную. Контроль вводимых данных осуществляется библиотекарем. Также существует контроль через информационные системы, который осуществляется автоматически.

Сортировку и корректировку данных осуществляет библиотекарь по мере поступления новых данных. Обязательно корректируются справочники с книгами, т.к. информация, содержащаяся в них, со временем устаревает [30].

Передача информации по каналам связи осуществляется с помощью электронной почты. Вся информация хранится в архиве на бумажных носителях.

Технологическая схема сбора, обработки и выдачи информации представлена на рисунке 11.

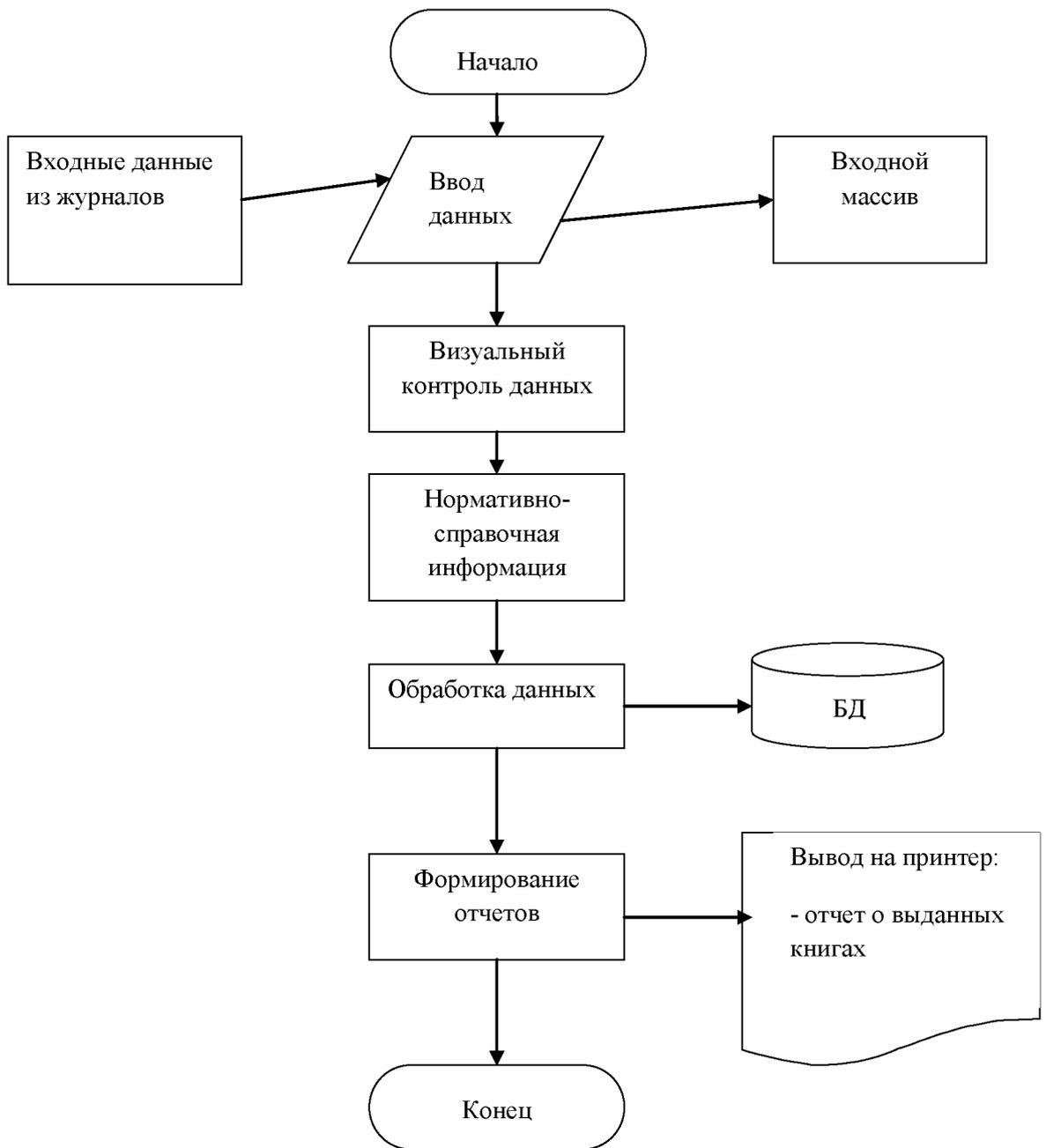


Рисунок 11 – Технологическая схема сбора, обработки и выдачи информации

### 3.3 Описание контрольного примера реализации проекта

После запуска программы перед пользователем отображается специальная главная форма, которая позволяет начать работу с программой.

Во вкладке «Книга» находится информация о книге: название, год издания, количество страниц, ISBN книги, читатель. Также предусмотрены дополнительные кнопки для добавления, изменения и удаления информации.

Интерфейс программы показан на рисунке 12.

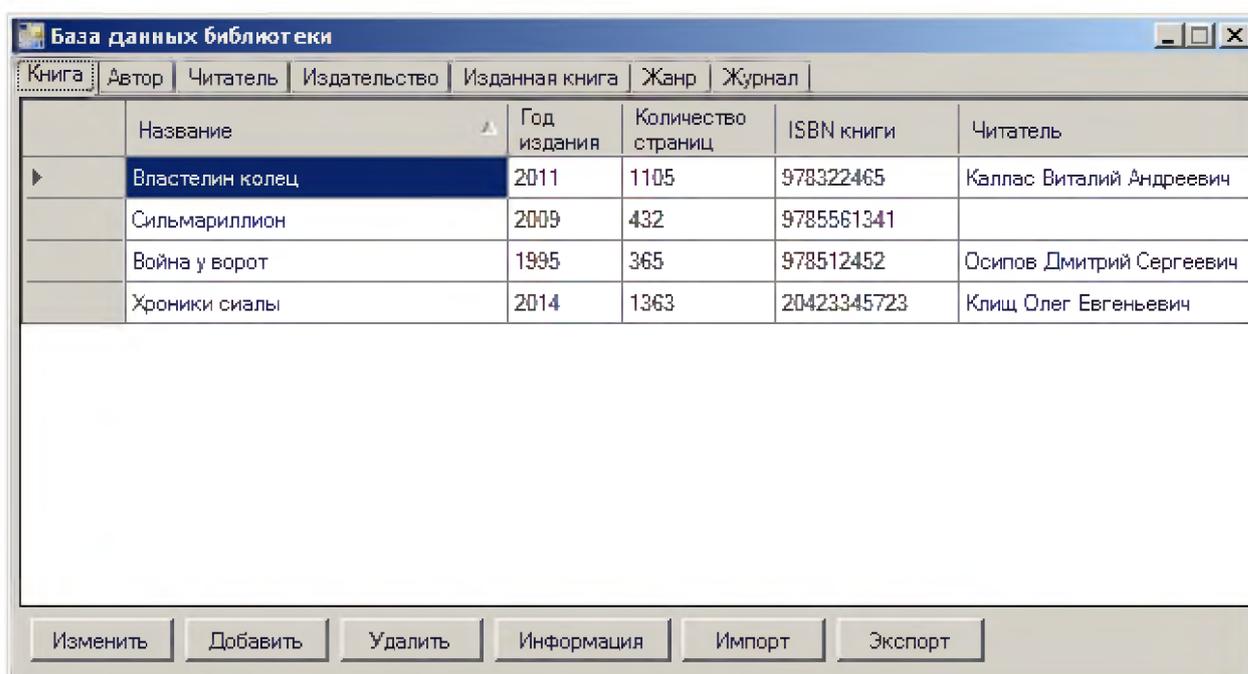


Рисунок 12 – Интерфейс программы

Ниже представлен код программы, при помощи которого в форму окна «Книга» были добавлены кнопки

```
// tabPageBook
//
this.tabPageBook.Controls.Add(this.ExportBook);
this.tabPageBook.Controls.Add(this.ImportBook);
this.tabPageBook.Controls.Add(this.buttonOtherBook);
```

```

this.tabPageBook.Controls.Add(this.buttonDeleteBook);
this.tabPageBook.Controls.Add(this.buttonAddBook);
this.tabPageBook.Controls.Add(this.buttonChangeBook);
this.tabPageBook.Controls.Add(this.dataGridViewBook);
this.tabPageBook.Location = new System.Drawing.Point(4, 22);
this.tabPageBook.Name = "tabPageBook";
this.tabPageBook.Padding = new System.Windows.Forms.Padding(3);
this.tabPageBook.Size = new System.Drawing.Size(646, 304);
this.tabPageBook.TabIndex = 0;
this.tabPageBook.Text = "Книга";
this.tabPageBook.UseVisualStyleBackColor = true;
this.tabPageBook.Enter+=new
System.EventHandler(this.tabPageBook_Enter);

```

Также программа поддерживает показ информации о жанрах книг с информацией о них. Кроме того, также показываются все книги, которым присвоен этот жанр. Пример показан на рисунке 13.

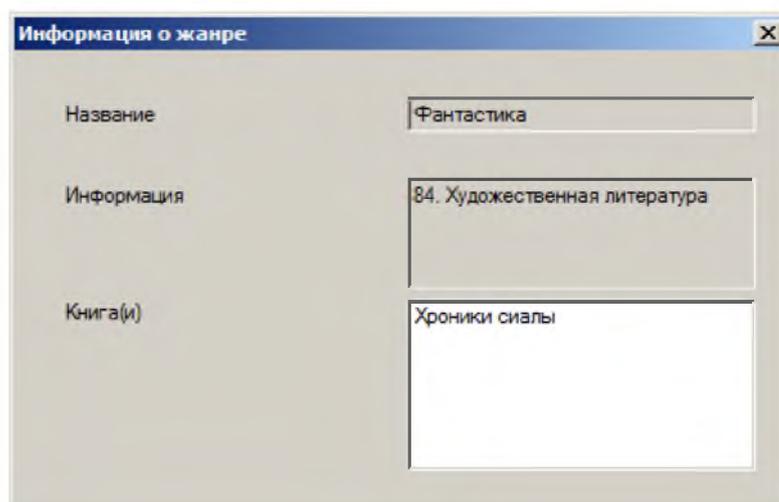


Рисунок 13 – Пример информации о жанре

На рисунке 14 показан пример регистрации книги в программе. Для этого заранее должны быть введены ISBN, автор или авторы и жанр.

Регистрация книги в базе данных

Название книги \* Хроники сиалы

Год издания 2014

Количество страниц 1363

ISBN \* 20423345723

Все авторы

Выбранные авторы \* Толкин Джон Рональд Руэл  
Пехов Алексей Григорьевич

Добавить автора Удалить автора

Все жанры

Выбранные жанры \* Фэнтези  
Фантастика

\* - обязательное поле

Добавить жанр Удалить жанр

Ok Отмена

Рисунок 14 – Пример регистрации книги

Программа поддерживает возможность создавать выборки по читателям в формате таблиц Excel. Для этого необходимо выделить необходимого читателя во вкладке «Читатель», а затем нажать кнопку «Создать анкету». Пример такой выборки показан на рисунке 15.

	А	В
1		
2	<b>Анкета читателя</b>	
3	Фамилия	Каллас
4	Имя	Виталий
5	Отчество	Андреевич
6	Год рождения	1995
7	Контактные данные	8-920-207-11-00
8		
9	<b>Книги читателя</b>	
10		
11	Властелин колец	
12		

Рисунок 15 – Пример выборки по читателю

На рисунке 16 показана вкладка «Изданная книга». В неё по нажатию кнопки «Добавить» вносится ISBN книги, который должен быть в строго заданном числовом формате и иметь не более 11 символов. Также должно быть заранее указано издательство книги. Если нарушены условия ввода данных, программа подсвечивает некорректные поля красным цветом, а также указывает надпись в нижней части окна.

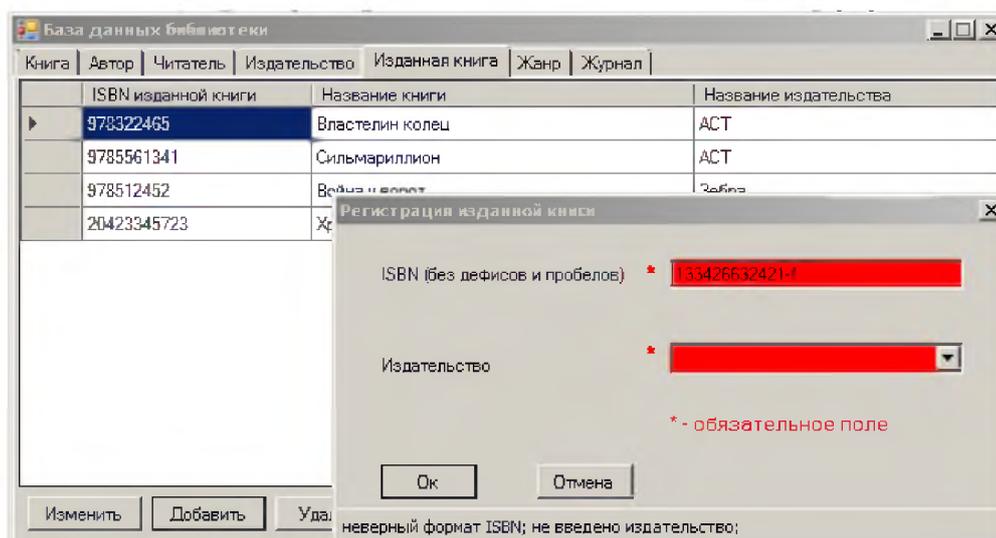


Рисунок 16 – Пример неправильного внесения ISBN книги

Также в программе ведётся учёт выданных и возвращённых книг. Эта информация отображается во вкладке «Журнал». Журнал позволяет увидеть,

когда и кому была произведена операция выдачи или возврата книги. Окно программы показано на рисунке 17.

Время	ФИО	Книга	Операция
15.06.2017 7:28:54	Осипов Дмитрий Сергеевич	Властелин колец	Возврат
15.06.2017 7:27:38	Каллас Виталий Андреевич	Сильмариллион	Возврат
15.06.2017 7:27:33	Каллас Виталий Андреевич	Сильмариллион	Выдача
15.06.2017 7:27:30	Осипов Дмитрий Сергеевич	Властелин колец	Выдача
15.06.2017 7:27:26	Осипов Дмитрий Сергеевич	Сильмариллион	Возврат
15.06.2017 7:27:23	Каллас Виталий Андреевич	Властелин колец	Возврат
15.06.2017 7:27:17	Осипов Дмитрий Сергеевич	Сильмариллион	Выдача
15.06.2017 7:27:13	Каллас Виталий Андреевич	Властелин колец	Выдача

Вывести отчет

Рисунок 17 – Вкладка программы «Журнал»

В программе также поддерживается вывод отчёта о движении книг библиотеки. Отчёт создаётся на основе данных вкладки «Журнал». Пример отчёта показан на рисунке 18.

	А	В	С	Д	Е
1	Журнал движения книг библиотеки				
2					
3	Время	Читатель	Книга	Операция	
4	11.05.2017 23:57:07	Каллас Виталий Андреевич	Властелин колец	Выдача	
5	12.05.2017 23:57:09	Осипов Дмитрий Сергеевич	Сильмариллион	Возврат	
6	14.05.2017 23:57:11	Каллас Виталий Андреевич	Властелин колец	Возврат	
7	14.05.2017 23:57:12	Осипов Дмитрий Сергеевич	Властелин колец	Выдача	
8	08.06.2017 0:12:41	Клищ Олег Евгеньевич	Евгений Онегин	Выдача	
9					
10			За последний месяц было выдано 3 книг(и)		
11			За последний месяц было возвращено 3 книг(и)		
12					

Рисунок 18 – Пример отчёта движения книг библиотеки

### **3.4 Обоснование экономической эффективности разработки**

#### **3.4.1 Выбор и обоснование показателей эффективности**

Основным критерием, который определяет необходимость автоматизированной обработки данных, является экономическая эффективность [7].

В связи с тем, что любое экономическое мероприятие требует оценки его экономической эффективности, необходимо провести оценку эффективности функционирования внедряемой программы. Для оценки эффективности автоматизированной информационной системы (АИС) может служить соотношение потребностей и затрат [12].

Экономическая эффективность рассчитывается для возможности соотношения полученного после внедрения АИС результата с произведенными затратами.

Конечной целью функционирования АИС является удовлетворение информационных потребностей конечного пользователя.

Внедрение ЭВМ для автоматизированной обработки информации обычно связано с определенными капитальными вложениями, однако при исполнении данной работы вложений не потребовалось, поскольку:

- Вычислительная техника на предприятии располагает достаточными вычислительными возможностями;
- Подготовительные работы не требуют затрат, так как все необходимые средства для работы программы являются бесплатными, либо включены в базовую версию ОС;
- Персонал библиотеки (библиотекарь) обучен в достаточной степени, чтобы самостоятельно выполнять работу в разработанной программе.

### 3.4.2 Расчет показателей эффективности

Капитальные затраты на обработку системы.

Рассчитаем эксплуатационные расходы до разработки и внедрения ИС.

Капитальные затраты не производились.

Исходные данные по заработной плате персонала, расценкам и нормативным коэффициентам приведены в таблице 1.

Таблица 1 – Исходные данные

№ п/п	Наименование показателей	Усл. обозначение	Ед. измерения	Значения показателей	
				до ИС	с ИС
1	Коэффициент отчислений	F	%	38,5	38,5
2	З/п Зав. Библиотекой	31	руб.	10000,00	12000,00
3	З/п библиотекаря	32	руб.	8500,00	-
4	Стоимость электроэнергии		руб.	2	2
5	Норм. Коэффициент приведения затрат к единому году	$E_n$	-	-	0,25
6	З/п программиста	33	руб.	-	4500,00
7	Время на разработку	T	мес.	-	6

За один месяц потребление электроэнергии – 300 кВт/ч.

$$P_{\text{эп}} = 300 * 6 * 2 = 3600,00 \text{ руб.}$$

Расходы на материалы за шесть месяцев составили:

$$P_{\text{м}} = 5400,00 \text{ руб.}$$

Расходы на ремонт:

$$P_{\text{рем}} = 2100,00 \text{ руб.}$$

$$P_{\text{зап}} = 1500,00 \text{ руб.}$$

Поскольку  $P_{\text{общ}} - P_{\text{пр}} = 3600,00 + 5400,00 + 2100,00 + 1500,00 = 12600,00$  руб., то

$P_{\text{пр}} = 0,6 * 7120,00 = 7560,00$  руб., следовательно,

$P_{\text{общ}} = 12600,00 + 7560,00 = 20160,00$  руб.

Расходы на материалы до внедрения ИС приведены в таблице 2.

Таблица 2 – Расходы на материалы до внедрения ИС

Материалы	Ед. измерения	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	лист	500	2	1000,00
Канцтовары	-		-	400,00
Картридж для принтера	шт.	1	1200,00	1200,00
Гибкий накопитель (3.5")	шт.	5	25,00	125,00
Итого				2725,00

До разработки ИС работало 2 библиотекаря. Расходы до разработки ИС ( $P_0$ ). Расчетный период – 6 месяцев.

$P_{\text{осн}} = (12000,00 + 8500,00) * 6 * 1,385 = 170355,00$  руб.

$P_{\text{ауп}} = 0,25 * 170355,00 = 42588,75$  руб.

Таким образом, эксплуатационные расходы на решение задачи до внедрения системы составляют:

$P = 170355,00 + 42588,75 + 20160,00 = 233103,75$  руб.

Теперь рассчитаем приведенные затраты после разработки и внедрения ИС.

Капитальные (единовременные) затраты:

Разработкой системы занимается 1 программист, месячная заработная плата – 4500,00 руб. Фактическое время, затраченное на разработку системы – 6 месяцев. Поскольку разработка программного обеспечения осуществляется на компьютере, ранее установленном для решения других задач, затраты на основное оборудование в капитальные затраты не включаются. Следовательно, капитальные затраты будут равны затратам на проектирование [10].

$K = K_{\text{пр}}$

$K_{\text{пр}} = 4500,00 * 6 * 1,385 + D_{\text{пр}}$

$$K = 37395,00 \text{ руб.}$$

Рассчитаем годовые эксплуатационные расходы на решение задачи после внедрения информационной системы.

Расходы на электроэнергию – 300 кВт/ч.

$$P_{\text{эн}} = 400 * 6 * 2 = 4800,00 \text{ руб.}$$

В таблице 3 приведены расходы на материалы после внедрения ИС.

Таблица 3 – Расходы на материалы после внедрения ИС

Материалы	Ед. измерения	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	лист	300	2	600,00
Канцтовары	-		-	20,00
Картридж для принтера	шт.	2	1200,00	2400,00
Гибкий накопитель (3.5“)	шт.	2	25,00	50,00
Итого				3070,00

Расходы на материалы за шесть месяцев составили:

$$P_{\text{м}} = 7000,00 \text{ руб.}$$

Расходы на ремонт (цена договорная):

$$P_{\text{рем}} = 2500,00 \text{ руб.}$$

$$P_{\text{зап}} = 2000,00 \text{ руб.}$$

$$P_{\text{общ}} + P_{\text{пр}} = 4800,00 + 7000,00 + 2500,00 + 2000,00 = 16300,00 \text{ руб.}$$

$$P_{\text{пр}} = 0,6 * 8080,00 = 9780,00 \text{ руб.}$$

$$P_{\text{общ}} = 16300,00 + 9780,00 = 26080,00 \text{ руб.}$$

После разработки и внедрения ИС остался работать 1 библиотекарь.

Таким образом, расходы за расчетный период (6 месяцев) после разработки и внедрения ИС составили:

$$P_{\text{осн}} = 12000 * 6 * 1,385 = 99720,00 \text{ руб.}$$

$$P_{\text{ауп}} = 0,25 * 99720,00 = 24930,00 \text{ руб.}$$

$$P = 99720,00 + 24930,00 + 26080,00 = 150730,00 \text{ руб.}$$

$$P_1 = 150730,00 \text{ руб.}$$

Приведенные затраты:

$$З = 150730,00 + 0,25 * 37395,00 = 160078,75$$

$$\text{Условный экономический эффект} = 233103,75 -$$

$$150730,00 = 82373,75 \text{ руб.}$$

Срок окупаемости:

$$T_p = 37395,00 / 82373,75 = 0,45$$

Расчетный коэффициент:

$$E_p = 82373,75 / 37395,00 = 2,2$$

Результаты расчета представлены в таблице 4.

Таблица 4 – Выходные данные

№ п/п	Наименование показателей	Усл. обозн.	Ед. измер.	Значение показателей
1	Приведенные затраты	З	руб.	160078,75
2	Расходы до внедрения ИС	P <sub>0</sub>	руб.	233103,75
3	Расходы после внедрения ИС	P <sub>1</sub>	руб.	150730,00
4	Экономический эффект	Э	руб.	82373,75
5	Срок окупаемости	T <sub>p</sub>		0,45
6	Расчетный коэффициент	E <sub>p</sub>		2,2

Расчеты показали, что произошло значительное сокращение затрат.

Получен условный экономический эффект в размере 82373,75 руб.

При внедрении ИС сократилось число сотрудников. Снизилась расходы на заработную плату. Срок окупаемости составляет 3 месяца. Расчетный коэффициент 2,2, что доказывает целесообразность разработки и внедрения ИС.

## ЗАКЛЮЧЕНИЕ

В данной работе выполнена автоматизация учёта книг в школьной библиотеке. Цель работы достигнута в полном объёме. Также были выполнены следующие задачи:

- Изучены теоретические и практические аспекты учёта книг в школьных библиотеках;
- Спроектирована система учёта книг в библиотеке;
- Разработана система учёта книг в библиотеке;
- Расчитана экономическая эффективность системы.

Автоматизация учета в образовательном учреждении дает возможность более оперативно выбрать тактику подготовки и объективно формировать стратегию развития важных качеств учеников. С помощью автоматизации учета формируется отлаженный механизм получения многосторонней и качественной информации, представляющей возможность комплексно рассмотреть вопросы планирования, оперативного контроля и учета.

Внедрение на предприятии автоматизированных систем учета предоставляет возможность сформировать в автоматизированном режиме, в соответствии с информационными запросами субъекта управления, многосторонние сведения об учёте фондов учреждения и перспективах его развития. Также для успешного решения вопросов проектирования и внедрения этих систем должна быть организована соответствующая подготовка работников предприятия [38].

В данной ВКР главной задачей являлся учет книг в библиотеке Разуменской средней общеобразовательной школы №2. При решении этой задачи самым трудным процессом оказался сбор информации, т.к. почти вся информация по книгам собирается вручную из фонда. Большой объем информации невозможно обработать вручную, поэтому автоматизация этой задачи является важным процессом.

В первой главе были подробно изучены процессы работы библиотеки, и на их основе строилась дальнейшая работа по созданию автоматизированной информационной системы.

Разработанная в данной работе система поддерживает все основные современные информационные системы, а также не требует установки платных продуктов. Программа имеет интуитивно понятный интерфейс, а также подсказки по вводу данных, из чего можно сделать вывод, что любой сотрудник, справляющийся с Microsoft Office Word, может работать с программой.

Как показало обоснование экономической эффективности разработки, данная система помогает сильно сократить затраты, что подтверждает её обоснованность в экономическом плане.

Данная работа может быть использована в дальнейшем для модернизации автоматизированной системы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт, К. Дж. Введение в системы баз данных, 7-е издание. [Текст]: Пер. с англ. – М.: Издательский дом «Вильямс», 2013. – 1072 с.
2. Хотек, М. Microsoft SQL Server 2008. Реализация и обслуживание. Учебный курс Microsoft [Текст] / Пер. С англ. – М. : Издательство «Русская Редакция», 2012. - 576 с. : ил.
3. Тернстрем, Т. Microsoft SQL Server 2008. Разработка баз данных. Учебный курс Microsoft 70-433 [Текст] (2010).– М: Русская Редакция, 2012. - 496 с.
4. Лобел, Л. Разработка приложений на основе Microsoft SQL Server 2014. [Текст] : Пер. с англ. – СПб: БХВ-Петербург, 2015. – 1024 с.
5. Филина, Ф. Н. Инвентаризация. Бухгалтерский учет и налогообложение [Текст] / Ф.Н. Филина. - М.: ГроссМедиа Ферлаг, РОСБУХ, 2015. - 264 с.
6. План счетов бухгалтерского учета бюджетных учреждений. [Текст] - М.: Проспект, 2015. - 232 с.
7. Федеральный закон Российской Федерации «О бухгалтерском учете» от 21.11.96, № 129-ФЗ: принят ГД ФС РФ 23.02.1996, [Текст] (ред. от 28.09.2013)
8. Методические указания по инвентаризации имущества и финан-совых обязательств. [Текст] Утверждены приказом Минфина РФ от 13.06.95 г. № 49.
9. Агеева, О.А. Бухгалтерский учет и анализ: Учебник для академического бакалавриата [Текст] / О.А. Агеева, Л.С. Шахматова. - Люберцы: Юрайт, 2016. - 509 с.
10. Астахов, В.П. Бухгалтерский (финансовый) учет в 2 томах: Учебник для академического бакалавриата [Текст] / В.П. Астахов. - Люберцы: Юрайт, 2015. - 922 с.

11. Бабаев, Ю.А. Бухгалтерский учет: Учебник для бакалавров [Текст] / Ю.А. Бабаев, А.М. Петров, Л.А. Мельникова. - М.: Проспект, 2016. - 424 с.
12. Бахолдина, И.В. Бухгалтерский финансовый учет: Учебное пособие [Текст] / И.В. Бахолдина, Н.И. Голышева. - М.: Форум, НИЦ ИНФРА-М, 2013. - 320 с.
13. Шамрай, Д. Д. Архивное наследие Российской национальной библиотеки. Выпуск 1. Избранные статьи по библиотечному делу. [Текст] - Москва: Машиностроение, 2015. - 112 с.
14. ГОСТ. Стандарты по библиотечному делу и библиографии. [Текст] - М.: Стандартгов, 2016. - 280 с.
15. Глазков, М. Н. Библиотечное дело в России (октябрь 1929 - май 1941. Документы и материалы. В 2 частях. Часть 2 [Текст] / М.Н. Глазков. - М.: Пашков дом, 2013. - 352 с.
16. Голубенко, Н. Б. Библиотечное дело. Инновации и перспективы [Текст] / Н.Б. Голубенко. - Москва: Огни, 2016. - 132 с.
17. Коготков, Д.Я. Библиографическая деятельность библиотеки: организация, технология, управление: учебник [Текст] / Д.Я. Коготков. - СПб. : Профессия, 2012.
18. Труды по библиотековедению. Руководство для общественных библиотек.: Практ. пособие. [Текст] - М.: Либерия, - 2011.
19. Баженов, Ю. К. Менеджмент организации: методические рекомендации по подготовке и защите выпускных квалификационных работ [Текст] / Ю. К. Баженов, В. А. Пронько. - М.: "Дашков и К", 2014. - 168 с.
20. Безрукова, В.С. Педагогика: Учебное пособие [Текст] / В.С. Безрукова. - Рн/Д: Феникс, 2013. - 381 с.
21. Воронов, В.В. Педагогика школы: новый стандарт [Текст] / В.В. Воронов. - М.: ПО России, 2012. - 288 с.
22. Джурицкий А.Н. Сравнительная педагогика : Учебник для бакалавриата и магистратуры [Текст] / А.Н. Джурицкий. - Люберцы: Юрайт, 2016. - 440 с.

23. Захаренко М.П. Молодые кадры современной библиотеки [Текст]: организационно - управленческий подход педагогика / М.П. Захаренко. - СПб.: Профессия, 2013. - 240 с.
24. Пронькина С.Н. Информационная система электронной библиотеки [Текст] / С.Н. Пронькина, А.А. Силаев // Молодой учёный. - 2014. - № 6 (65, май-1), ч. 1. -117 с.
25. Интегрированная среда разработки Visual Studio // MSDN home. URL: <https://msdn.microsoft.com/ru-ru/library/dn762121.aspx> (дата обращения: 10.06.2017).
26. Справочное руководство по MySQL // MySQL. URL: <http://www.mysql.ru/docs/man/Introduction.html> (дата обращения: 10.06.2017).
27. Березин Б.И., Березин С.Б. Начальный курс С и С++. [Текст] — М.: ДИАЛОГ-МИФИ, 2012. -416 с.
28. Керниган Б., Ритчи Д. Язык программирования Си: Пер. с англ. [Текст] — М.: Финансы и статистика, 2013. -737 с.
29. Агальцов, В.П. Базы данных. В 2-х т.Т. 1. Локальные базы данных: Учебник [Текст] / В.П. Агальцов. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 352 с.
30. Кириллов, В.В. Введение в реляционные базы данных. Введение в реляционные базы данных [Текст] / В.В. Кириллов, Г.Ю. Громов. - СПб.: БХВ-Петербург, 2012. - 464 с.
31. Кузин, А.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений [Текст] / А.В. Кузин, С.В. Левонисова. - М.: ИЦ Академия, 2012. - 320 с.
32. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования [Текст] / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: ИЦ Академия, 2012. - 320 с.
33. Алексеев, Г.В. Компьютерные технологии при проектировании и эксплуатации технологического оборудования: Учебное пособие [Текст] /

Г.В. Алексеев, И.И. Бриденко, В.А. Головацкий. - СПб.: ГИОРД, 2012. - 256 с.

34. Соловьева, Л.Ф. Компьютерные технологии для преподавателя [Текст] / Л.Ф. Соловьева. - СПб.: ВHV, 2008. - 464 с.

35. Афонин, А.М. Теоретические основы разработки и моделирования систем автоматизации: Учебное пособие [Текст] / А.М. Афонин, Ю.Н. Царегородцев, А.М. Петрова, Ю.Е. Ефремова. - М.: Форум, 2011. - 192 с.

36. Пантелеев, В.Н. Основы автоматизации производства: Учебник для учреждений начального профессионального образования [Текст] / В.Н. Пантелеев, В.М. Прошин. - М.: ИЦ Академия, 2013. - 208 с.

37. Друри, К. Управленческий и производственный учет: Учебный комплекс для студентов вузов [Текст] / К. Друри; Пер. с англ. В.Н. Егоров. - М.: ЮНИТИ-ДАНА, 2013. - 1423 с.

38. Лихтенштейн, В.Е. Стандартизация и разработка программных систем: Учебное пособие [Текст] / В.Е. Лихтенштейн. - М.: Финансы и статистика, 2010. - 288 с.

39. Гагарина, Л.Г. Разработка и эксплуатация автоматизированных информационных систем. ГРИФ [Текст] / Л.Г. Гагарина. - Вологда: Инфра-Инженерия, 2016. - 384 с.

40. Маркин, А.В. Разработка отчетов в информационных системах: Учебное пособие [Текст] / А.В. Маркин. - М.: Диалог-МИФИ, 2012. - 312 с.

## ПРИЛОЖЕНИЕ А

### «SQL-запросы для создания таблицы БД»

```
CREATE DATABASE Library  
GO
```

```
USE Library  
Go
```

```
CREATE TABLE PublishingHouse  
  (Id int PRIMARY KEY IDENTITY(1, 1),  
   Title varchar(50) NOT NULL CHECK(Title <> ""),  
   Contact varchar(100) NULL)
```

```
CREATE TABLE PublishingBook  
  (ISBN int PRIMARY KEY,  
   IdPublishingHouse int FOREIGN KEY REFERENCES  
   PublishingHouse(Id) ON DELETE NO ACTION ON UPDATE CASCADE  
   NOT NULL)
```

```
CREATE TABLE Reader  
  (Id int PRIMARY KEY IDENTITY(1, 1),  
   Surname varchar(50) NOT NULL CHECK(Surname <> ""),  
   Name varchar(50) NOT NULL CHECK(Name <> ""),  
   MidName varchar(50) NULL CHECK(Midname <> ""),  
   [Year] int NULL, Contact varchar(500) NOT NULL)
```

```
CREATE TABLE Book  
  (Code int PRIMARY KEY IDENTITY (1, 1),  
   Title varchar(50) NOT NULL CHECK(Title <> ""),  
   [Year] int NULL, [PageCount] int NULL,  
   IdReader int FOREIGN KEY REFERENCES Reader(Id) ON DELETE NO  
   ACTION ON UPDATE  
   CASCADE NULL,  
   ISBNBook int FOREIGN KEY REFERENCES PublishingBook(ISBN) ON  
   DELETE NO ACTION  
   ON UPDATE CASCADE NOT NULL)
```

```
CREATE TABLE Gerne  
  (Id int PRIMARY KEY IDENTITY(1, 1),  
   Title varchar(50) NOT NULL CHECK(Title <> ""),
```

Information varchar(100) NULL)

CREATE TABLE Author

(Id int PRIMARY KEY IDENTITY(1, 1),  
Surname varchar(50) NOT NULL CHECK(Surname <> ""),  
Name varchar(50) NOT NULL CHECK(Name <> ""),  
MidName varchar(50) NULL CHECK(Midname <> ""),  
[Year] int NULL)

CREATE TABLE AuthorBook

(IdAuthor int NOT NULL REFERENCES Author(Id) ON DELETE NO  
ACTION ON UPDATE  
CASCADE,  
IdBook int NOT NULL REFERENCES Book(Code) ON DELETE  
CASCADE ON UPDATE CASCADE  
PRIMARY KEY(IdAuthor, IdBook))

CREATE TABLE GerneBook

(IdGerne int NOT NULL REFERENCES Gerne(Id) ON DELETE NO  
ACTION ON UPDATE  
CASCADE,  
IdBook int NOT NULL REFERENCES Book(Code) ON DELETE  
CASCADE ON UPDATE CASCADE  
PRIMARY KEY(IdGerne, IdBook))

## ПРИЛОЖЕНИЕ Б

### «Классы для представления таблиц БД и описание их отображения»

#### Класс «Автор»

```
using System.Collections.Generic;

namespace LibDB
{
    public class Author
    {
        public virtual int Id { get; set; }
        public virtual string Surname { get; set; }
        public virtual string Name { get; set; }
        public virtual string MidName { get; set; }
        public virtual int? Year { get; set; }
        public virtual IList<Book> Books { get; set; }
        public virtual string FullName
        {
            get
            {
                return Surname + " " + Name + " " + MidName;
            }
        }
    }
}

<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
    namespace="LibDB"
    assembly="LibDB">
    <class name="Author" table="Author" lazy="true">
        <id name="Id">
            <column name="Id" />
            <generator class="increment" />
        </id>
        <property name="Surname" not-null="true"/>
        <property name="Name" not-null="true"/>
        <property name="MidName" />
        <property name="Year" />
    </class>
</hibernate-mapping>
```

```

    <bag name="Books" table="AuthorBook" inverse="true">
      <key column="IdAuthor"/>
      <many-to-many column="IdBook" class="Book" />
    </bag>
  </class>
</hibernate-mapping>

```

## Класс «Книга»

```
using System.Collections.Generic;
```

```

namespace LibDB
{
    public class Book
    {
        public Book()
        {
            Authors = new HashSet<Author>();
            Gernes = new HashSet<Gerne>();
        }

        public virtual int Code { get; set; }
        public virtual string Title { get; set; }
        public virtual int? Year { get; set; }
        public virtual int? PageCount { get; set; }
        public virtual Reader Reader { get; set; }
        public virtual PublishingBook PublishingBook { get; set; }
        public virtual ISet<Author> Authors { get; set; }
        public virtual ISet<Gerne> Gernes { get; set; }
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
    namespace="LibDB"
    assembly="LibDB">
  <class name="Book" table="Book" lazy="true">
    <id name="Code">
      <column name="Code" />
      <generator class="increment" />
    </id>
    <property name="Title" not-null="true"/>
    <property name="Year" />
    <property name="PageCount" />

```

```

<many-to-one name="Reader"
  class="Reader"
  column="IdReader"
  cascade="save-update"/>
<many-to-one name="PublishingBook"
  class="PublishingBook"
  column="ISBNBook"
  cascade="save-update"
  not-null="true"/>
<set name="Authors" table="AuthorBook">
  <key column="IdBook" />
  <many-to-many column="IdAuthor" class="Author"/>
</set>
<set name="Gernes" table="GerneBook">
  <key column="IdBook" />
  <many-to-many column="IdGerne" class="Gerne"/>
</set>
</class>
</hibernate-mapping>

```

### Класс «Жанр»

```
using System.Collections.Generic;
```

```
namespace LibDB
```

```
{
  public class Gerne
  {
    public virtual int Id { get; set; }
    public virtual string Title { get; set; }
    public virtual string Information { get; set; }
    public virtual IList<Book> Books { get; set; }
  }
}
```

```

<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
  namespace="LibDB"
  assembly="LibDB">
  <class name="Gerne" table="Gerne" lazy="true">
    <id name="Id">
      <column name="Id" />
      <generator class="increment" />
    </id>

```

```

    <property name="Title" not-null="true"/>
    <property name="Information" />
    <bag name="Books" table="GerneBook" inverse="true">
        <key column="IdGerne"/>
        <many-to-many column="IdBook" class="Book" />
    </bag>
</class>
</hibernate-mapping>

```

### Класс «Изданная книга»

```
using System.Collections.Generic;
```

```
namespace LibDB
```

```

{
    public class PublishingBook
    {
        public virtual int Id { get; set; }
        public virtual long ISBN { get; set; }
        public virtual PublishingHouse PublishingHouse { get; set; }
        public virtual IList<Book> Books { get; set; }
    }
}

```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
```

```
    namespace="LibDB"
```

```
    assembly="LibDB">
```

```
<class name="PublishingBook" table="PublishingBook" lazy="true">
```

```
<id name="Id">
```

```
    <column name="Id" />
```

```
    <generator class="increment" />
```

```
</id>
```

```
<property name="ISBN" not-null="true"/>
```

```
<many-to-one name="PublishingHouse"
```

```
    class="PublishingHouse"
```

```
    column="IdPublishingHouse"
```

```
    cascade="save-update"
```

```
    not-null="true"/>
```

```
<bag name="Books" cascade="save-update" lazy="true">
```

```
    <key column="ISBNBook" />
```

```
    <one-to-many class="Book" />
```

```
</bag>
```

```
</class>
```

```
</hibernate-mapping>
```

## Класс «Издательство»

```
using System.Collections.Generic;

namespace LibDB
{
    public class PublishingHouse
    {
        public virtual int Id { get; set; }
        public virtual string Title { get; set; }
        public virtual string Contact { get; set; }
        public virtual IList<PublishingBook> PublishingBooks { get; set; }
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
    namespace="LibDB"
    assembly="LibDB">
    <class name="PublishingHouse" table="PublishingHouse" lazy="true">
        <id name="Id">
            <column name="Id" />
            <generator class="increment" />
        </id>
        <property name="Title" not-null="true"/>
        <property name="Contact" />
        <bag name="PublishingBooks" cascade="save-update" lazy="true">
            <key column="IdPublishingHouse" />
            <one-to-many class="PublishingBook" />
        </bag>
    </class>
</hibernate-mapping>
```

## Класс «Читатель»

```
using System.Collections.Generic;

namespace LibDB
{
    public class Reader
    {
        public Reader()
        {
            Books = new List<Book>();
        }
    }
}
```

```

    }
    public virtual int Id { get; set; }
    public virtual string Surname { get; set; }
    public virtual string Name { get; set; }
    public virtual string MidName { get; set; }
    public virtual int? Year { get; set; }
    public virtual string Contact { get; set; }
    public virtual IList<Book> Books { get; set; }
    public virtual string Class { get; set; }
    public virtual string FullName
    {
        get
        {
            return Surname + " " + Name + " " + MidName;
        }
    }
}
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping xmlns="urn:hibernate-mapping-2.2"
    namespace="LibDB"
    assembly="LibDB">
<class name="Reader" table="Reader" lazy="true">
    <id name="Id">
        <column name="Id" />
        <generator class="increment" />
    </id>
    <property name="Surname" not-null="true"/>
    <property name="Name" not-null="true"/>
    <property name="MidName" />
    <property name="Class" />
    <property name="Year" />
    <property name="Contact" not-null="true" />
    <bag name="Books" cascade="save-update" lazy="true">
        <key column="IdReader" />
        <one-to-many class="Book" />
    </bag>
</class>
</hibernate-mapping>

```

### **Класс «Журнал»**

```
using System.Collections.Generic;
```

```

namespace LibDB
{
    public class Logs
    {
        public virtual long DateTime { get; set; }
        public virtual Reader Reader { get; set; }
        public virtual Book Book { get; set; }
        public virtual bool Gave { get; set; }
    }
}

<?xml version="1.0" encoding="utf-8" ?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
    namespace="LibDB"
    assembly="LibDB">
    <class name="Logs" table="Logs" lazy="true">
        <id name="DateTime">
            <column name="DateTime" />
            <generator class="assigned" />
        </id>
        <many-to-one name="Reader"
            class="Reader"
            column="IdReader"
            cascade="save-update"/>
        <many-to-one name="Book"
            class="Book"
            column="CodeBook"
            cascade="save-update"/>
        <property name="Gave" />
    </class>
</hibernate-mapping>

```

## ПРИЛОЖЕНИЕ В

### «Импорт и экспорт данных БД»

#### XSD-схема для импорта данных:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="GUID">
    <xs:restriction base="xs:string">
      <xs:pattern value="([0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
        [0-9a-fA-F]{4}-[0-9a-fA-F]{12})|(\{[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-
        [0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\})"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Library">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Books">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Book" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Title">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:minLength value="1"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="Year" minOccurs="0" type="xs:int"/>
                    <xs:element name="PageCount" minOccurs="0" type="xs:int"/>
                    <xs:element name="ISBNBook" type="xs:int"/>
                    <xs:element name="Gernes">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Gerne" minOccurs="0"
                            maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Title">
```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Information" minOccurs="0"
                type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="BookAuthors">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="BookAuthor" maxOccurs="unbounded">
                <xs:simpleType>
                    <xs:restriction base="GUID"/>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Authors">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Author" maxOccurs="unbounded" minOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Surname">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:minLength value="1"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

    <xs:element name="Name">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="MidName" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Year" minOccurs="0" type="xs:int"/>
    <xs:element name="Id">
      <xs:simpleType>
        <xs:restriction base="GUID"/>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

### Пример XML-файла:

```

<?xml version="1.0" encoding="UTF-8"?>
<Library>
  <Books>
    <Book>
      <Title>"Война и мир"</Title>
      <Year>2000</Year>
      <PageCount>1000</PageCount>
      <ISBNBook>1</ISBNBook>
      <Gernes>
        <Gerne>
          <Title>Роман</Title>
          <Information>паваппаап</Information>
        </Gerne>
      </Gernes>
    </Book>
  </Books>
</Library>

```

```
</Gernes>
<BookAuthors>
  <BookAuthor>c17db3ff-4d6b-45f6-8c5e-3c57d3695227</BookAuthor>
  <BookAuthor>b76a2828-266c-4c8f-9e62-f7e5e1b312b4</BookAuthor>
</BookAuthors>
</Book>
```

```
<Book>
  <Title>"Зима"</Title>
  <Year>1791</Year>
  <PageCount>3000</PageCount>
  <ISBNBook>3</ISBNBook>
  <Gernes>
    <Gerne>
      <Title>Роман</Title>
      <Information>паваппаап</Information>
    </Gerne>
    <Gerne>
      <Title>Повесть</Title>
      <Information>
        </Information>
    </Gerne>
  </Gernes>
  <BookAuthors>
    <BookAuthor>c17db3ff-4d6b-45f6-8c5e-3c57d3695227</BookAuthor>
    <BookAuthor>e2416384-f930-4360-8895-799a05d221ef</BookAuthor>
    <BookAuthor>05ceca1a-cfea-483e-a014-4b314da8a19c</BookAuthor>
  </BookAuthors>
</Book>
```

```
</Books>
```

```
<Authors>
```

```
<Author>
```

```
<Surname>Толстой</Surname>
```

```
<Name>Лев</Name>
```

```
<MidName>Николаевич</MidName>
```

```
<Year>1780</Year>
```

```
<Id>c17db3ff-4d6b-45f6-8c5e-3c57d3695227</Id>
```

```
</Author>
```

```
<Author>
```

```
<Surname>Замятин</Surname>
```

```
<Name>Евгений</Name>
```

```
<MidName>Алексеевич</MidName>
```

```
<Id>b76a2828-266c-4c8f-9e62-f7e5e1b312b4</Id>
```

```
</Author>
```

```

<Author>
  <Surname>Пушкин</Surname>
  <Name>Александр</Name>
  <MidName>Сергеевич</MidName>
  <Year>1799</Year>
  <Id>e2416384-f930-4360-8895-799a05d221ef</Id>
</Author>
<Author>
  <Surname>Шолохов</Surname>
  <Name>Михаил</Name>
  <MidName>Владимирович</MidName>
  <Id>05сеса1а-сфеа-483е-а014-4b314da8a19с</Id>
</Author>
</Authors>
</Library>

```

## Импорт

```

private void ImportBook_Click(object sender, EventArgs e)
{
  XmlDocument doc = new XmlDocument();
  if (openFileDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
  {
    doc.Load(openFileDialog.FileName);
    ErrorMessage = "";
    doc.Schemas.Add(null, "XMLSchema.xsd");
    doc.Validate(ValidationHandler, doc);
    if (ErrorMessage.Length == 0)
    {
      XmlNodeList books =
        doc.DocumentElement.SelectSingleNode("Books").ChildNodes
        ;
      using (ISession ses = Helper.GetCurrentSession())
      {
        Dictionary<Guid, List<Book>> dictAuthor = new Dictionary<Guid,
          List<Book>>();
        foreach (XmlNode book in books)
        {
          Book b = new Book();
          b.Title = book.SelectSingleNode("Title").InnerText;
          b.Year = null;
          if (book.SelectSingleNode("Year") != null)
            b.Year =
              Convert.ToInt32(book.SelectSingleNode("Year").InnerText);

```

```

    b.PageCount = null;
    if (book.SelectSingleNode("PageCount") != null)
        b.PageCount = Convert.ToInt32(book.
            SelectSingleNode("PageCount").InnerText);
    PublishingBook tmp =
ses.Get<PublishingBook>(Convert.ToInt32(book.
            SelectSingleNode("ISBNBook").InnerText));
    if (tmp != null)
        b.PublishingBook = tmp;
    else
    {
        MessageBox.Show("Неизвестный ISBN");
        return;
    }
    XmlNodeList gernes =
book.SelectSingleNode("Gernes").ChildNodes;
    foreach (XmlNode gerne in gernes)
    {
        string titleGerne = gerne.SelectSingleNode("Title").InnerText;
        string information = null;
        if (gerne.SelectSingleNode("Information") != null)
            information = gerne.SelectSingleNode("Information").InnerText;
        var list = (from x in ses.CreateCriteria<Gerne>().List<Gerne>()
            where x.Title == titleGerne select x).ToArray();
        Gerne g;
        if (list.Length > 0)
        {
            g = list[0];
            g.Information = information;
        }
        else
        {
            g = new Gerne();
            g.Title = titleGerne;
            g.Information = information;
        }
        ses.SaveOrUpdate(g);
        b.Gernes.Add(g);
    }
    ses.SaveOrUpdate(b);
    XmlNodeList authors = book.SelectSingleNode("BookAuthors").
        ChildNodes;
    foreach (XmlNode author in authors)
    {

```

```

        Guid guidAuthor = Guid.Parse(author.InnerText);
        if (dictAuthor.ContainsKey(guidAuthor))
            dictAuthor[guidAuthor].Add(b);
        else
        {
            List<Book> list = new List<Book>();
            list.Add(b);
            dictAuthor.Add(guidAuthor, list);
        }
    }
}
XmlNodeList AuthorsBook =
    doc.DocumentElement.SelectSingleNode("Authors").ChildNodes;
foreach (XmlNode authorBook in AuthorsBook)
{
    string surname =
authorBook.SelectSingleNode("Surname").InnerText;
    string name = authorBook.SelectSingleNode("Name").InnerText;
    string midName = null;
    if (authorBook.SelectSingleNode("MidName") != null)
        midName =
authorBook.SelectSingleNode("MidName").InnerText;
    int? year = null;
    if (authorBook.SelectSingleNode("Year") != null)
        year = Convert.ToInt32(authorBook.
            SelectSingleNode("Year").InnerText);
    Guid guid =

        Guid.Parse(authorBook.SelectSingleNode("Id").InnerText);
    var list = (from x in
                ses.CreateCriteria<Author>().List<Author>()
                where x.Surname == surname && x.Name == name &&
                    x.MidName == midName && x.Year == year
                select x).ToArray();
    Author author;
    if (list.Length>0)
    {
        author = list[0];
    }
    else
    {
        author = new Author();
    }
}

```



```

        author.AppendChild(midName);
    }
    if (a.Year.HasValue)
    {
        XmlNode year = doc.CreateElement("Year");
        year.InnerText = a.Year.Value.ToString();
        author.AppendChild(year);
    }
    Guid guid = Guid.NewGuid();
    dict.Add(a.Id, guid);
    XmlNode id = doc.CreateElement("Id");
    id.InnerText = guid.ToString();
    author.AppendChild(id);
    authors.AppendChild(author);
}
foreach (Book b in ses.CreateCriteria<Book>().List<Book>())
{
    XmlNode book = doc.CreateElement("Book");
    XmlNode title = doc.CreateElement("Title");
    title.InnerText = b.Title;
    book.AppendChild(title);
    if (b.Year.HasValue)
    {
        XmlNode year = doc.CreateElement("Year");
        year.InnerText = b.Year.Value.ToString();
        book.AppendChild(year);
    }
    if (b.PageCount.HasValue)
    {
        XmlNode pageCount = doc.CreateElement("PageCount");
        pageCount.InnerText = b.PageCount.Value.ToString();
        book.AppendChild(pageCount);
    }
    XmlNode ISBN = doc.CreateElement("ISBNBook");
    ISBN.InnerText = b.PublishingBook.ISBN.ToString();
    book.AppendChild(ISBN);
    XmlNode gernes = doc.CreateElement("Gernes");
    foreach (Gerne g in b.Gernes)
    {
        XmlNode gerne = doc.CreateElement("Gerne");
        XmlNode titleGerne = doc.CreateElement("Title");
        titleGerne.InnerText = g.Title;
        gerne.AppendChild(titleGerne);
        if (g.Information != null)

```

```

    {
        XmlNode information = doc.CreateElement("Information");
        information.InnerText = g.Information;
        gerne.AppendChild(information);
    }
    gernes.AppendChild(gerne);
}
book.AppendChild(gernes);
XmlNode bookAuthors = doc.CreateElement("BookAuthors");
foreach (Author a in b.Authors)
{
    XmlNode bookAuthor = doc.CreateElement("BookAuthor");
    bookAuthor.InnerText = dict[a.Id].ToString();
    bookAuthors.AppendChild(bookAuthor);
}
book.AppendChild(bookAuthors);
books.AppendChild(book);
}
}
XmlNode library = doc.CreateElement("Library");
library.AppendChild(books);
library.AppendChild(authors);
doc.AppendChild(library);
doc.Save(saveFileDialog.FileName);
}
}
}
}
}

```