

Дипломная работа

**На тему: Проектирование
информационной подсистемы
библиотеки**

Содержание:

Введение.....	4
Техническое задание.....	7
Назначение разработки.....	7
Требования к программе.....	8
Требования к программной документации.....	11
1. Анализ исходных данных.....	12
1.1 Описание предметной области.....	12
2. Специальная часть. Выбор средств/методологии проектирования.....	16
2.1 Сравнительные характеристики СУБД.....	16
2.2 Выбор СУБД.....	22
2.3 Сравнительные характеристики языков программирования.....	24
2.4 Выбор среды и языка программирования для реализации системы.....	29
2.5 Построение инфологической (концептуальной) модели предметной области.....	32
2.6 Проектирование логической структуры базы данных.....	38
2.7 Проектирование физической структуры базы данных.....	39
2.8 Создание датологической модели базы данных.....	46
2.9 Алгоритмизация приложения.....	48
2.10 Программирование, создание приложения.....	51
3. Экспериментальная часть.....	59
3.1 Объект испытаний.....	59
3.2 Цель испытаний.....	59
3.3 Перечень документов, предъявляемых на испытания.....	59
3.4 Объем испытаний.....	60
3.5 Методика проведения проверки комплектности программной документации.....	64
3.6 Методика проверки выполнения основных функций системы.....	67
4. Технологическая часть.....	79

4.1. Назначение программы	79
4.2. Условия выполнения программы	81
4.3. Требования к персоналу (пользователю).....	82
4.4. Выполнение программы	83
5. Экономическая часть	104
5.1 Расчет экономической эффективности программного обеспечения... ..	104
5.2. Выводы по экономической эффективности создания.....	114
программного обеспечения	114
6. Охрана труда и окружающей среды.....	115
6.1 Обеспечение условий труда на рабочем месте библиотекаря.....	115
6.2. Анализ состояния охраны труда в помещении	117
6.3. Анализ вентиляции, искусственного и природного освещения.....	118
6.4. Мероприятия по обеспечению благоприятных условий труда.....	127
6.5. Микроклимат рабочей зоны.....	128
6.6. Выводы.....	129
Выводы.....	130
Список использованной литературы.....	132

Введение

Не вызывает сомнений, что в последние годы потребность в базах данных различной архитектуры в нашей стране значительно возросла. База данных (БД) представляет собой организованную структуру, используемую для хранения данных, т.е. любых сведений о явлениях, процессах, действиях и т.д.

Если еще 20 лет назад в большинстве организаций использовались картотеки, то сегодня «шкафы с ящичками» можно встретить все реже и реже – компьютерные базы данных, значительно более компактные (раньше в научных библиотеках каталоги занимали не одну комнату, теперь же помещаются на крошечном компакт-диске), удобные в работе и позволяющие достичь высочайших скоростей обработки информации, стремительно завоевывают популярность.

В зависимости от рода деятельности организации формируются ее требования к используемой базе данных. Так, операторам сотовой связи нужны сложные биллинговые системы – миллионы абонентов, десятки тарифных планов, миллиарды звонков, каждый из которых надо правильно протарифицировать... Разумеется, они не жалеют денег на оплату труда программиста, который способен написать такую систему. В большинстве же случаев записей в базе данных сравнительно немного, алгоритмы обработки тоже стандартные (поиск и выборка). Примером может служить разрабатываемая в данном задании база данных, посвященная учету расчетов с клиентами в банке.

Представьте работу библиотеки: пришедший читатель просит дать ему литература, но объяснить, чего он хочет, в форме «автор – тип литературы – название издания – год выпуска» он не может. Обычно в памяти откладываются лишь обрывочные сведения. Сотрудники библиотеки тоже не в состоянии запомнить размещение тысяч изданий – а найти надо быстро,

нельзя же сказать читателю: «Подождите часика три, мы пока поищем». Здесь и приходит на помощь разрабатываемая база данных, позволяющая найти нужную литературу по жанру (в соответствии с пожеланием «Хочу какую-нибудь художественную и фантастическую книгу»), по языку («Только чтоб по-русски!»), по автору («Мне очень нравится, как пишет Василий Головачев. Найдите какую-нибудь его книжку») и т.п.

В связи с вышеупомянутым **цель** данной работы:

1. Проанализировать деятельность работы библиотекарей и читателей.
2. Спроектировать автоматизированную систему библиотекаря, максимально учитывая все предъявляемые требования к системе.

Объект исследования - автоматизация процесса поиска и выбора нужной литературы читателем, а также быстрой выдачи заказанной литературы библиотекарями.

Предмет исследования - разработка базы данных, хранящей все каталоги и справочники библиотекаря, а также программы, являющейся системой управления базой данных (СУБД), реализующей действия поиска литературы, ведения всех каталогов и справочников и учитывающей все требования читателей и библиотекарей.

Цель, объект и предмет данной работы обусловили постановление и решение следующей **задачи** - разработать **Автоматизированную Систему Библиотека (АС Библиотека)**, представляющую собой программу и базу данных, позволяющие автоматизировать взаимодействие библиотекаря и читателя.

Решение поставленной задачи требовало адекватного методологического обеспечения. Был использован широкий спектр технической, математической, научно-методической литературы, а также использованы санитарные нормы и правила.

Структура и объем работы: выпускная работа состоит из вступления, основных разделов, заключительной части, приложений и списка использованных источников.

Научная новизна данной работы заключается в том, что был разработан комплексный метод автоматизации взаимодействия библиотекаря и читателя, позволяющий наиболее полно учитывать все требования, предъявляемые к поиску литературы и ведению основных каталогов и вспомогательных справочников, а так же спроектирована система, обеспечивающая ввод исходных данных, хранения, изменения, структурирования, удаления информации, выполнение автоматизированной обработки новых поступлений в библиотеку, упрощает поиск и анализ библиотечной информации.

Практическое значение данной работы состоит в том, что его результаты позволяют:

- Выявить проблемы взаимодействия библиотекарей и читателей.
- Спроектировать автоматизированную систему Библиотека.
- Реализовать данную автоматизированную систему Библиотека в высокоэффективной и простой в применении программе (АС Библиотека), готовой для практического использования в библиотеках.
- Разработанная программа (АС Библиотека) может получить широкое применение во всех библиотеках, различного масштаба.

Техническое задание

Любой организация, в том числе библиотекам, необходимо хранить и иметь быстрый доступ к информации связанной с ее деятельностью. Современным средством реализации данных функций являются базы данных. Они способны обеспечить надежное хранение информации, ее структурирование, быстрый и простой доступ к ней.

Поэтому любой современной организация необходима автоматизированная система, состоящей из базы данных и программы управляющей нею, выполняющей функции хранения, администрирования, управления данными.

Данная автоматизированная система должна быть построена в соответствии с современными принципами и технологиями. Она должна хранить и предоставлять по запросам различную библиотечную информацию.

Помимо этого в программу автоматизированной системы должна быть встроена подпрограмма, позволяющая читателю быстро найти нужную ему литературу.

Программа, управляющая базой данных, должна быть с понятным простому пользователю интерфейсом. Работать в диалоговом режиме, используя различные формы, поля и списки.

Назначение разработки

Функциональное назначение:

Автоматизированная система предназначена для ввода, хранения, изменения, структурирования, удаления информации, что означает

выполнение автоматизированной обработки новых поступлений в библиотеку, упрощает поиск и анализ библиотечной информации.

Эксплуатационное назначение:

Автоматизированная система должна использоваться в библиотеках для освобождения сотрудников от ряда рутинных работ по подготовке карточек, изданий, списков, а так же для получения читателем рекомендаций по выбору книг.

Требования к программе

Требования к функциональным характеристикам

Состав выполняемых функций

Программа должна выполнять следующие функции:

- Ввод данных;
- Структурирование;
- Хранение данных;
- Редактирование данных;
- Поиск
- Удаление данных;
- Тестирование и выдача рекомендаций.

Системные требования (входные данные для работы программы).

Системные требования состоят в обеспечении работы программы с входными данными, которыми служат информация о книгах (название, автор, издательство), которые вводятся в программу вручную, набирая с клавиатуры.

Выходные данные

В данном случае выходными данными является литература. В окне отображения информации должна быть представлена информация о названии, авторе, издательстве, дате, типе, издания, наличии, местонахождении, а также сообщения, выводимые на экран во время работы с программой (уведомления об ошибках ввода, результатов поиска и т.д.).

В случае поиска литературы – рекомендации.

Требования к надежности программы

Для обеспечения надёжного функционирования необходимо провести проверку работы программы. Для чего базу данных необходимо заполнить. Провести проверку функции структурирования, поиска данных, а так же их редактирование и удаление. Помимо этого необходима проверка работы поиска литературы.

Условия эксплуатации

Для разрабатываемой программы необходимо определить группу пользователей, которые работают с данной задачей. Далее поместить на соответствующий этой группе рабочий стол ссылку для работы с программой.

Для эксплуатации программы требуется библиотечный сотрудник знающий предметную область.

Требования к средствам программирования

Операционная система: Windows 2000 Professional / Windows XP Professional.

Язык программирования и формат базы данных - выбрать самостоятельно (ориентируясь при выборе на язык Delphi и базы данных – MS Access).

Дополнительные программные средства не требуются.

Требования к составу и параметрам технических средств

Для функционирования программы необходим компьютер типа IBM PC со следующей минимальной конфигурацией:

- Процессор Intel 1ГГц;
- Память: 512 Mb;
- Жесткий диск емкостью не менее 20 Гб;
- Дисплей SVGA с соответствующим видеоадаптером;
- Минимальное разрешение экрана 800x600 (рекомендуется 1024x768)
- Клавиатура, указательное устройство “мышь”.

Требования к информационной и программной совместимости

Для работы exe-модуля программы использовать следующие операционные системы: Windows 2000 Professional / Windows XP Professional.

Требования к программной документации

Состав программной документации

В ходе разработки программы должна быть подготовлена следующая документация:

- Текст программы.
- Программа и методика испытаний.
- Руководство пользователя.
- Пояснительная записка.

1. Анализ исходных данных

1.1 Описание предметной области

В данной работе затрагивается лишь та деятельность библиотеки, которая касается:

- книжного фонда, состоящего из учебной, научно-технической, художественной и прочей литературы;
- пользователей библиотеки, которые выбирают по каталогам нужную им литературу, причем на руки выдается только один экземпляр книги, а в один заказ может входить несколько различных книг (код заказа, по которому можно найти студента, идентифицируется с номером читательского билета);
- библиотекарей, которые выдают заказанные книги (ища их в местах размещения и хранения книжного фонда).

Для ведения библиотечных каталогов, организации поиска требуемых изданий и библиотечной статистики библиотекари в каталогах хранят сведения, большая часть которых размещаются в аннотированных каталожных карточках (рис.1).

Д27 Дейт К. Руководство по реляционной СУБД DB2 / Пер. с англ. и предисл. М.Р.Когаловского. – М.: Финансы и статистика, 1988. – 320 с.: ил.
ISBN 5-279-00063-9
Книга американского специалиста в области реляционных баз данных К.Дейта, автора популярной в СССР монографии "Введение в системы баз данных" (М.: Наука, 1981), представляет собой руководство по перспективной СУБД фирмы ИБМ DB2, сочетающей возможности широко известной системы IMS/VS и реляционной СУБД.
Для специалистов по программному обеспечению информационных систем и студентов вузов.
ББК 32.973

Рисунок 1.1 – Пример каталожной карточки

Анализ запросов на литературу (как читателями, так и сотрудниками библиотек) показывает, что для поиска подходящих изданий (по тематике, автору, художнику, издательству и т.п.) и отбора нужного (например, по аннотации) следует выделить следующие атрибуты каталожной карточки:

1. Автор (фамилия и имена (инициалы) или псевдоним каждого автора издания).
2. Название (заглавие) издания.
3. Номер тома (части, книги, выпуска).
4. Вид издания (сборник, справочник, монография, ...).
5. Составитель (фамилия и имена (инициалы) каждого из составителей издания).
6. Язык, с которого выполнен перевод издания.
7. Переводчик (фамилия и инициалы каждого переводчика).
8. Под чей редакцией (фамилия и имена (инициалы) каждого из титульных редакторов).
9. Художник (фамилия и имена (инициалы) каждого художника-иллюстратора) - для художественных изданий, иллюстрируемых оригинальными рисунками.
10. Повторность издания (второе, одиннадцатое и т.п.).
11. Характер переиздания (исправленное, дополненное, переработанное, стереотипное и т.п.).
12. Место издания (город).
13. Издательство (название издательства).
14. Год выпуска издания.
15. Издательская аннотация или реферат.
16. Библиотечный шифр (например, ББК 32.973).
17. Авторский знак (например, Д27).

Библиотечный шифр и авторский знак используются при составлении каталогов и организации расстановки изданий на полках: по содержанию (в

соответствии с библиотечным шифром) и алфавиту (в соответствии с авторским знаком).

Библиотечно-библиографическая классификация (ББК) распределяет издания по отраслям знания в соответствии с их содержанием. В ней используется цифро-буквенные индексы ступенчатой структуры.

Каждый из девяти классов (1. Марксизм-ленинизм; 2. Естественные науки; 3. Техника. Технические науки; 4. Сельское и лесное хозяйство; 5. Здравоохранение; 6/8. Общественные и гуманитарные науки; 9. Библиографические пособия. Справочные издания. Журналы.) делится на подклассы и следующие ступени деления:

3. Техника. Технические науки.

32 Радиоэлектроника.

32.97 Вычислительная техника.

32.973 Электронные вычислительные машины и устройства.

32.973.2 Электронно вычислительные машины и устройства дискретного действия.

Шифр ББК используется при выделении хранимым изданиям определенных комнат, стеллажей и полок, а также для составления каталогов и статистических отчетов.

Авторский знак, состоящий из первой буквы фамилии (псевдонима) автора или названия издания (для изданий без автора) и числа, соответствующего слогу, наиболее приближающегося по написанию к первым буквам фамилии (названия), упрощает расстановку книг на полках в алфавитном порядке.

К объектам и атрибутам, позволяющим охарактеризовать отдельные экземпляры изданий (переплеты), места их хранения и читателей, можно отнести:

18. Номер комнаты (помещения для хранения переплетов).

19. Номер стеллажа в комнате.

20. Номер полки на стеллаже.

21. Номер (инвентарный номер) переплета.
22. Дата приобретения конкретного переплета.
23. Цена конкретного переплета.
24. Дата размещения конкретного переплета на конкретном месте.
25. Дата изъятия переплета с установленного места.
26. Номер читательского билета (формуляра).
27. Фамилия читателя.
28. Имя читателя.
29. Отчество читателя.
30. Адрес читателя.
31. Телефон читателя.
32. Дата выдачи читателю конкретного переплета.
33. Срок, на который конкретный переплет выдан читателю.
34. Дата возврата переплета.

В настоящее время библиотека работает по следующей структурной схеме:

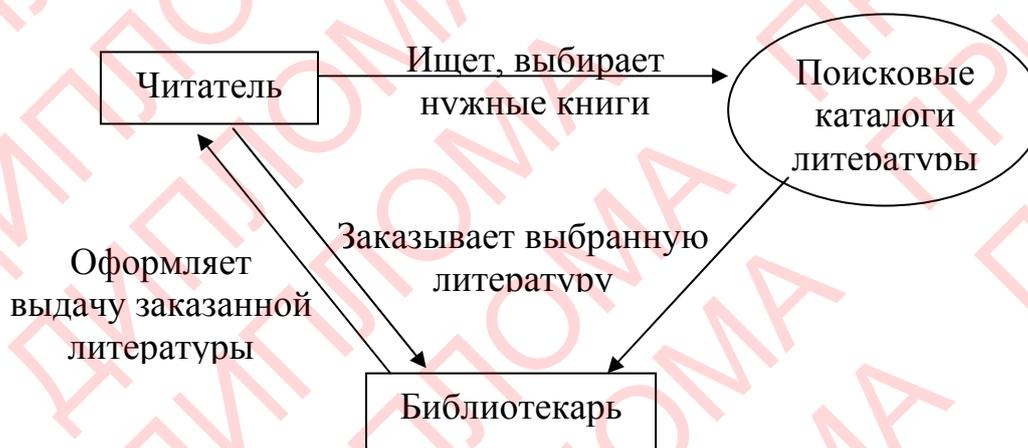


Рисунок 1.2 - Структурная схема работы библиотеки.

2. Специальная часть. Выбор средств/методологии проектирования

2.1 Сравнительные характеристики СУБД

В настоящее время для построения информационных систем применяются различные системы управления базами данных (СУБД), различающиеся как своими возможностями, так и требованиями к вычислительным ресурсам.

Все многообразие применяемых СУБД, однако, можно свести к двум основным их классам:

1. Персональные (настольные);
2. Многопользовательские (профессиональные, или промышленные).

К первому классу относятся СУБД, ориентированные для работы на персональном компьютере (dBASE, FoxPro, MS Access и т.п.). Изначально они поддерживали работу с данными только одного пользователя. Вся СУБД такого класса выполняется как единая программа, таблицы базы данных представляются отдельными файлами на диске того же персонального компьютера.

С развитием локальных сетей разработчики СУБД этого класса стали приспособлять их к работе в сетевой среде, в которой потенциально стало возможным организовать доступ к данным с нескольких персональных компьютеров, включенных в локальную сеть. Файлы базы данных при этом размещаются на файловом сервере. На каждом же рабочем месте работает собственная копия программы-СУБД и прикладная программа, и на их выполнение могут оказывать существенное влияние характеристики компьютера этого рабочего места. Таким образом, при наличии в сети N рабочих мест с одними и теми же данными работают N копий программы-

СУБД, одними и теми же данными управляют сразу N копий СУБД. Ошибка в выполнении одной из копий не будет замечена другими копиями. При выполнении запросов к базе данных копия СУБД может либо производить поиск данных в удаленных файлах на файловом сервере, либо копировать все файлы, в которых ведется поиск в свою локальную файловую систему. В первом случае возникают проблемы одновременного доступа к данным при их изменении. Данные, над которыми производится изменения, должны быть заблокированы. Средства файлового сервера позволяют выполнять блокировку на уровне файлов, но не на уровне записей, что существенно снижает эффективность параллельной работы с базой данных многих пользователей. Во втором же случае, во-первых, требуется передача по сети больших объемов информации, а во-вторых, получается, что разные рабочие места работают с разными копиями данных и эти копии могут стать неидентичными.

Персональные системы управления данными - это программное обеспечение, ориентированное на решение задач локального пользователя или компактной группы пользователей и предназначенное для использования на микроЭВМ (персональном компьютере). Это объясняет и их второе название - настольные. Определяющими характеристиками настольных систем являются:

- относительная простота эксплуатации, позволяющая создавать на их основе работоспособные приложения как "продвинутым" пользователям, так и тем, чья квалификация невысока;
- относительно ограниченные требования к аппаратным ресурсам.

Исторически первой среди персональных СУБД, получивших массовое распространение, стала Dbase фирмы Ashton-Tate (впоследствии права на нее перешли к фирме Borland, а с 1999 г. данная программа поддерживается фирмой dBASE Inc.). В дальнейшем серия реляционных персональных СУБД пополнилась такими продуктами, как FoxBase/FoxPRO (Fox Software, в дальнейшем - Microsoft), Clipper (Nantucket, затем - Computer Associates),

R:base (Microrim), Paradox (Borland, на настоящий момент правами владеет фирма Corel), Access (Microsoft), Approach (Lotus).

Завоевавшие широкую популярность в России системы Dbase, FoxPRO и Clipper работали с таблицами данных, размещавшихся в файлах, имевших расширение *.dbf (термин dbf-формат стал общепринятым). Впоследствии семейство этих баз данных получило интегрированное наименование Xbase.

Несмотря на неизбежные различия, обуславливавшиеся замыслами разработчиков, все перечисленные системы в ходе своей эволюции приобрели ряд общих конструктивных черт, среди которых, прежде всего, могут быть названы:

- наличие визуального интерфейса, автоматизирующего процесс создания средств манипуляции данными, - экранных форм, шаблонов отчетов, запросов и т. п.;
- наличие инструментов создания объектов базы данных в режиме диалога: Experts в Paradox, Wizards в Access, Assistants в Approach;
- наличие развитого инструментария создания программных расширений в рамках единой среды СУБД: язык разработки приложений PAL в Paradox, VBA (Visual Basic for Applications) в Access, Lotus Script в Approach;
- встроенная поддержка универсальных языков управления данными, например SQL или QBE (Query By Example).

Среди СУБД, которые, условно говоря, занимают промежуточное положение между настольными и промышленными системами, могут быть названы SQLWindows/ SQLBase фирмы Centura (до 1996 г. Gupta), InterBase (Borland), наконец, Microsoft SQL Server.

СУБД второго класса изначально создавались для выполнения на больших компьютерах и обеспечения параллельной работы многих пользователей.

Профессиональные (промышленные) СУБД представляют собой программную основу для разработки автоматизированных систем

управления крупными экономическими объектами. На их базе создаются комплексы управления и обработки информации крупных предприятий, банков или даже целых отраслей. Первостепенными условиями, которым должны удовлетворять профессиональные СУБД, являются:

- возможность организации совместной параллельной работы большого количества пользователей;
- масштабируемость, то есть возможность роста системы пропорционально расширению управляемого объекта;
- переносимость на различные аппаратные и программные платформы;
- устойчивость по отношению к сбоям различного рода, в том числе наличие многоуровневой системы резервирования хранимой информации;
- обеспечение безопасности хранимых данных и развитой структурированной системы доступа к ним.

Такие СУБД, как правило, состоят из ядра, постоянно присутствующего в памяти, (сервера) и большого количества программ-агентов, обслуживающих запросы конечных пользователей и прикладных программ (клиентов). В этом случае и ядро СУБД, и данные находятся на одном и том же компьютере. Одна копия СУБД управляет одной копией данных. Единая управляющая система позволяет эффективно организовать одновременный доступ к данным многих агентов, предотвращая конфликты между ними. Ошибка в работе СУБД локализована и может быть эффективно исправлена самой же СУБД. При работе в условиях сети ядро СУБД выполняет запросы агентов на выборку данных и передает по сети только результаты выборки. Поскольку быстродействие современных дисковых систем обычно выше, чем скорость передачи данных по сети, уменьшение объема передаваемых данных существенно увеличивает общую эффективность работы системы. При этом не накладывается никаких ограничений на масштаб сети, агенты могут быть связаны с ядром СУБД через любую сеть и любые протоколы передачи данных.

Многопользовательские СУБД обладают также неоспоримыми преимуществами в таких аспектах, как надежность, безопасность, доступность. Многопользовательские СУБД с самого начала своей истории использовали в качестве интерфейса запросов язык SQL, отсюда произошло одно из их альтернативных названий - SQL-серверы. Хотя в последнее время подмножества SQL становятся доступными и в персональных СУБД, но в эти подмножества не включаются средства обеспечения безопасности и параллельного доступа к данным - те средства, которые персональные СУБД обеспечить просто не могут.

Промышленные СУБД к настоящему моменту имеют уже достаточно богатую историю развития. В частности, можно отметить, что в конце 70-х - начале 80-х годов в автоматизированных системах, построенных на базе больших вычислительных машин, активно использовалась СУБД Adabas. В настоящее время характерными представителями профессиональных СУБД являются такие программные продукты, как Oracle, DB2, Sybase, Informix, Ingres, Progress.

Основоположниками СУБД Oracle стала группа американских разработчиков (Ларри Эллисон, Роберт Майнер и Эдвард Оутс), которые более двадцати лет тому назад создали фирму Relational Software Inc. и поставили перед собой задачу создать систему, на практике реализующую идеи, изложенные в работах Э. Ф. Кодда и К. Дж. Дейта. Результатом их деятельности стала реализация переносимой реляционной системы управления базами данных с базовым языком обработки SQL. В 1979 г. заказчикам была представлена версия Oracle для мини-компьютеров PDP-11 фирмы Digital Equipment Corporation сразу для нескольких операционных систем: RSX-11, IAS, RSTS и UNIX. Чуть позже Oracle был перенесен на компьютеры VAX под управлением VAX VMS. Значительная часть кода была написана на ассемблере, и поэтому процесс переноса системы на новую платформу требовал значительных усилий. Основным отличием Oracle очередной, третьей версии было то, что она была полностью написана на

языке C. Такое решение обеспечивало переносимость системы на многие новые платформы, в частности, на различные клоны UNIX. Второй важной особенностью новой (1983 г.) версии была поддержка концепции транзакции. Примерно в это же время фирма получила новое имя - Oracle Corporation - и заняла лидирующее место на рынке производителей СУБД. Четвертая версия Oracle характеризовалась расширением перечня поддерживаемых платформ и операционных систем. Oracle был перенесен как на большие ЭВМ фирмы IBM (мэйнфреймы), так и на персональные компьютеры, работающие под управлением MS DOS. Именно в четвертой версии был сделан важный шаг в развитии технологии поддержки целостности баз данных. Для многопользовательских систем было предложено оригинальное решение Oracle поддержки "непротиворечивости чтения". В пятой версии была впервые реализована СУБД с архитектурой "клиент- сервер". Последующие версии СУБД Oracle были ориентированы на построение крупномасштабных систем обработки транзакций, изменение методов реализации систем ввода/вывода, буферизации, подсистем управления параллельным доступом, резервирования и восстановления. Также была реализована поддержка симметричных мультипроцессорных архитектур.

Проект и экспериментальный вариант СУБД Ingres были разработаны в университете Беркли под руководством одного из наиболее известных в мире ученых и специалистов в области баз данных Майкла Стоунбрейкера. С самого начала СУБД Ingres разрабатывалась как мобильная система, функционирующая в среде ОС UNIX. Первая версия Ingres была рассчитана на 16-разрядные компьютеры и работала главным образом на машинах серии PDP. Это была первая СУБД, распространяемая бесплатно для использования в университетах. Впоследствии группа Стоунбрейкера перенесла Ingres в среду ОС UNIX BSD, которая также была разработана в университете Беркли. Семейство СУБД Ingres из университета Беркли принято называть университетской Ingres. В начале 80-х была образована компания RTI (Relational Technology Inc.), которая разработала и стала продвигать

коммерческую версию СУБД Ingres. В настоящее время коммерческая Ingres поддерживается, развивается и продается компанией Computer Associates. Сейчас это одна из наиболее развитых коммерческих реляционных СУБД. В то же время, по поводу университетской Ingres имеется много высококачественных публикаций. Более того, университетскую Ingres можно опробовать на практике и даже посмотреть ее исходные тексты.

Перечисленные выше (для СУБД Oracle) тенденции носят универсальный характер и определяют пути развития других программных продуктов, что вполне объясняется жесткой конкурентной ситуацией, сложившейся на данном рынке.

В последние годы наметилась устойчивая тенденция к стиранию четких граней между настольными и профессиональными системами. Последнее, в первую очередь, объясняется тем, что разработчики в стремлении максимально расширить потенциальный рынок для своих продуктов постоянно расширяют набор их функциональных характеристик.

2.2 Выбор СУБД

На сегодняшний день существует много разнообразных систем управления базами данных. Это такие СУБД как Paradox, FoxPro, Clipper, Access и др. Для работы с большинством из них требуются достаточно глубокие знания данной СУБД и опыт программирования.

Microsoft Access в настоящее время является одной из самых популярных среди настольных (персональных) программных систем управления базами данных. Среди причин такой популярности следует отметить:

- высокую степень универсальности и продуманности интерфейса, который рассчитан на работу с пользователями самой различной квалификации. В частности, реализована система управления объектами базы

данных, позволяющая гибко и оперативно переходить из режима конструирования в режим их непосредственной эксплуатации;

- глубоко развитые возможности интеграции с другими программными продуктами, входящими в состав Microsoft Office, а также с любыми программными продуктами, поддерживающими технологию OLE;

- богатый набор визуальных средств разработки.

Успех Microsoft Access заключается в прекрасной реализации продукта, рассчитанного как на начинающего, так и квалифицированного пользователя. Microsoft Access – это самая популярная сегодня настольная система управления базами данных.

В Microsoft Access присутствует язык программирования Visual Basic, который позволяет создавать массивы, свои типы данных, контролировать работу приложений. MS Access имеет один из самых лучших наборов визуальных средств разработки и представления информации среди аналогичных программных продуктов.

Одно из основных преимуществ MS Access – интеграции с популярным офисным пакетом Microsoft Office.

Вся работа с базой данных осуществляется через окно контейнера базы данных. Отсюда осуществляется доступ ко всем объектам: таблицам, запросам, формам, отчетам, макросам, модулям.

Встроенный язык запросов SQL позволяет максимально гибко работать с данными и значительно ускоряет доступ к внешним данным.

Access воспринимает большое количество форматов данных, включая файловые структуры других СУБД. Поэтому приложение в Access может импортировать из текстовых файлов или электронных таблиц и экспорт в них: предоставлять прямой доступ и обновлять файлы Paradox, FoxPro и других БД. Можно также импортировать данные из этих файлов в таблицы Access.

Преимуществом Access является наличие средств проектирования приложения БД без знания языка программирования. Работа в Access

начинается с определения реляционных таблиц и полей, предназначенных для хранения данных. Сразу после этого с помощью форм, отчетов, макросов и VBA можно определять действия над этими данными. Формы и отчеты используются для вывода на экран и дополнительных вычислений при работе с таблицами. В случае разработки более сложного приложения можно использовать язык Visual Basic.

Архитектура Access называет объектами все, что может иметь имя. В БД Access основными объектами являются таблицы, запросы, формы, отчеты, макросы и модули. Термин БД обычно относится только к файлам, в которых хранятся данные. В Access БД включает все объекты, связанные с хранимыми данными, в том числе и те, которые определяются для автоматизации работы.

В таблицах хранятся данные. Используя формы, можно выводить данные на экран или изменять их. Формы и отчеты получают данные как непосредственно из таблиц, так и через запросы. Для выполнения вычислений запросы могут использовать встроенные функции или функции, созданные с помощью Visual Basic для приложений.

2.3 Сравнительные характеристики языков программирования

Языки программирования предназначены для кодирования задач в более понятной форме и автоматическому переводу (трансляции) из этой формы на язык машинных кодов.

Классификация языков программирования производится по разным признакам:

1. По принципу трансляции - интерпретаторы (построчная трансляция) и компиляторы (трансляция всей программы). Например, FoxPro, Basic являются интерпретаторами, C++, Delphi - компиляторами. Что из этого

следует? При помощи интерпретаторов можно быстро собрать работающее приложение и изменять его по ходу работы программы, вплоть до того, что создавать программы “на лету”. Но за это приходится расплачиваться скоростью работы конечной программы – откомпилированная программа всегда работает быстрее. Правда, при таком развитии компьютерной техники и на тех задачах, которые решаются при выполнении инженерных расчетов – это различие в скорости становится незаметным.

2. По способу программирования – процедурные и не процедурные (объектно-ориентированные). Процедурные языки были исторически более ранними и основаны на программировании ДЕЙСТВИЙ над переменными (фортран, алгол, паскаль, бэйсик). Современные языки программирования предназначены для ОП-программирования, т.е. описываются объекты, свойства и методы, а программа пишется как обработчик событий, происходящих с объектами. Во многие старые процедурные языки добавлены средства для ОП-программирования: C → C++ → C#, Basic → Visual Basic → VBA → VBScript, Pascal → Object Pascal → Delphi. Современные ЯП интерпретирующего типа называют скриптами (JavaScript, Python, PHP, Perl). Они получили особое распространение при создании Интернет-приложений.

3. По уровню абстракции конструкций – языки программирования высокого уровня (полностью независимы от системы команд процессора, на котором будут выполняться) и низкого уровня (зависят от системы команд конкретного процессора). Среди языков программирования высокого уровня выделяют группу языков сверхвысокого уровня абстракции (например, лисп, пролог), которые предназначены для решения сложнейших задач, называемых задачами искусственного интеллекта. ЯП низкого уровня называют ассемблерами, добавляя в конце марку процессора, систему команд которого они поддерживают.

4. По принципу выполнения программы - однопоточные и многопоточные (с возможностью параллельного выполнения программ - Java).

Основные критерии для выбора языков программирования:

1. простота написания программ и понятность языка программирования для широкого круга пользователей;
2. возможности предоставляемые языком для решения разнообразных задач;
3. простота компиляции программ и установки их на различные компьютеры пользователей;
4. скорость работы программ;

Ассемблер

Этот язык наиболее близок к аппаратуре и поэтому является машинно-зависимым и не является мобильным. Необходимость программирования на ассемблере возникает, как правило, при невозможности воспользоваться языками высокого уровня.

Фортран

Несмотря на то, что Фортран является старейшим из языков высокого уровня, он и сейчас успешно применяется для математических расчетов. Так, на Фортране удобно программировать задачи статистического анализа и другие научные приложения. Основной целью при проектировании языка было достижение высокой эффективности при выполнении программ. Цель была достигнута, но для этого пришлось в некоторой степени пожертвовать гибкостью языка. Например, имеется только один вид оператора цикла. Кроме того, тексты программ должны иметь довольно жесткий формат. Отметим, правда, что использование препроцессоров несколько сглаживает указанные недостатки.

Язык С

Язык С тесно связан с ОС UNIX, так как первоначально был разработан именно для реализации ядра операционной системы. Поэтому, в первую очередь, он очень удобен для программирования задач, использующих системные вызовы операционной системы, например, для организации низкоуровневого ввода/вывода, управления памятью или физическими устройствами, организации связи между процессами и т.д. Кроме того, язык С может успешно применяться и для реализации программ, не требующих такого непосредственного взаимодействия с операционной системой.

При выборе языка программирования следует иметь в виду следующие характеристики языка С:

1. Набор типов данных: символьный, целый, длинный целый, вещественный, и вещественный двойной точности.
2. Наличие низкоуровневых возможностей (напомним, что большая часть ядра ОС UNIX написана на С).
3. Возможность определения производных типов данных, таких как массивы, функции, указатели, структуры и объединения.
4. Наличие многомерных массивов.
5. Возможность определения указателей на данные конкретного типа и выполнения арифметических действий над указателями с учетом типа адресуемых ими данных.
6. Побитные операции.
7. Множество управляющих конструкций: if, if-else, switch, while, do-while и for.
8. Высокая степень мобильности программ.

Язык С естественным образом ориентирован на структурное программирование. Большие программы подразделяются на функции, которые можно считать отдельно компилируемыми единицами. Кроме облегчения внесения изменений в программы, при таком подходе в

наибольшей степени реализуется идеология программирования в ОС UNIX: стараться в максимальной степени использовать уже имеющиеся программы.

Язык С довольно труден в изучении. Чтобы научиться программировать на С, используя все его возможности, необходимо несколько месяцев интенсивной практики. Поэтому, если Вы программируете лишь эпизодически, лучше выбрать какой-нибудь другой, более простой язык.

Паскаль

Поскольку Паскаль первоначально проектировался как язык для обучения программированию, он прост в обращении и получил широкое признание. Кроме простоты, к достоинствам языка можно отнести высокую степень структурированности программ и возможность обращения к системным вызовам (аналогично языку С). Удобнее всего использовать Паскаль для небольших программ, что естественно объясняется его первоначальным предназначением. Недостатками языка являются, например, отсутствие возможности инициализации переменных и недостаточные средства для работы с файлами.

2.4 Выбор среды и языка программирования для реализации системы

Вся мировая индустрия средств разработки приложений движется в направлении максимального упрощения процесса создания программ, переводя его на визуальный уровень, что позволяет программистам сосредоточиться только на логике решаемых задач. В таком контексте данная программная система оптимизации была реализована в близкой к идеальной визуальной среде проектирования Borland Delphi 2006, на объектно-ориентированном языке программирования Object Pascal.

Преимущества выбранной среды проектирования Borland Delphi 2006:

- 1) Повышенное удобство работы программиста — усовершенствованная интегрированная среда разработчика. Добавленные средства навигации и просмотра для исходных текстов, классов, объектов. Повсеместно реализованный режим перетаскивания. Улучшенная поддержка групп проектов и файлов ресурсов (файлов RC) в Менеджере проектов. Возможность полной индивидуальной настройки среды, в том числе создания собственных произвольных клавиатурных раскладок редактора. При создании модулей данных доступен специальный режим визуального проектирования взаимосвязей между таблицами базы данных.
- 2) Расширенная библиотека компонентов. Более 200 компонентов позволяют, не прибегая к программированию, вести разработку сложных приложений, работающих с базами данных, в локальных сетях и в Интернете, только на визуальном уровне.
- 3) Улучшенные средства отладки приложений. Точки прерываний можно объединять в группы и задавать сложные логические условия их срабатывания. Допускается отлаживать различные внешние процессы, в том числе и выполняющиеся на других компьютерах.

- 4) Поддерживаются язык расширенной разметки *XML* и динамическая версия языка разметки гипертекста *HTML*. Новые компоненты для работы с *XML* дают возможность разворачивать свои программы в Интернете, создавать эффективные Web-модули, переносить ранее созданные приложения в сети, работающие на основе протоколов *TCP/IP*. Поддержка языка *HTML 4* позволяет использовать браузеры последнего поколения как готовые клиентские программы для работы с серверами приложений и баз данных.
- 5) Ориентация на Интернет технологии. Новые компоненты технологии *MIDAS* позволяют разворачивать базы данных в Интернете и строить в глобальной сети многоуровневые защищенные программные комплексы со встроенными серверами приложений.
- 6) Расширенные возможности по работе с СУБД *InterBase*. Значительно повышенная функциональность компонентов, позволяющих организовать прямую связь с *InterBase*, не прибегая к механизму *BDE*.
- 7) Включена поддержка технологии *ADO*. Наиболее передовая технология доступа к данным *Microsoft ADO*, не требующая драйверов *ODBC* или *BDE*, реализованная в виде большого набора компонентов. С помощью *ADO* возможно организовывать доступ к данным в произвольном формате.
- 8) Реализована система групповой работы *TeamSource*. Успешная реализация больших современных проектов немислима без согласованной работы обширных групп программистов. Для этого в среду *Delphi 5* включена система *TeamSource*, позволяющая отслеживать и контролировать все изменения, происходящие в проекте.
- 9) Автоматизирован перенос приложений на национальные языковые платформы. В систему *Delphi 5* встроен набор служебных программ для поддержки работы программы на нескольких национальных языках. Процесс локализации реализуется на визуальном уровне.

И это список лишь наиболее важных особенностей системы Borland Delphi 2006.

2.5 Построение инфологической (концептуальной) модели предметной области

Концептуальное проектирование начинается с анализа предметной области, включает анализ концептуальных требований и информационных потребностей, выявление информационных объектов (ИО) и связей между ними, построение концептуальной модели (схемы) данных, названной ER-диаграмма.

Одна из особенностей ER-диаграммы состоит в том, что результаты анализа предметной области отражены в достаточно простом, наглядном, но в то же время формализованном и достаточно информативном виде. В ER-диаграмме сочетаются функциональный и информативный подходы, что позволяет представлять как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных. Графическая форма позволяет отобразить типологию и свойства сущностей и связей, а формализмы, положенные в основу диаграммы, позволяют использовать на следующем шаге проектирования логической БД строгий аппарат нормализации.

Основными конструктивными элементами ER-диаграмм являются сущности, связи между ними и их атрибуты (свойства).

Сущность - любой конкретный (реальный) или абстрактный объект в рассматриваемом ПО.

Связь - наблюдаемая взаимосвязь (ассоциация) между сущностями.

В ER-диаграммах сущности помечаются прямоугольниками, ассоциации (характеры объединения сущностей) - ромбами или шестиугольниками, а связи между атрибутами - рёбрами, над которыми проставляются типы связей.

Построение модели БД

Для проектирования структуры БД необходима исходная информация о предметной области желательно в формализованном виде. Предметной областью (ПО) называется часть реальной системы, представляющая интерес для данного исследования.

При проектировании автоматизированных информационных систем ПО представляется в виде моделей данных нескольких уровней. Число используемых уровней зависит от сложности системы, но в любом случае включает логический и физический уровни. ПО может относиться к любому типу организации (институт, завод, малое предприятие и т.п.).

Необходимо различать полную ПО предприятия в целом и организационную единицу этой ПО. Организационная единица, в свою очередь, может представлять свою ПО (отделы).

Информация, необходимая для описания ПО, зависит от реальной модели и может включать сведения о персонале, заработной плате, товарах, то есть сведения о людях, местах, предметах, событиях и понятиях.

Описание ПО, без ориентации на используемые программные и технические средства, называется инфологической моделью (ИЛМ).

Инфологическая модель базы данных

Компоненты инфологической модели (ИЛМ) включают ряд описаний объектов ПО и связи между ними. Описание ПО всегда представлено в какой-то знаковой системе. Поэтому кроме отношений, присущих ПО, возникают еще и отношения, обусловленные особенностями отображения ПО в языковой среде.

В настоящее время не существует единого стандарта или общепринятого способа построения ИЛМ. Для описания ИЛМ используются как языки аналитического (описательного) типа, так и графические средства.

В процессе анализа в ПО выделяют классы объектов. Классом объектов называют совокупность объектов, обладающих одинаковым набором свойств. Например, если в качестве ПО рассмотреть вуз, то в ней можно выделить следующие классы объектов: учащиеся, преподаватели, аудитории и т. д. Объекты могут быть реальными или абстрактными, как, например, предметы, которые изучают студенты.

При отражении в ИС каждый объект представляется своим идентификатором, который отличает один объект класса от другого, а каждый класс объектов представляется именем этого класса. Так, для объектов класса «Изучаемые предметы» идентификатором каждого объекта будет «Название предмета». Идентификатор должен быть уникальным. Каждый объект обладает определенным набором свойств. Для объектов одного класса набор этих свойств одинаков, а их значения могут различаться.

Каждому классу объектов в ИЛМ присваивается уникальное имя.

При описании ПО надо отразить связи между объектом и характеризующими его свойствами. Это изображается в виде линии, соединяющей обозначение объекта и его свойств.

Связь между объектом и его свойством может быть различной. Объект может обладать только одним значением какого-то свойства. Например, каждый человек может иметь только одну дату рождения. Назовем такие свойства *единичными*. Для других свойств возможно существование одновременно нескольких значений у одного объекта. Пусть, например, при описании «Сотрудника» фиксируется в качестве его свойства «Иностранный язык», которым он владеет. Так как сотрудник может знать несколько иностранных языков, то такое свойство будем называть *множественным*. При изображении связи между объектом и его свойствами для единичных свойств будем использовать одинарную стрелку, а для множественных

свойств – двойную. Кроме того, некоторые свойства являются постоянными, их значение не может измениться с течением времени. Назовем такие свойства *статическими*, а те свойства, значение которых может изменяться со временем, будем называть *динамическими*.

Кроме связи между объектом и его свойствами, в ИЛМ фиксируются связи между объектами разных классов. Различают связи типа «один к одному» (1:1), «один ко многим» (1:M), «многие к одному» (M:1) и «многие ко многим» (M:M). Иногда эти типы связей называются степенью связи. Кроме степени связи в ИЛМ для характеристики связи между разными сущностями надо указывать так называемый *класс принадлежности*, который показывает, может ли отсутствовать связь объекта данного класса с каким-либо объектом другого класса. Класс принадлежности сущности должен быть либо обязательным, либо необязательным.

Анализ определенных объектов и атрибутов предметной области позволяет выделить объекты (сущности, таблицы) проектируемой базы данных и, приняв решение о создании реляционной базы данных, построить ее инфологическую модель на языке "Таблицы-связи".

С учетом вышеперечисленных особенностей построения ИЛМ к основным объектам можно отнести следующие объекты данных:

1. Объект Библиотекари содержит: данные о сотрудниках библиотеки, которые выдают литературу читателям (Фамилия, Имя, Отчество, Адрес, Телефон);
2. Объект Читатели содержит: данные о читателях, пользующихся услугами библиотеки (Номер билета, Фамилия, Имя, Отчество, Адрес, Телефон);
3. Объект Издание содержит: подробные данные о литературных изданиях (большая часть данных этой таблицы хранятся во вспомогательных 13-ти объектах-справочниках);
4. Объект Переплеты содержит: идентификатор конкретного переплета (экземпляра издания) и признак есть ли он в данный момент в наличии;

5. Объект Размещение содержит: данные о месте размещения (во вспомогательном объекте) конкретных переплетов и датах размещения и изъятия их;
6. Объект Заказы содержит: данные о заказах литературы, читателями;
7. Объект Выдача содержит: данные о выдаваемых на руки читателям переплетов, а так же сроках их возврата.

А так же к вспомогательным объектам ИЛМ можно отнести следующие объекты данных:

- О создателях издания - для хранения сведений об основных людях, принимавших участие в подготовке рукописи издания (авторах, составителях, титульных редакторах, переводчиках и художниках). Такое объединение допустимо, так как данные о разных создателях выбираются из одного домена (фамилия и имена) и исключает дублирование данных. Так как фамилия и имена (инициалы) создателя могут быть достаточно громоздкими (М.Е. Салтыков-Щедрин, Франсуа Рене де Шатобриан, Остен Жюль Жан-Батист Ипполит и т.п.) и будут многократно встречаться в разных изданиях, то их целесообразно нумеровать и ссылаться на эти номера. Для этого вводится целочисленный атрибут "Код_автора", который будет автоматически наращиваться на единицу при вводе в базу данных нового автора. Аналогично создаются объекты: Редакторы, Составители, Художники, Переводчики.
- Объект Издательства содержит: данные о издательстве.
- Объект Города содержит: данные о городах издательства.
- Объект Заглавия содержит: данные о заглавии издания

Выделение этой сущности позволит сократить объем данных и снизить вероятность возникновения противоречивости (исключается необходимость ввода длинных текстовых названий для различных томов собраний сочинений, повторных изданий, учебников и т.п.).

- Объект Вид_издания содержит: данные о виде издания.
- Объект Характеры содержит: данные о характере переиздания.

- Объект Языки содержит: данные о языке издания.
Кроме названия языка хранится его общепринятое сокращение (англ., исп., нем., фр.), если оно существует.
- Объект Места содержит: данные о месте хранения конкретного переплета (Номер_комнаты, Номер_стеллажа, Номер_полки).
- Объект Предметная рубрика содержит: данные о предметной рубрике.
- Объект Аннотации содержит: данные об аннотациях.
- Объект Пароли содержит: данные о паролях библиотекарей и пользователей.

На основании структурной схемы деятельности библиотеки и описанных выше свойств объектов, строится ИЛМ проектируемой БД.

ER-диаграмма «Библиотеки» изображена на рисунке 2.1.

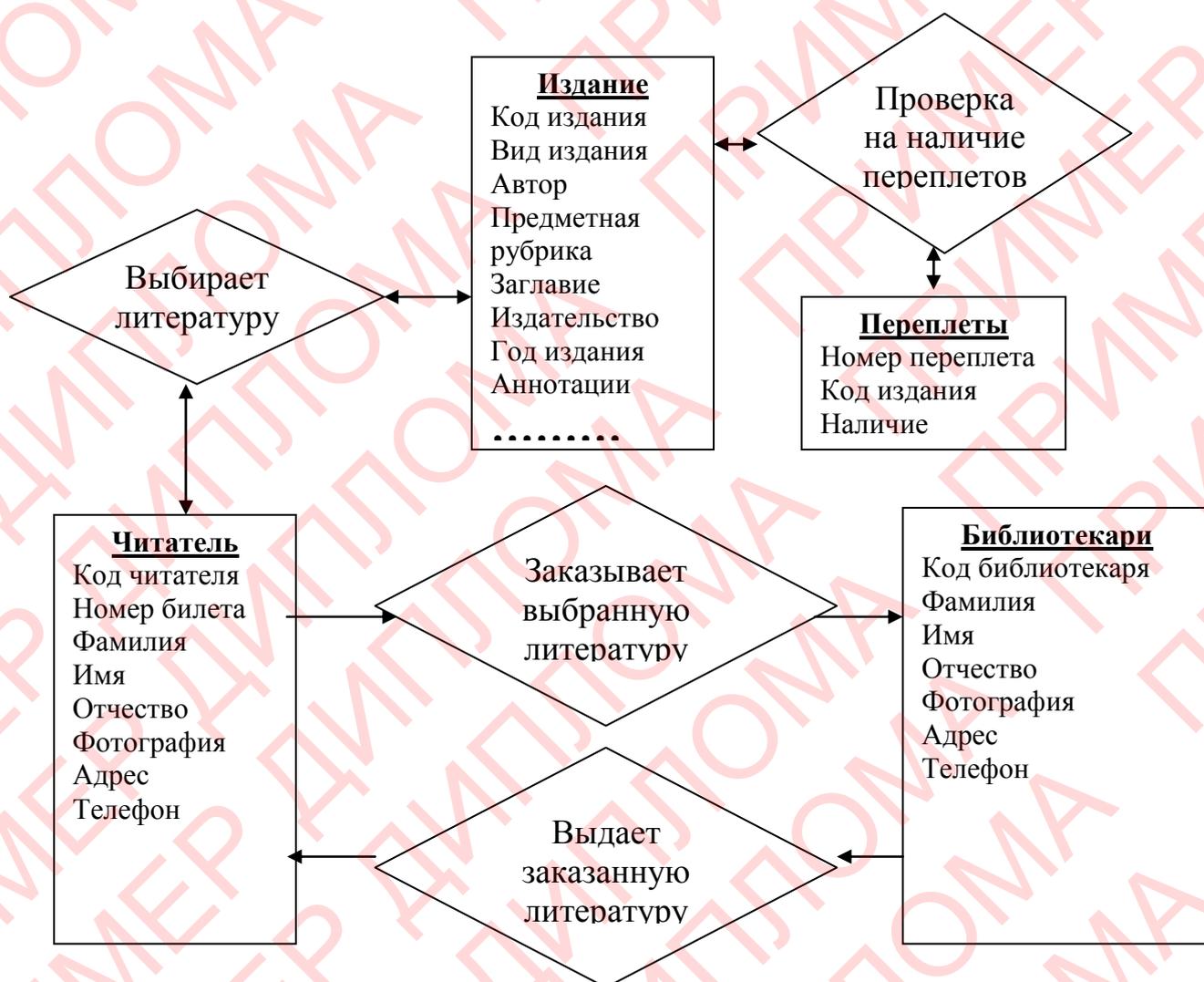


Рисунок 2.1 - ER-диаграмма «Библиотеки».

2.6 Проектирование логической структуры базы данных

Задачей следующей стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Другими словами, модель предметной области разрабатываемой системы должна быть представлена в терминах модели данных концептуального уровня выбранной конкретной СУБД. Эту стадию называют логическим или даталогическим проектированием базы данных, а ее результатом является концептуальная схема базы данных, включающая определение всех информационных элементов и связей, в том числе задание типов, характеристик и имен.

Задача этого этапа - преобразование ER-диаграммы в реляционную схему. Реляционная база данных - база данных, состоящая из отношений. Здесь вся информация, доступная пользователю, организована в виде таблиц, обычно имеющих уникальные имена, состоящих из строк и столбцов, на пересечении которых содержатся значения данных, а операции над данными сводятся к операциям над этими таблицами.

Основными понятиями, определяющими реляционную модель, являются:

Домен - совокупность значений, из которых берутся значения соответствующих атрибутов определённого отношения (с точки зрения программирования - это тип данных, определяемый системой или пользователем).

Первичный ключ - это столбец или некоторое подмножество столбцов, которое уникально, т.е. единственным образом определяют строки.

Внешний ключ - это столбец или подмножество одной таблицы, который может служить в качестве первичного ключа для другой таблицы. Внешний ключ таблицы является ссылкой на первичный ключ *другой* таблицы.

Все создаваемые таблицы должны отвечать следующим требованиям:

- данные в ячейках должны быть структурно неделимыми
- данные в одном столбце должны быть одного типа
- каждый столбец должен быть уникальным
- столбцы размещаются в произвольном порядке
- строки размещаются в произвольном порядке
- столбцы имеют уникальные наименования

Реляционный подход, в основе которого лежит принцип разделения данных и связей, обеспечивает, с одной стороны, независимость данных, а с другой - более простые способы хранения и обновления.

Каждая сущность ER-диаграммы превращается в таблицу, каждое свойство становится атрибутом - столбцом соответствующей таблицы.

Далее необходимо преобразовать связи во внешние ключи, используя различные связи: «многие ко многим», «один ко многим» и т.д. Таблицы могут находиться также в разных формах: первой, второй и третьей.

2.7 Проектирование физической структуры базы данных

Стадия физического проектирования БД в общем случае включает:

- выбор способа организации БД
- разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня
- описание отображения концептуальной схемы во

внутреннюю

База данных будет состоять из пяти таблиц: **Библиотека, Клиенты, Поставщики, Заказы, Закупки литературы.**

Таблицы следует создавать в режиме конструктора следующим образом:

1. Выбрать закладку **Таблица**, нажать **Создать**
2. В появившемся диалоговом окне нажать **Создание таблицы в режиме конструктора**

Дальше были созданы в режиме конструктора следующие таблицы:

- «Читатели» с такие полями:

№ п.п	Имя	Тип	Размер	Обязательное	Индексированное
1	Код_читателя	Счетчик	Длинное целое	да	Да (Совпадения не доп-ся)
2	Номер_билета	Текстовый	10	да	Да (Совпадения не доп-ся)
3	Фамилия	Текстовый	250	да	Да (Допускаются совпадения)
4	Имя	Текстовый	250	да	Да (Допускаются совпадения)
5	Отчество	Текстовый	250	да	Да (Допускаются совпадения)
6	Адрес	Текстовый	250	да	Да (Допускаются совпадения)
7	Телефон_дом	Текстовый	20	нет	Да (Допускаются совпадения)
8	Телефон_моб	Текстовый	20	нет	Да (Совпадения не доп-ся)
9	Фотография	Поле объекта OLE		нет	

Существуют следующие типы данных:

текстовый - тип данных, используемый для хранения простого неформатированного текста, число символов в котором не должно превышать 255;

поле MEMO - специальный тип данных, применяемый для хранения больших объемов текста (до 65 535 символов);

числовой - тип данных для хранения чисел;

дата/время - тип данных для хранения значений даты и времени;
денежный - тип данных для хранения денежных значений (длина поля 8 байт);

счётчик - специальный тип данных, используемый для автоматической нумерации записей;

логический - для хранения логических данных, которые могут иметь одно из двух возможных значений (**Да** или **Нет**);

поле объекта OLE - специальный тип данных, предназначенный для хранения объектов OLE (электронных таблиц Microsoft Excel, документов Microsoft Word, звукозаписей и др.);

гиперссылка - специальное поле для хранения адресов URL Web-объектов;

мастер подстановок - тип данных, запускающий мастер подстановок, что позволяет выбирать данные из раскрывающегося списка, а не вводить их в поле вручную.

Ключевое поле - Код_читателя.

Ключевое поле - это поле с уникальными записями. Таблицы связываются по ключам; ключ может состоять из одного или нескольких полей.

Имя поля	Тип данных	Иса
Код_читателя	Счетчик	
Номер_билета	Текстовый	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Адрес	Текстовый	
Телефон_дом	Числовой	
Телефон_моб	Текстовый	
Фотография	Поле объекта	

Свойства поля

Общие Подстановка

Подпись

Обязательное поле Нет

Рисунок 2.2 - Структура таблицы «Читатели»

- «Библиотекари» с такие полями:

Ключевое поле - Код_библиотекаря.

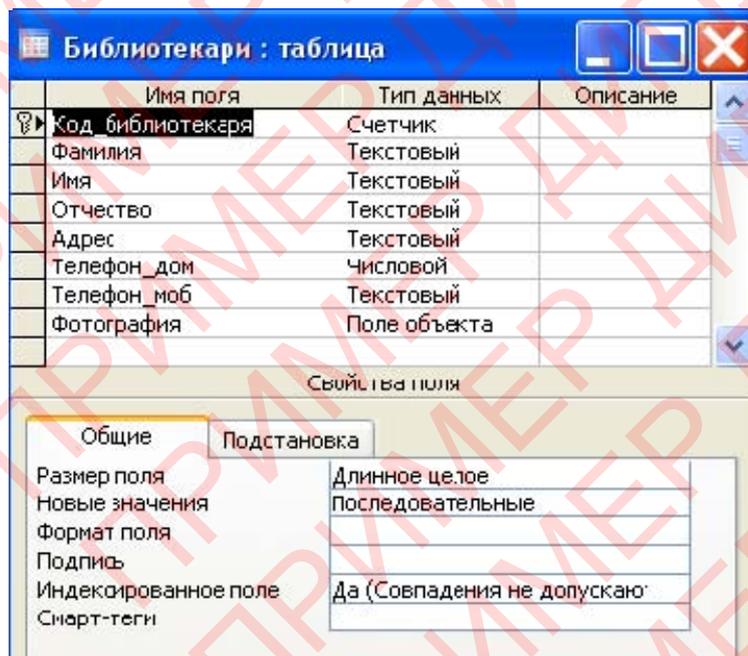


Рисунок 2.3 - Структура таблицы «Библиотекари»

- «Издание» с такие полями:

Ключевое поле - Код_издания. Поля: Код_заглавия, Код_вида_издания, Код_издательства, Год_издания, Код_автора, Код_аннотации, Код_переводчика, Код_редактора, Код_составителя, Код_художника, Код_языка, Код_характера_переиздания, Код_предметной_рубрики – внешние ключи для связи со справочными таблицами.

Имя поля	Тип данных	Описание
Код_издания	Счетчик	
Код_заглавия	Числовой	
Код_вида_издания	Числовой	
Номер_тома	Числовой	
Авторский_знак	Текстовый	
Библиотечн_шифр	Текстовый	
Повторность	Числовой	
Код_издательства	Числовой	
Год_издания	Дата/время	
Код_автора	Числовой	
Код_аннотации	Числовой	
Код_переводчика	Числовой	
Код_редактора	Числовой	
Код_составителя	Числовой	
Код_художника	Числовой	
Код_языка	Числовой	
Код_характера_переиздан	Числовой	
Индекс ISBN	Текстовый	
Код_предметной_рубрики	Числовой	
Фото_обложки	Поле объекта	
Цена	Денежный	
Дата_приобретения	Дата/время	
Количество_общее	Числовой	
Количество_наличия	Числовой	

Свойства поля

Общие Подстановка

Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допуск:
Смарт-теги	

Рисунок 2.4 - Структура таблицы «Издание»

- «Переплеты» с такие полями:

Ключевое поле - Код_номера_переплета. Поле Код_издания – внешний ключ для связи с другой таблицей.

Имя поля	Тип данных	Описание
Код_номера_переплета	Счетчик	
Номер_переплета	Текстовый	
Код_издания	Числовой	
Наличие	Логический	

Свойства поля

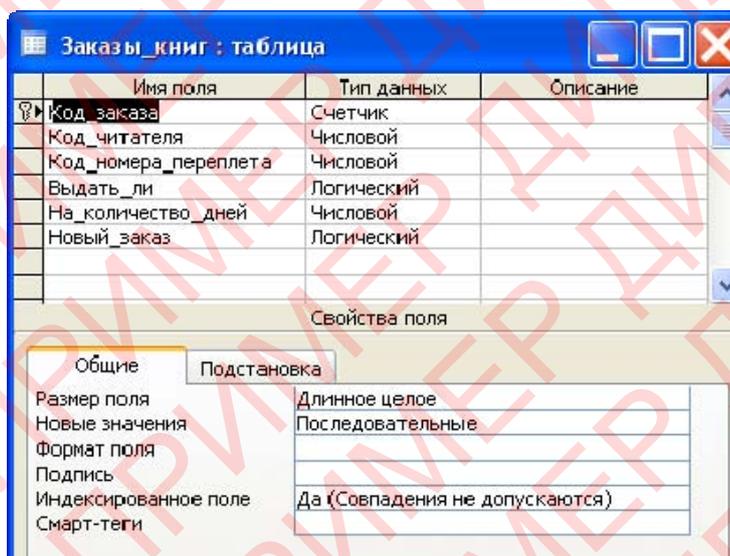
Общие Подстановка

Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допус:
Смарт-теги	

Рисунок 2.5 - Структура таблицы «Переплеты»

- «Заказы_книг» с такие полями:

Ключевое поле - Код_заказа. Поля Код_номера_переплета, Код_читателя – внешние ключи для связи с другими таблицами.



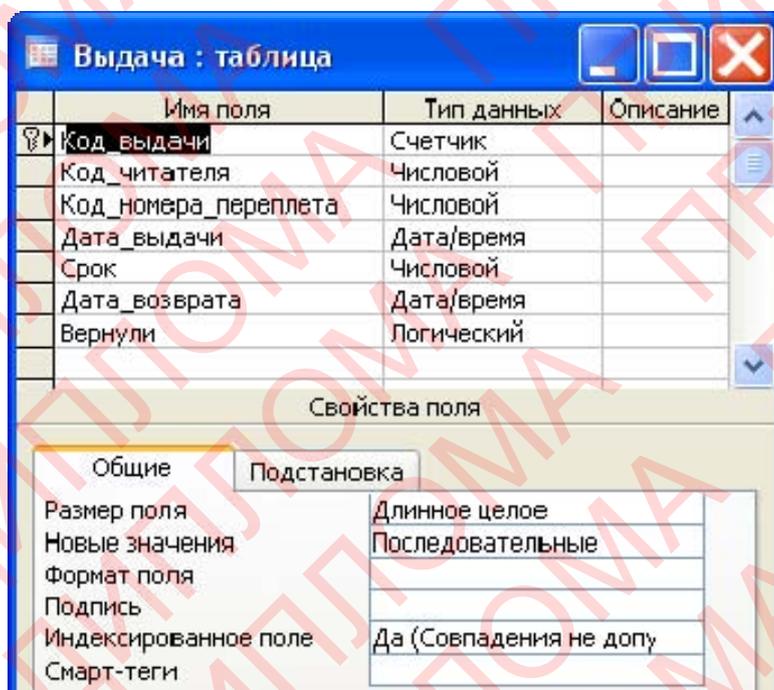
Имя поля	Тип данных	Описание
Код_заказа	Счетчик	
Код_читателя	Числовой	
Код_номера_переплета	Числовой	
Выдать_ли	Логический	
На_количество_дней	Числовой	
Новый_заказ	Логический	

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)
Смарт-теги	

Рисунок 2.6 - Структура таблицы «Заказы_книг»

- «Выдача» с такие полями:

Ключевое поле - Код_выдачи. Поля Код_номера_переплета, Код_читателя – внешние ключи для связи с другими таблицами.



Имя поля	Тип данных	Описание
Код_выдачи	Счетчик	
Код_читателя	Числовой	
Код_номера_переплета	Числовой	
Дата_выдачи	Дата/время	
Срок	Числовой	
Дата_возврата	Дата/время	
Вернули	Логический	

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допу
Смарт-теги	

Рисунок 2.7 - Структура таблицы «Выдача»

- «Размещение» с такие полями:

Ключевое поле - Код_размещения. Поля Код_номера_переплета, Код_места – внешние ключи для связи с другими таблицами.

Имя поля	Тип данных	Описание
Код_размещения	Счетчик	
Код_места	Числовой	
Код_номера_переплета	Числовой	
Дата_размещения	Дата/время	
Дата_изъятия	Дата/время	

Свойства поля

Общие Подстановка

Размер поля Длинное целое

Новые значения Последовательные

Формат поля

Подпись

Индексированное поле Да (Совпадения не до

Смарт-теги

Рисунок 2.8 - Структура таблицы «Размещение»

- «Места» с такие полями:

Ключевое поле - Код_места.

Имя поля	Тип данных	Описание
Код_места	Счетчик	
Номер_комнаты	Числовой	
Номер_стеллажа	Числовой	
Номер_полки	Числовой	

Свойства поля

Общие Подстановка

Размер поля Длинное целое

Новые значения Последовательные

Формат поля

Подпись

Индексированное поле Да (Совпадения не допус

Смарт-теги

Рисунок 2.9 - Структура таблицы «Места»

- «Пароли» с такие полями:

Ключевое поле - Код_пароля. Поля Код_читателя, Код_библиотекаря – внешние ключи для связи с другими таблицами.



Рисунок 2.10 - Структура таблицы «Пароли»

По вышеприведённому примеру создаются структуры остальных вспомогательных таблиц. Это построение пустых таблиц БД.

2.8 Создание датологической модели базы данных

1. Щелкнуть по пиктограмме - **Схема данных**. Появится окно **Добавление таблицы**.
2. Выделить таблицы **все таблицы** и щёлкнуть по кнопке **Добавить**. В окне **Схема данных** появится условный вид этих таблиц.
3. Поставить мышку на имя поля **Код_издания** в таблице **Издание** и, не отпуская кнопку мышки, перетащить её на поле **Код_издания** в таблице **Переплеты**. Отпустить мышку, и в появившемся диалоговом окне **Связи** включить значок **Обеспечение целостности данных**, затем **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. Щёлкнуть по кнопке **Создать**, появится связь **один-ко-многим**. По такому же алгоритму связать остальные таблицы так, чтобы получилась схема БД, показанная на рисунке 2.11.

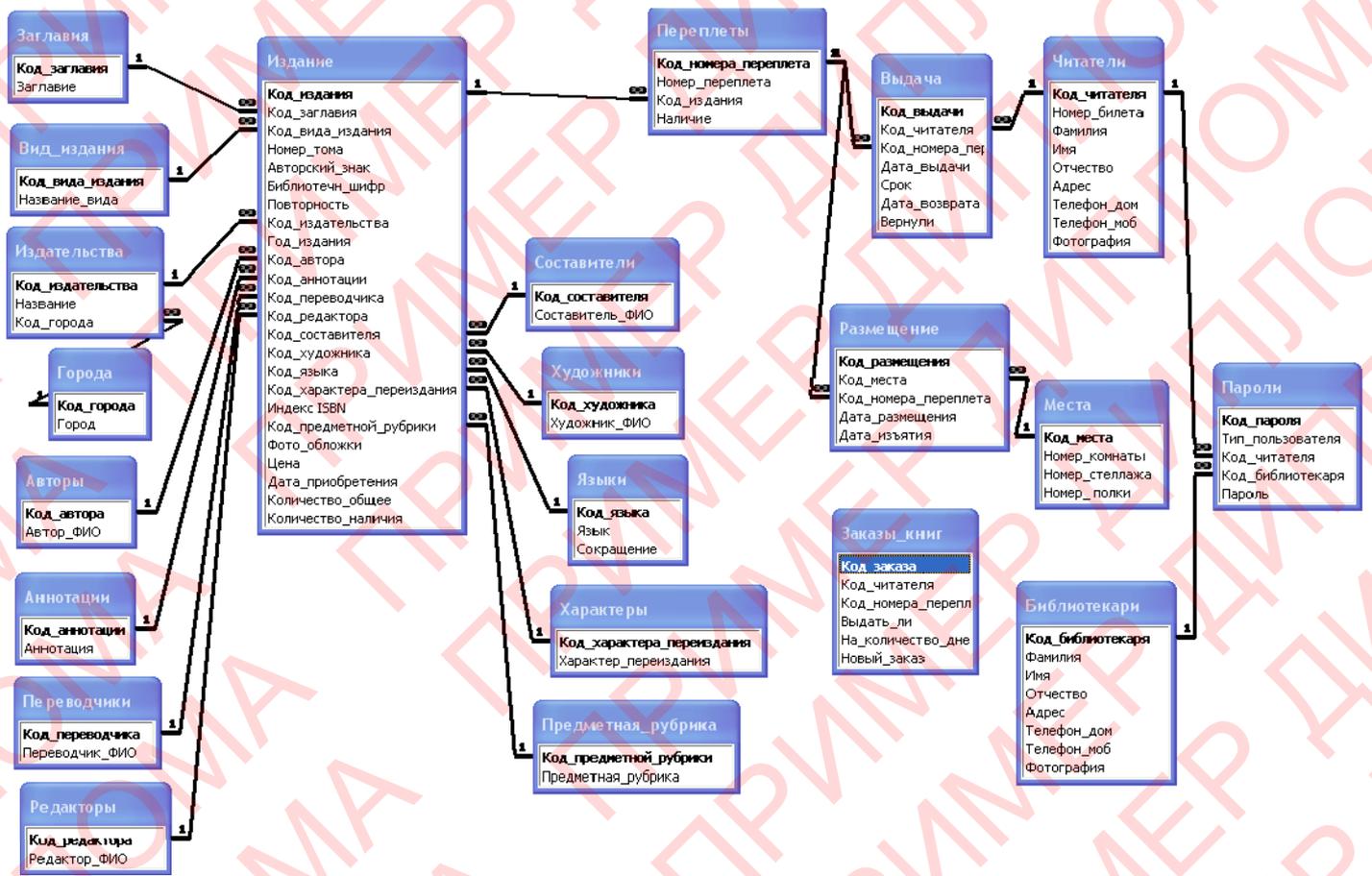


Рисунок 2.11 - Датологическая модель ИС
(связи между таблицами базы данных Библиотека)

2.9 Алгоритмизация приложения

При проектировании рабочей модели системы, с учетом информационных потребностей пользователя, был разработан общий алгоритм работы программы, который показан на рис. 2.12. Из этого рисунка хорошо просматриваются функциональные возможности системы.

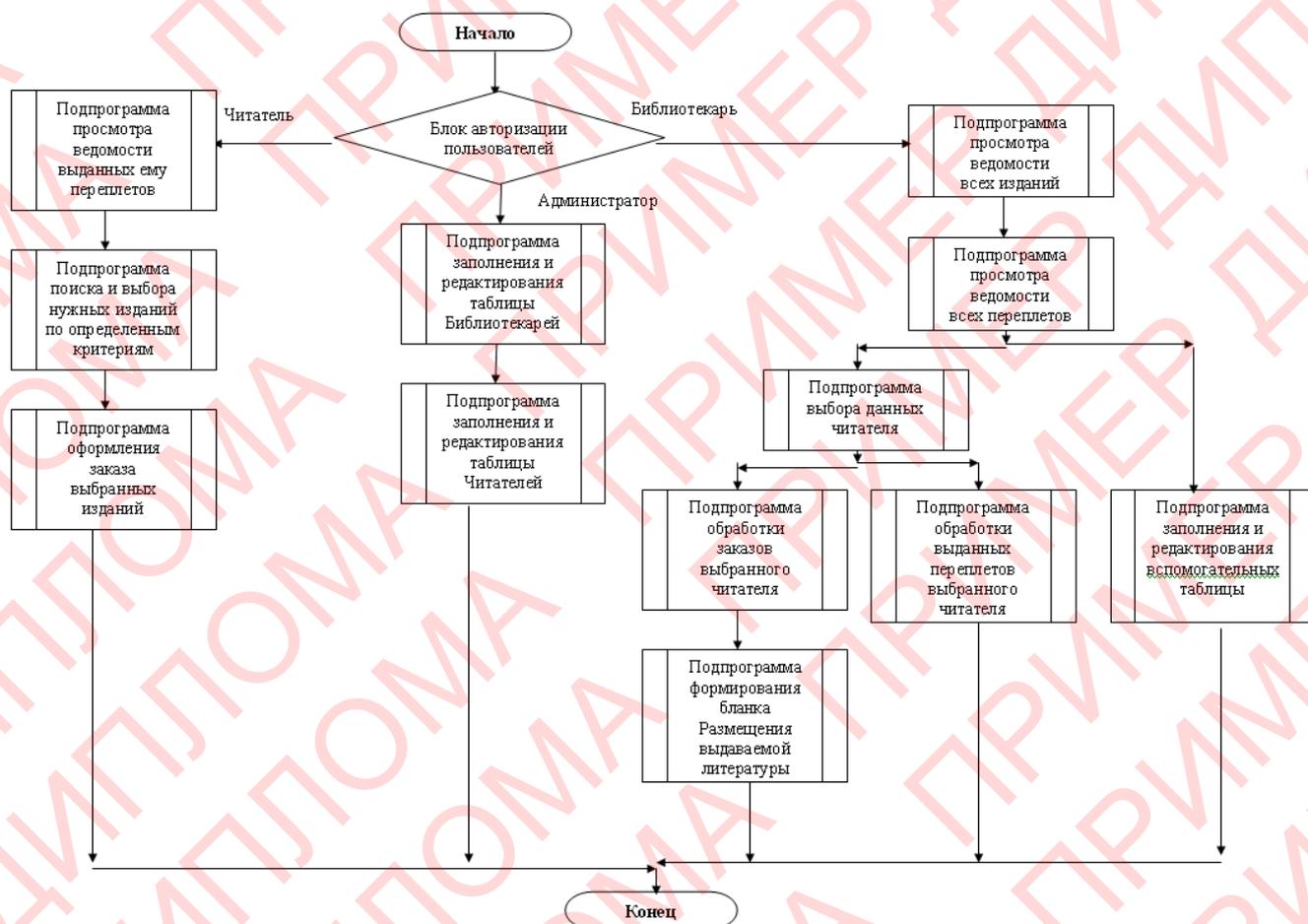


Рисунок 2.12 - Общий алгоритм работы программы

Эти возможности реализуются через отдельные блоки подпрограмм.

При входе в САБ, пользователь выбирает «Роль пользователя».

Если пользователь выбрал «Администратор», то откроется окно в котором можно выполнить либо подпрограмму заполнения и редактирования таблицы Библиотекарей, либо подпрограмму заполнения и редактирования таблицы Читателей.

Если пользователь выбрал «Читатель», то откроется окно в котором вначале выполнится подпрограмма просмотра ведомости выданных ему переплетов, а затем читатель может воспользоваться подпрограммой поиска и выбора нужных изданий по определенным критериям, после чего он должен выполнить подпрограмму оформления заказа выбранных изданий.

Если пользователь выбрал «Библиотекарь», то откроется окно в котором вначале можно выполнить подпрограмму просмотра ведомости всех изданий и можно выполнить подпрограмму просмотра ведомости всех переплетов или выполнить подпрограмму заполнения и редактирования вспомогательных таблиц. Затем библиотекарь выполняет подпрограмму выбора данных читателя и далее подпрограмму обработки заказов выбранного читателя или выполняет подпрограмму обработки выданных переплетов выбранного читателя, потом выполняет подпрограмму формирования бланка Размещения выдаваемой литературы.

Функциональные особенности подпрограмм заключаются в следующем:

1. Подпрограмма заполнения и редактирования таблицы Библиотекарей - предоставляет администратору готовые формы для ввода данных о библиотекарях, которые заносятся в таблицу «Библиотекари».
2. Подпрограмма заполнения и редактирования таблицы Читателей - предоставляет администратору готовые формы для ввода данных о читателях, которые заносятся в таблицу «Читатели».
3. Подпрограмма просмотра ведомости выданных читателю переплетов - предоставляет читателю готовые формы для просмотра ведомости выданных ему переплетов.
4. Подпрограмма поиска и выбора нужных изданий по определенным критериям – позволяет читателю производить поиск нужной ему литературы по определенным критериям поиска и затем выбирать нужные переплеты из предложенного списка изданий.

5. Подпрограмма оформления заказа выбранных изданий – оформляет заказ литературы и записывает в базу данных отобранную читателем литературу.
6. Подпрограмма просмотра ведомости всех изданий – позволяет библиотекарю просмотреть все данные обо всех изданиях, которые есть в библиотеке.
7. Подпрограмма просмотра ведомости всех переплетов – позволяет библиотекарю просмотреть данные обо всех переплетах, которые есть в библиотеке.
8. Подпрограмма заполнения и редактирования вспомогательных таблицы – позволяет библиотекарю заполнять и редактировать все вспомогательные таблицы.
9. Подпрограмма обработки выданных переплетов выбранного читателя – позволяет библиотекарю отмечать возвращаемые переплеты в библиотеку и списывать их с читателей.
10. Подпрограмма обработки заказов выбранного читателя - позволяет библиотекарю выдавать заказанную литературу читателю и отмечать сроки, на которые выдается литература.
11. Подпрограмма формирования бланка Размещения выдаваемой литературы – позволяет формировать бланк с данными о размещении выдаваемой литературы, для быстрого их поиска в архивах библиотеки.

2.10 Программирование, создание приложения

В случае, когда на предыдущем этапе был получен детально разработанный алгоритм, составление программы на выбранном для программирования языке сводится к переводу этого алгоритма на язык программирования.

Основные трудности и, следовательно, причины ошибок на этом этапе заключаются, во-первых, в необходимости знания всех требований и ограничений выбранного языка программирования и, во-вторых, в необходимости постоянного внимания ко многим деталям языка, которые приходится учитывать в ходе написания программы. Если предыдущий этап был выполнен некачественно и алгоритм представлен недостаточно детально, то его доводку придется выполнять «на ходу», во время программирования. Это затруднит процесс программирования-перевода и поведет к возникновению дополнительных ошибок в программе. Чем более процесс программирования будет походить на перевод, чем более механическим будет такой перевод, тем более легким будет составление программы и тем меньше возникнет ошибок на этом этапе, самом щедрым на ошибки.

После составления программы проводится ее проверка для обнаружения и исправления ошибок, внесенных на этом этапе. Если при проверке обнаруживаются ошибки, допущенные на предыдущем этапе, то соответствующие исправления вносятся и в алгоритм, поскольку к нему еще придется обращаться на следующих этапах, и тексты алгоритма и программы должны соответствовать друг другу.

Данное приложение было создано в визуальной среде проектирования Borland Developer Studio 2006 на объектно-ориентированном языке программирования Delphi 2006.

Был разработан интуитивно понятный интерфейс программы, состоящей из целого ряда форм, которые обеспечивают работу с данными основных и вспомогательных таблиц.

Первая форма программы – это заставка, которая появляется на экране на несколько секунд, а затем появляется главная форма программы.

Прототип первой формы имеет следующий вид:

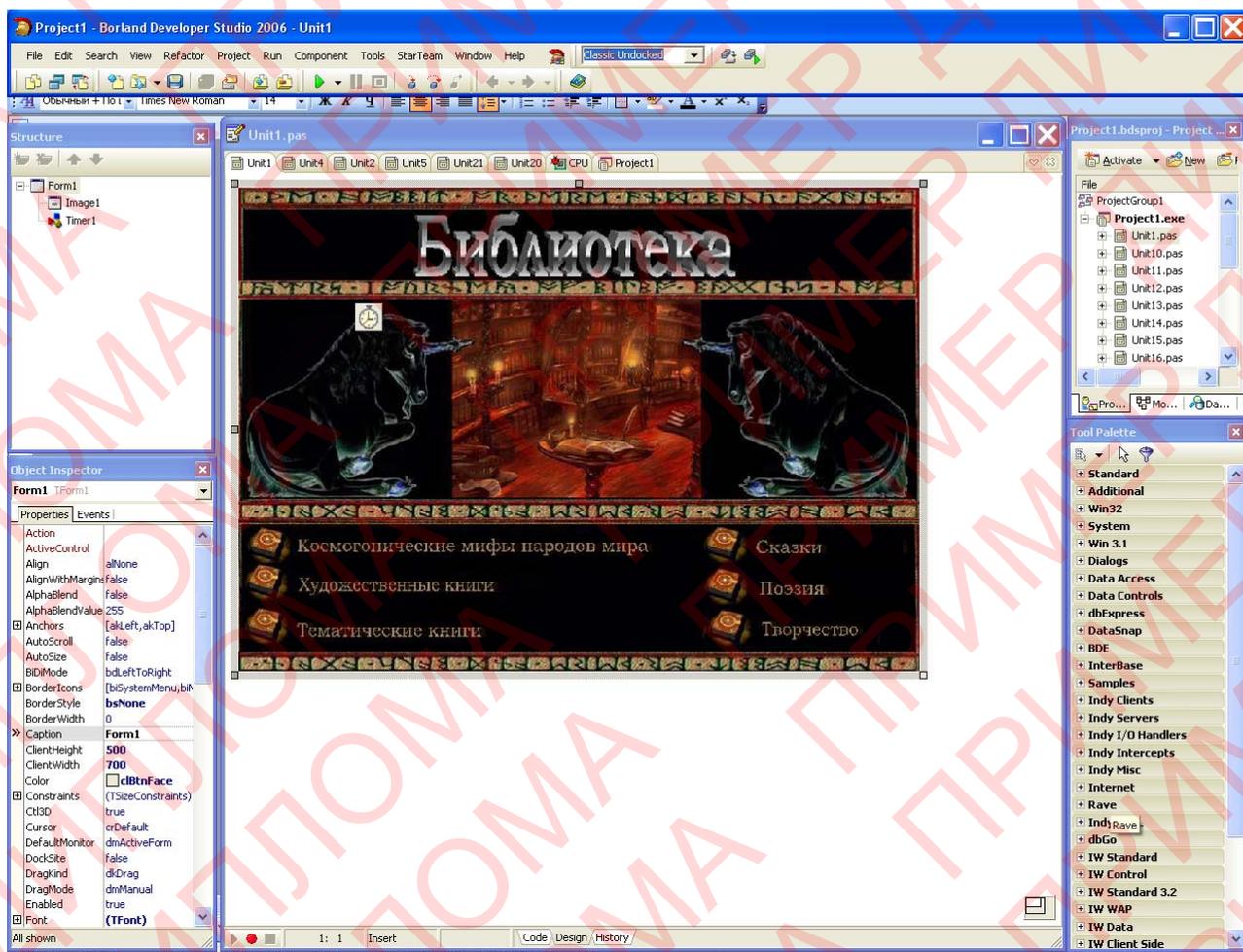


Рисунок 2.13 - Прототип первой формы

Ее описание сохранено в файле «Unit1.dfm», а код, реализующий ее действия, сохранен в файле «Unit1.pas».

Содержимое файла «Unit1.pas»:

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, DB, ADODB, ExtCtrls, jpeg;  
type  
  TForm1 = class(TForm)  
    Image1: TImage;
```

```

Timer1: TTimer;
procedure FormActivate(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var Form1: TForm1;
implementation
uses Unit2;
{$R *.dfm}
procedure TForm1.FormActivate(Sender: TObject);
begin
  Timer1.Enabled:=true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Form2.Show;
  Form1.Visible:=false;
  Timer1.Enabled:=false;
end;

end.

```

Задержка заставка и открытие главной формы – реализовано с помощью компонента Timer1.

Прототип формы по авторизации пользователей имеет следующий вид:

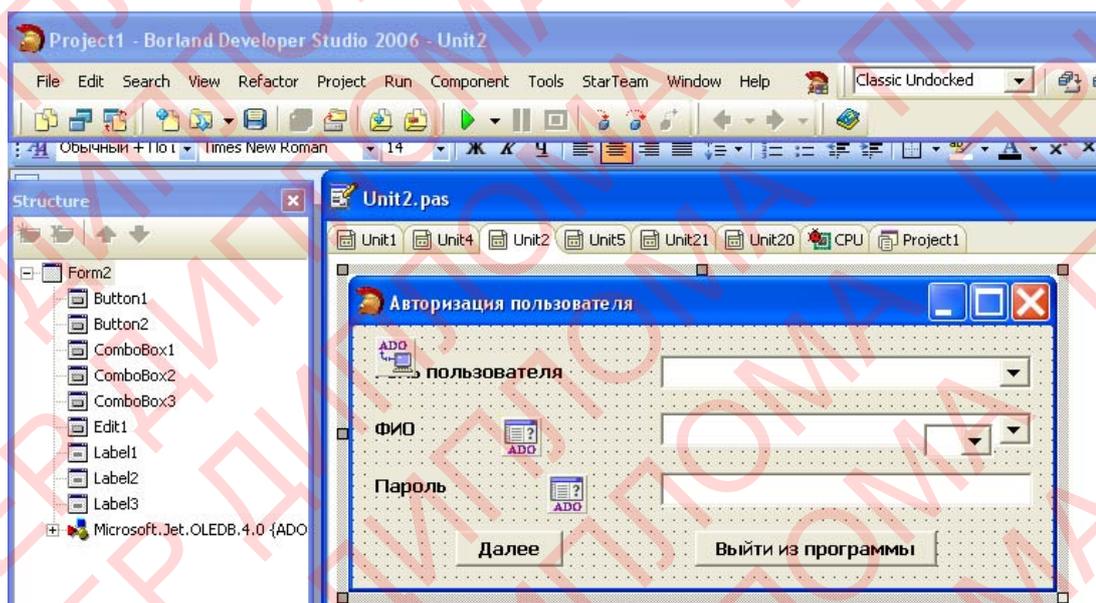


Рисунок 2.14 - Прототип формы авторизации пользователей

Ее описание сохранено в файле «Unit2.dfm», а код, реализующий ее действия, сохранен в файле «Unit2.pas».

Содержимое файла «Unit2.pas»:

```
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls;

type TPolz = record
  rol:string;
  Kod:integer;
  Danie:string;
end;

type
  TForm2 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    Edit1: TEdit;
    Button1: TButton;
    Button2: TButton;
    ADOQuery1: TADOQuery;
    ADOQuery2: TADOQuery;
    ADOConnection1: TADOConnection;
    ComboBox3: TComboBox;
    procedure Button2Click(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure ComboBox1Click(Sender: TObject);
    procedure ComboBox2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    s1,s2,pass:string;
  public
    { Public declarations }
    Polz:TPolz;
  end;

var Form2: TForm2;
implementation
uses Unit3, Unit4, Unit5;
{$R *.dfm}

procedure TForm2.Button1Click(Sender: TObject);
```

```

begin
ADOQuery2.Active:=false;
ADOQuery2.SQL.Clear;
if ComboBox1.Text='Администратор'
then
begin
ADOQuery2.SQL.Add('select * from Администратор where Администратор=1;');
ADOQuery2.Active:=true;
Polz.rol:='Администратор';
Polz.Danie:="";
pass:=ADOQuery2.FieldByName('Пароль_Администратора').AsString;
if pass=Edit1.Text then
begin
Form3.Show;
Form2.Close;
end
else
begin
MessageDlg('Вы ввели не верный пароль Администратора!',
mtInformation,[mbOk],0);
Edit1.Clear;
Edit1.SetFocus;
end;
end
else
if ComboBox1.Text='Библиотекарь'
then
begin
ComboBox3.ItemIndex:=ComboBox2.ItemIndex;
ADOQuery2.SQL.Add('select * from Пароли where Тип_пользователя=0'+
' and Код_библиотекаря='+ComboBox3.Text);
ADOQuery2.Active:=true;
Polz.rol:='Библиотекарь';
Polz.Kod:=StrToInt(ComboBox3.Text);
Polz.Danie:=ComboBox2.Text;
pass:=ADOQuery2.FieldByName('Пароль').AsString;
if pass=Edit1.Text then
begin
Form4.Show;
Form2.Close;
end
else
begin
MessageDlg('Вы ввели не верный пароль Библиотекаря!',
mtInformation,[mbOk],0);
Edit1.Clear;
Edit1.SetFocus;
end;
end
else
if ComboBox1.Text='Читатель'
then

```

```

begin
  ComboBox3.ItemIndex:=ComboBox2.ItemIndex;
  ADOQuery2.SQL.Add('select * from Пароли where Тип_пользователя=1'+
    ' and Код_читателя='+ComboBox3.Text);
  ADOQuery2.Active:=true;
  Polz.rol:='Читатель';
  Polz.Kod:=StrToInt(ComboBox3.Text);
  Polz.Danie:=ComboBox2.Text;
  pass:=ADOQuery2.FieldByName('Пароль').AsString;
  if pass=Edit1.Text then
    begin
      Form5.Show;
      Form2.Close;
    end
  else
    begin
      MessageDlg('Вы ввели не верный пароль Читателя!',
        mtInformation,[mbOk],0);
      Edit1.Clear;
      Edit1.SetFocus;
    end;
  end;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
  if MessageDlg('Вы уверены что хотите выйти из программы? ',
    mtInformation,[mbYes,mbNo],0)=mrYes
  then
    begin
      if ADOConnection1.Connected then ADOConnection1.Close;
      Application.Terminate;
    end;
  end;

procedure TForm2.ComboBox1Change(Sender: TObject);
begin
  ComboBox1.Text:=s1;
end;

procedure TForm2.ComboBox1Click(Sender: TObject);
var i:integer;
begin
  ComboBox2.Enabled:=true;
  ComboBox2.Items.Clear;
  ComboBox3.Items.Clear;
  ComboBox2.Text:='';
  s1:=ComboBox1.Text;

  if ComboBox1.Text='Читатель'
  then
    begin

```

```

// Выполняем запрос к базе данных на выборку нужных данных
ADOQuery1.Active:=false;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('select * from Читатели order by Номер_билета;');
ADOQuery1.Active:=true;
ADOQuery1.First;
// Выбираем нужные данные из базы данных
for I := 1 to ADOQuery1.RecordCount do
begin
  ComboBox2.Items.Add(ADOQuery1.FieldByName('Номер_билета').AsString);
  ComboBox3.Items.Add(ADOQuery1.FieldByName('Код_читателя').AsString);
  ADOQuery1.Next;
end;
Label2.Caption:='Абонементский № читателя';
ComboBox2.SetFocus;
end
else
if ComboBox1.Text='Библиотекарь'
then
begin
  // Выполняем запрос к базе данных на выборку нужных данных
  ADOQuery1.Active:=false;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('select * from Библиотекари order by Фамилия;');
  ADOQuery1.Active:=true;
  ADOQuery1.First;
  // Выбираем нужные данные из базы данных
  for I := 1 to ADOQuery1.RecordCount do
  begin
    ComboBox2.Items.Add(ADOQuery1.FieldByName('Фамилия').AsString+' '+
      ADOQuery1.FieldByName('Имя').AsString+' '+
      ADOQuery1.FieldByName('Отчество').AsString);
    ComboBox3.Items.Add(ADOQuery1.FieldByName('Код_библиотекаря').AsString);
    ADOQuery1.Next;
  end;
  Label2.Caption:='ФИО Библиотекаря';
  ComboBox2.SetFocus;
end
else
if ComboBox1.Text='Администратор'
then
begin
  Label2.Caption:='';
  ComboBox2.Enabled:=false;
  Edit1.SetFocus;
end;
end;

procedure TForm2.ComboBox2Change(Sender: TObject);
begin
  ComboBox2.Text:=s2;
end;

```

```
procedure TForm2.ComboBox2Click(Sender: TObject);
begin
  s2:=ComboBox2.Text;
  Edit1.SetFocus;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  ADOConnection1.Open;
end;

end.
```

Остальные окна программы созданы по тому же модульному принципу
(в виде форм).

3. Экспериментальная часть

3.1 Объект испытаний

3.1.1 Наименование

Наименование – «Автоматизированная система библиотекаря».

3.1.2 Область применения

Автоматизированная система библиотекаря предназначена для использоваться в библиотеках, для освобождения сотрудников библиотеки от ряда рутинных работ по подготовке картотек, изданий, списков, а так же для получения читателем рекомендаций по выбору книг, т.е. для автоматизированной обработки новых поступлений в библиотеку, упрощения поиска и анализа библиотечной информации.

3.2 Цель испытаний

Цель проведения испытаний - проверка соответствия характеристик разработанной автоматизированной системы библиотекаря функциональным требованиям, изложенным в документе Техническое задание.

3.3 Перечень документов, предъявляемых на испытания

Документом, предъявляемым на испытания, является моя пояснительная записка к дипломному проекту, которая состоит из разделов, соответствующим следующим программным документам:

- Техническое задание;
- Программу и методику испытаний;
- Руководство программиста;
- Руководство пользователя;

3.4 Объем испытаний

3.4.1 Перечень этапов испытаний

Испытания проводятся в два этапа:

1. ознакомительный;
2. испытания.

3.4.1.1 Перечень проверок, проводимых на 1 этапе испытаний

Перечень проверок, проводимых на 1 этапе испытаний, должен включать в себя:

1. проверку комплектности программной документации;
2. проверку комплектности состава программных средств.

3.4.1.2 Перечень проверок проводимых на 2 этапе испытаний

Перечень проверок проводимых на 2 этапе испытаний, должен включать в себя:

1. проверку соответствия технических характеристик программы;
2. проверку степени выполнения требований функционального назначения программы.

3.4.2 Количественные и качественные характеристики, подлежащие оценке

3.4.2.1. Количественные характеристики, подлежащие оценке

В ходе проведения приемо-сдаточных испытаний оценке подлежат количественные характеристики, такие как:

- а) комплектность программной документации.
- б) комплектность состава технических и программных средств.

3.4.2.2 Качественные характеристики, подлежащие оценке

В ходе проведения приемо-сдаточных испытаний оценке подлежат качественные (функциональные) характеристики программы. Проверке подлежит возможность выполнения программой перечисленных ниже функций:

- Защищенная паролем авторизация пользователей при запуске программы;
- Корректный выход из программы, с предупреждающим сообщением.
- Для роли пользователя «администратор» главное окно должно позволять:
 - Вводить, редактировать, удалять основные данные о новых библиотекарях, а также устанавливать им пароли;
 - Вводить, редактировать, удалять основные данные о новых читателях, а также устанавливать им пароли;
- Для роли пользователя «читатель» главное окно должно позволять:
 - Выбор необходимых изданий, с детальным просмотром всех его данных
 - Автоматизированный поиск изданий по определенным критериям

- Из списка отобранных изданий, занесение нужной литературы в список заказа
- Для роли пользователя «библиотекарь» главное окно должно позволять:
 - Выбор читателя по его абонементскому номеру, просмотр его анкеты, списка ранее выдаваемых ему книг и списка заказанных им книг
 - Просмотр только записей выданных книг, которых читатель еще не вернул
 - Прием сдаваемых книг (обработка списка выдаваемых книг)
 - Формирование бланка Размещения переплетов, для быстрой выдачи литературы из хранилища
 - Обрабатывать заказы читателя
 - Заполнение, редактирование или удаление из списка изданий
 - Заполнение, редактирование или удаление из списка переплетов
 - Заполнение, редактирование или удаление из списка размещения переплетов

3.4.3. Последовательность проведения и режимы испытаний

В ходе испытания будут проверяться функциональные возможности системы в следующей последовательности:

- Защищенная паролем авторизация пользователей при запуске программы;
- Для роли пользователя «администратор» главное окно должно позволять:

- Вводить, редактировать, удалять основные данные о новых библиотекарях, а также устанавливать им пароли;
- Для роли пользователя «читатель» главное окно должно позволять:
 - Автоматизированный поиск изданий по определенным критериям
 - Из списка отобранных изданий, занесение нужной литературы в список заказа
- Для роли пользователя «библиотекарь» главное окно должно позволять:
 - Выбор читателя по его абонементскому номеру, просмотр его анкеты, списка ранее выдаваемых ему книг и списка заказанных им книг
 - Прием сдаваемых книг (обработка списка выдаваемых книг)
 - Формирование бланка Размещения переплетов, для быстрой выдачи литературы из хранилища
 - Обращивать заказы читателя

3.4.4 Требования к программе

При проведении испытаний функциональные характеристики программы подлежат проверке на соответствие требованиям.

3.4.5 Требования к программной документации

Состав документации - это пояснительная записка к дипломному проекту, которая состоит из разделов, соответствующим следующим программным документам:

- Техническое задание;
- Программу и методику испытаний;
- Руководство программиста;
- Руководство пользователя;

3.5 Методика проведения проверки комплектности программной документации

Проверка комплектности программной документации на программное изделие проводилась мною визуально.

Моя пояснительная записка состоит из шести разделов:

- 1) Анализ исходных данных
- 2) Специальная часть. Выбор средств/методологии проектирования
- 3) Экспериментальная часть
- 4) Технологическая часть
- 5) Экономическая часть
- 6) Охрана труда и окружающей среды

И технического задания.

В техническом задании были четко сформулированы все основные требования, предъявляемые к разработчикам, для создания системы:

- Был составлен состав функциональных, выполняемых функций;
- Предъявлены требования к:
 - надежности программы
 - условиям эксплуатации
 - средствам программирования
 - составу и параметрам технических средств
 - информационной и программной совместимости
 - программной документации

В 1-ом разделе «Анализ исходных данных» - было проведено:

- детальное описание предметной области;

- были выделены все атрибуты каталожной карточки, которые необходимо структурировать и хранить в базе данных;
 - была составлена структурная схема работы библиотеки
- Во 2-ом разделе «Специальная часть. Выбор средств/методологии проектирования» - была проведена:
- Сравнительная характеристика существующих СУБД, были рассмотрены основные их преимущества и недостатки;
 - Проанализировав достоинства и недостатки, а также с учетом предъявляемых требований был выбран СУБД MS Access;
 - Сравнительная характеристика существующих языков программирования;
 - Проанализировав их достоинства и недостатки была выбрана среда Borland Delphi 2006 и язык программирования Object Pascal для реализации системы;
 - Построена инфологическая (концептуальная) модель предметной области, в результате которой были выделены основные и вспомогательные объекты;
 - Создана ER-диаграмма «Библиотеки» - которая показывает как созданные объекты взаимодействуют между собой;
 - Спроектирована физическая структура базы данных: была создана каждая таблица (в которую трансформировались объекты), с перечнем всех его полей, типов данных этих полей, их размерностью, ключевыми полями и внешними ключами. Предназначенными для связи таблиц между собой;
 - Создана датологическая модель базы данных – которая показывает все взаимосвязи между таблицами базы данных;
 - Проведена укрупненная алгоритмизация приложения, с выделением основных подпрограмм системы и описанием их функций;

- Проведено программирование, создание приложения, реализующее все основные функциональные подпрограммы;

В 3-ем разделе «Экспериментальная часть» - была проведена:

- Проверка комплектности всей программной документации;
- Проверка функциональных возможностей системы, заложенных в техническом задании.

В 4-ом разделе «Технологическая часть» - было проведено:

- Описано комплексное предназначение системы;
- Описаны особенности выполнения всех основных функций программы.

В 5-ом разделе «Экономическая часть» - было проведено определение эффективности создания и внедрения проектируемой автоматизированной системы библиотекаря на основе сравнения с базовым вариантом. Была рассчитана годовая экономия (45576 руб./год), годовой экономический эффект (23860,15 руб), определена эффективность внедрения программного продукта (22%), определен срок окупаемости капиталовложений (2,36 года). Был сделан вывод, что разработка и использование программного продукта является экономически оправданным и целесообразным.

В 6-ом разделе «Охрана труда и окружающей среды» - было проанализировано:

- Обеспечение условий труда на рабочем месте библиотекаря;
- Анализ состояния охраны труда в помещении;
- Анализ вентиляции, искусственного и природного освещения;
- Были проведены расчеты:
 - достаточности вентиляции в помещении;
 - освещенности помещения;
 - естественного освещения;
 - искусственного освещения;
- Были составлены мероприятия по обеспечению благоприятных условий труда;

- Определен микроклимат рабочей зоны

И сделан вывод, что в помещении библиотеки выполнены все требования к микроклимату, пожаробезопасности и электробезопасности в соответствии с нормативными документами, соблюдаются режимы труда и отдыха. Что все дисплейные терминалы и другое оборудование, применяемое в данном помещении, имеет гигиенический сертификат, соответствует требованиям действующих стандартов, правилам национальных стандартов и имеет в паспорте соответствующую отметку об этом.

В итоге можно сказать, что в моей программной документации (пояснительной записке) были описаны, реализованы и проверены в полном объеме все ее разделы, а так же правильно составлено техническое задание на разработку системы, широко описана и проанализирована предметная область. Были выбраны наиболее рационально подходящие средства создания программных продуктов, в которых по новейшим методологиям разработки систем была реализована автоматизированная система, которая выполняет все предъявляемые к ней функциональные требования.

3.6 Методика проверки выполнения основных функций системы

В техническом задании были представлены основные требования, предъявляемые к разрабатываемой системе. В ходе описания и анализа предметной области эти требования трансформировались в основные функциональные возможности системы. В процессе проектирования были детально расписаны все составляющие базы данных и программы, из которых состоит проектируемая системы. В ходе программирования были реализованы все эти составляющие, но так как программировал человек, а не машина, а человеку свойственно ошибаться, значит после создания системы необходимо провести комплексную проверку правильности выполнения основных функций системы.

3.6.1 Проверка правильности выполнения функции защищенной паролем авторизации пользователей при запуске программы

В окне авторизации выбрал роль «Администратор», второе поле автоматически стало недоступным, а в третье поле ввел неправильный пароль администратора «1651651» и нажал на кнопке «Далее» - на что система выдала сообщение об ошибке:

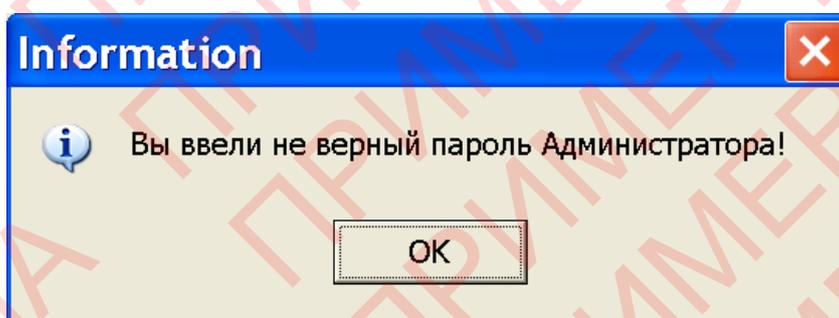


Рисунок 3.1 – Сообщение об ошибке

Т.е. человек не имеющий доступа и не знающий паролей не сможет войти в систему и внести какие то изменения в ней.

Затем в третье поле ввел правильный пароль администратора «111» и нажал на кнопке «Далее». В результате открылось главное окно администратора.

Вывод – функция работает правильно.

3.6.2 Проверка правильности выполнения функции, для роли пользователя «администратор» в главном окне, по вводу, редактированию, удалению основных данных о новых библиотекарях, а также установлению им паролей

В роли администратора вошел в систему, выполнил команду добавления новой записи и ввел основные данные о новом библиотекаре,

вставил его фотографию, а также установил ему пароль для авторизации при входе в программу. Нажал на кнопке «Принять» и убедился что данные появились в базе данных.

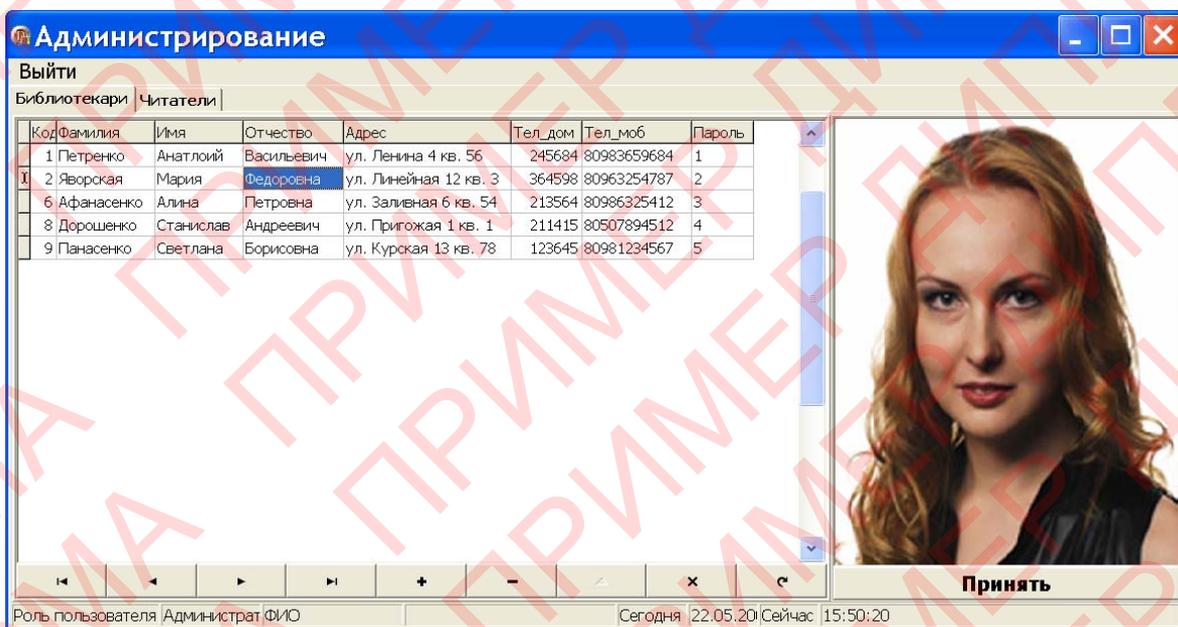


Рисунок 3.2 – Добавление нового библиотекаря в окне Администрирования, на вкладке библиотекаря

Вывод – функция работает правильно.

3.6.3 Проверка правильности выполнения функции, для роли пользователя «читатель» в главном окне, по Автоматизированному поиску изданий по определенным критериям

В главном окне на третьей вкладке «Выбрать издания» нажал на кнопке «Найти нужное издание» и в специальном открывшемся окне выбрал критерии поиска «заглавие - БЫВАЮТ РОКОВЫЕ ДНИ...», «вид издания - газета» и нажал на кнопке «Найти издания»:

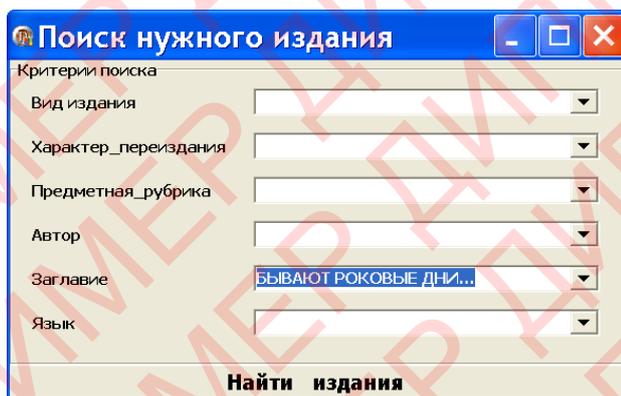


Рисунок 3.3 – Окно поиска изданий

В результате - ничего не было найдено и было выдано соответствующее сообщение с предложением изменить критерии и провести поиск снова:

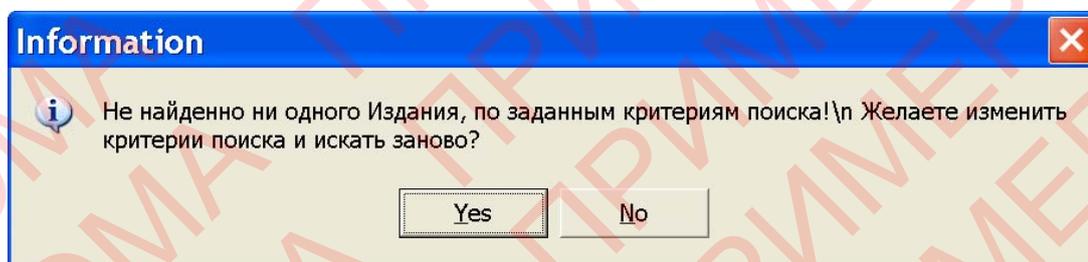


Рисунок 3.4 – Сообщение о том что ничего не найдено

Затем изменил критерии поиска - «заглавие - ВО ГЛУБИНЕ СИБИРСКИХ РУД...» и нажал на кнопке «Найти издания» - в итоге результат поиска отобразился в окне главного окна на третьей вкладке:

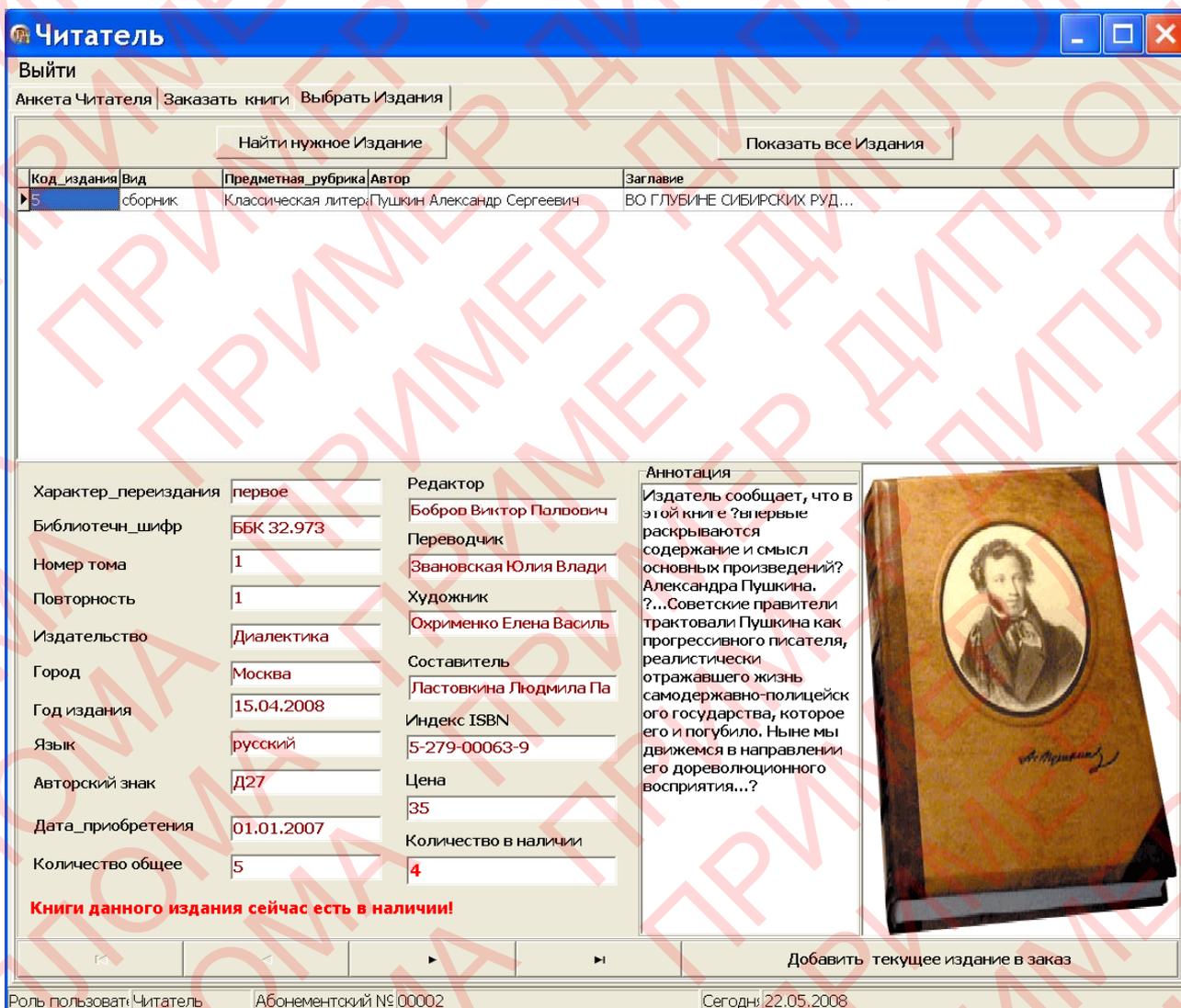


Рисунок 3.5 – Результат поиска издания

Вывод – функция работает правильно.

3.6.6 Проверка правильности выполнения функции, для роли пользователя «читатель» в главном окне, по занесению нужной литературы в список заказа из списка отображенных изданий

По результатам поиска (проверки предыдущей функции), в главном окне на третьей вкладке «Выбрать издания» в верхней таблице выбрал найденное издание и нажал на кнопке «добавить текущее издание в заказ».

Затем перешел на вторую вкладку и увидел, что выбранное издание появилось в нижней таблице «Заказанные книги»:

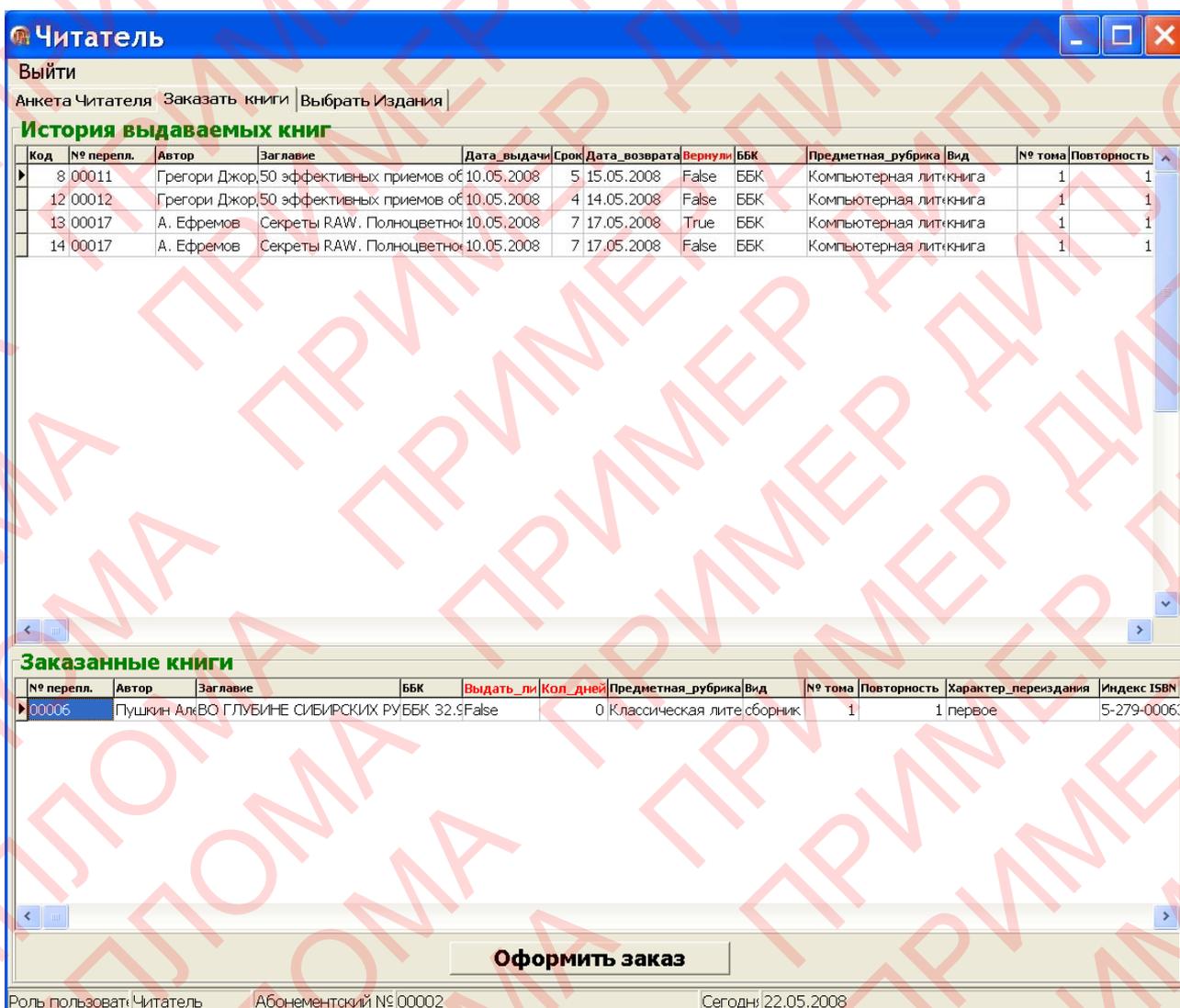


Рисунок 3.6 – Выбор издания в заказ литературы

Вывод – функция работает правильно.

3.6.7 Проверка правильности выполнения функции, для роли пользователя «библиотекарь» в главном окне на второй вкладке, выбора читателя по его абонементскому номеру и просмотру анкеты, списка ранее выдаваемых ему книг и списка заказанных им книг

Вошел в систему в роли библиотекаря, перешел на вторую вкладку, в раскрывавшемся списке читателей выбрал читателя с абонементским номером «00002» и увидел внизу: анкету пользователя с его фотографией, список книг которые он брал и список книг, которые он заказал.



Рисунок 3.7 – Выбор читателя и просмотр его данных и списка заказанных им книг

И убедился, что заказ, который сформировал в предыдущем пункте в роли читателя, правильно отображается для библиотекаря.

Вывод – функция работает правильно.

3.6.8 Проверка правильности выполнения функции, для роли пользователя «библиотекарь» в главном окне на второй вкладке, приема сдаваемых книг (обработки списка выдаваемых книг)

В главном окне на второй вкладке для роли библиотекаря, при выбранном читателе с абонементским номером «00002» имитировал процесс приема книг библиотекарем (возврата книг читателем). В таблице «История выдаваемых книг» выбрал литературу с кодом «8», выделил поле «вернули», нажал над ним правой кнопкой мыши и выбрал из контекстного меню «true»:

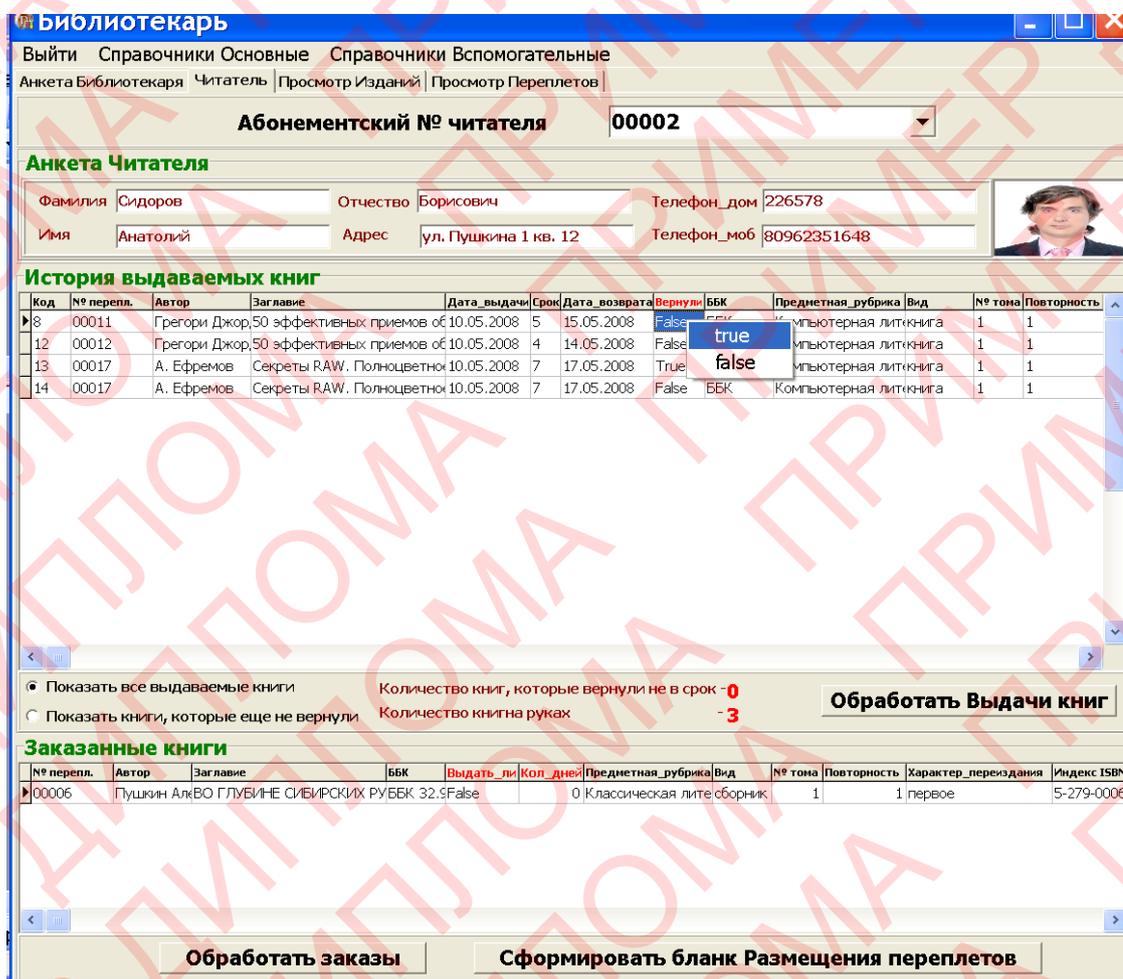


Рисунок 3.8 – Отмечание возврата книжки

В результате поле приняло значение «true». Затем нажал на кнопке «Обработать Выдачи книг» и увидел сообщение, что «выдачи успешно обработаны»:

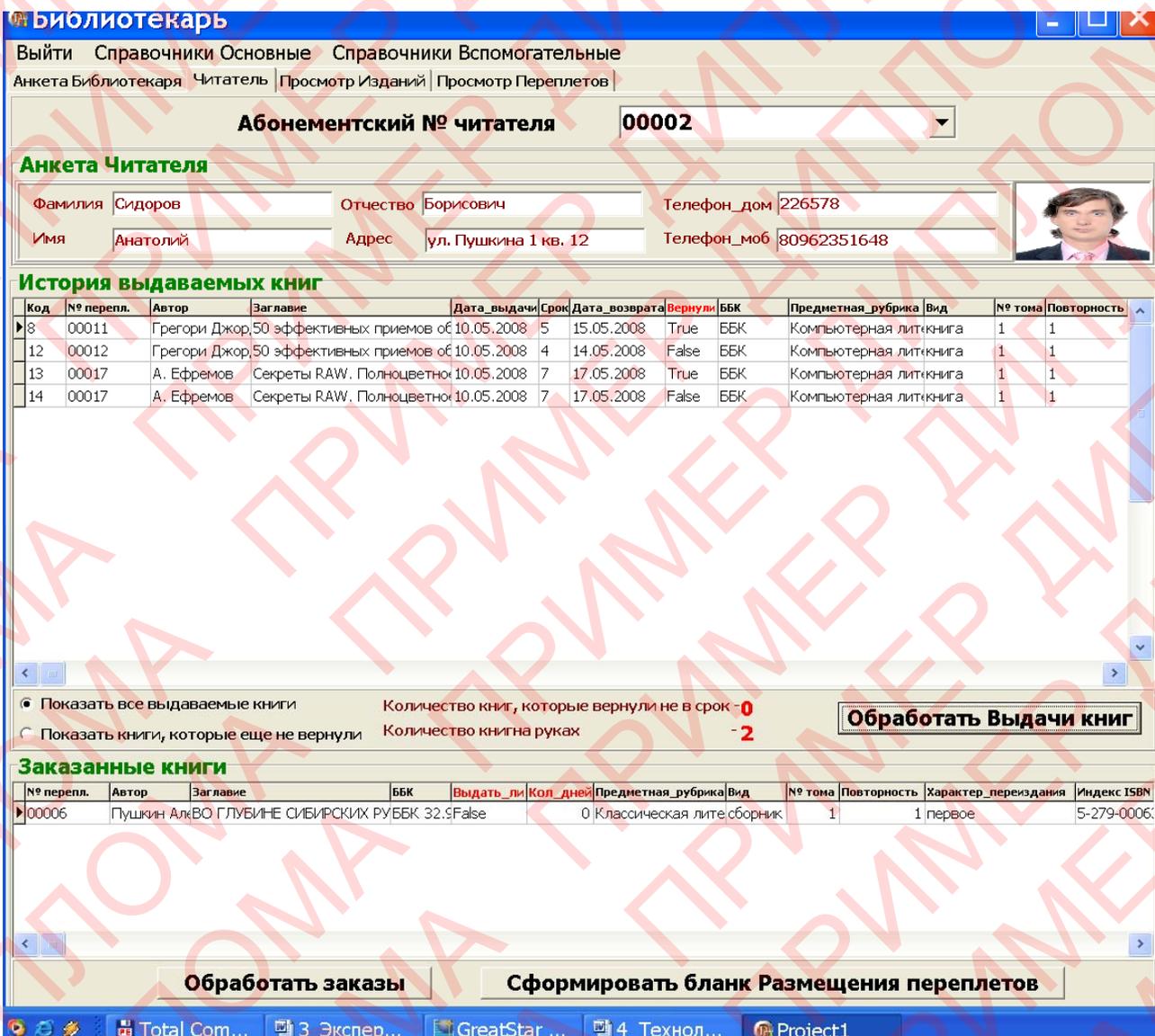


Рисунок 3.9 – Успешно обработанная выдача литературы

Затем перешел на четвертую вкладку «Просмотр переплетов» и убедился что значение поля «наличие», для переплета, который был выдан, а только что обработан как вернули – стало «true».

Вывод – функция работает правильно.

№переплета	Автор	Заглавие	№тома	ББК	Предметная_рубрика	Индекс ISBN	Наличие
00001	Пушкин Александр	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.9Классическая литер	5-279-00063	False	
00002	Пушкин Александр	СКАЗКИ	1	ББК 32.9Классическая литер	5-278-00043	False	
00003	Пушкин Александр	СКАЗКИ	1	ББК 32.9Классическая литер	5-278-00043	True	
00004	Пушкин Александр	СКАЗКИ	1	ББК 32.9Классическая литер	5-278-00043	True	
00005	Пушкин Александр	СКАЗКИ	1	ББК 32.9Классическая литер	5-278-00043	True	
00006	Пушкин Александр	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.9Классическая литер	5-279-00063	True	
00007	Пушкин Александр	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.9Классическая литер	5-279-00063	True	
00008	Пушкин Александр	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.9Классическая литер	5-279-00063	True	
00009	Пушкин Александр	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.9Классическая литер	5-279-00063	True	
00010	Скотт Келби	Adobe Photoshop CS3: справочн	1	ББК	Компьютерная лите 978-5-8459- True		
00011	Грегори Джордже	50 эффективных приемов обра	1	ББК	Компьютерная лите 5-8459-0939 False		
00012	Грегори Джордже	50 эффективных приемов обра	1	ББК	Компьютерная лите 5-8459-0939 False		
00013	Грегори Джордже	50 эффективных приемов обра	1	ББК	Компьютерная лите 5-8459-0939 True		
00014	Скотт Келби	Adobe Photoshop CS3: справочн	1	ББК	Компьютерная лите 978-5-8459- True		
00015	В. Мураховский,	С Азбука цифрового фото. Насто	1	ББК	Компьютерная лите 978-5-91180 True		
00016	В. Мураховский,	С Азбука цифрового фото. Насто	1	ББК	Компьютерная лите 978-5-91180 True		
00017	А. Ефремов	Секреты RAW. Полноцветное и	1	ББК	Компьютерная лите 978-5-91180 False		
00018	А. Ефремов	Секреты RAW. Полноцветное и	1	ББК	Компьютерная лите 978-5-91180 True		
00019	Дэвид Буш	Цифровые зеркальные фотока	1	ББК	Компьютерная лите 978-5-8459- True		
00020	Дэвид Буш	Цифровые зеркальные фотока	1	ББК	Компьютерная лите 978-5-8459- True		
00021	А. Жвалеvский	Цифровое фото и Photoshop бе	1	ББК	Компьютерная лите 978-5-388-0 True		
00022	А. Жвалеvский	Цифровое фото и Photoshop бе	1	ББК	Компьютерная лите 978-5-388-0 True		
00023	А. Ефремов	Секреты RAW. Полноцветное и	1	ББК	Компьютерная лите 978-5-91180 True		

Рисунок 3.10 – Отображение признаков наличия литературы

3.6.9 Проверка правильности выполнения функции, для роли пользователя «библиотекарь» в главном окне на второй вкладке, формирования бланка Размещения переплетов, для быстрой выдачи литературы из хранилища

В этом же окне, в роли библиотекаря для того же выбранного читателя, что и в пункте выше, сформировал для его заказа литературы бланк размещения переплетов, нажав на кнопке с соответствующим названием. В результате получил в MS Excel следующую таблицу:

1	A	B	C	D	E	F
2	Номер переплета	Автор	Бланк Размещения выдаваемых переплетов			
3		8 Пушкин Александр Сергеевич	Заглавие ВО ГЛУБИНЕ СИБИРСКИХ РУД...	Номер комнаты 1	Номер стеллажа 1	Номер полки 2
4						
5						
6						
7						
8						
9						

Рисунок 3.11 – Сформированный бланк размещения переплетов
 Вывод – функция работает правильно.

3.6.10 Проверка правильности выполнения функции, для роли пользователя «библиотекарь» в главном окне на второй вкладке, обработки заказа читателя

В этом же окне, в роли библиотекаря для того же выбранного читателя, что и в пункте выше, в таблице «Заказанные книги» в поле «Кол_дней» на которое можно выдать книгу читателю ввел «5» (если выдавать книгу этому читателю нельзя – необходимо ввести в это поле ноль) и нажал на кнопке «Обработать заказы». В результате заказанная книга исчезла в таблице заказов и появилась в таблице выданных книг, со сроком 5 дней, проставленной конечной датой сдачи книги и признаком, что книгу еще не вернули:

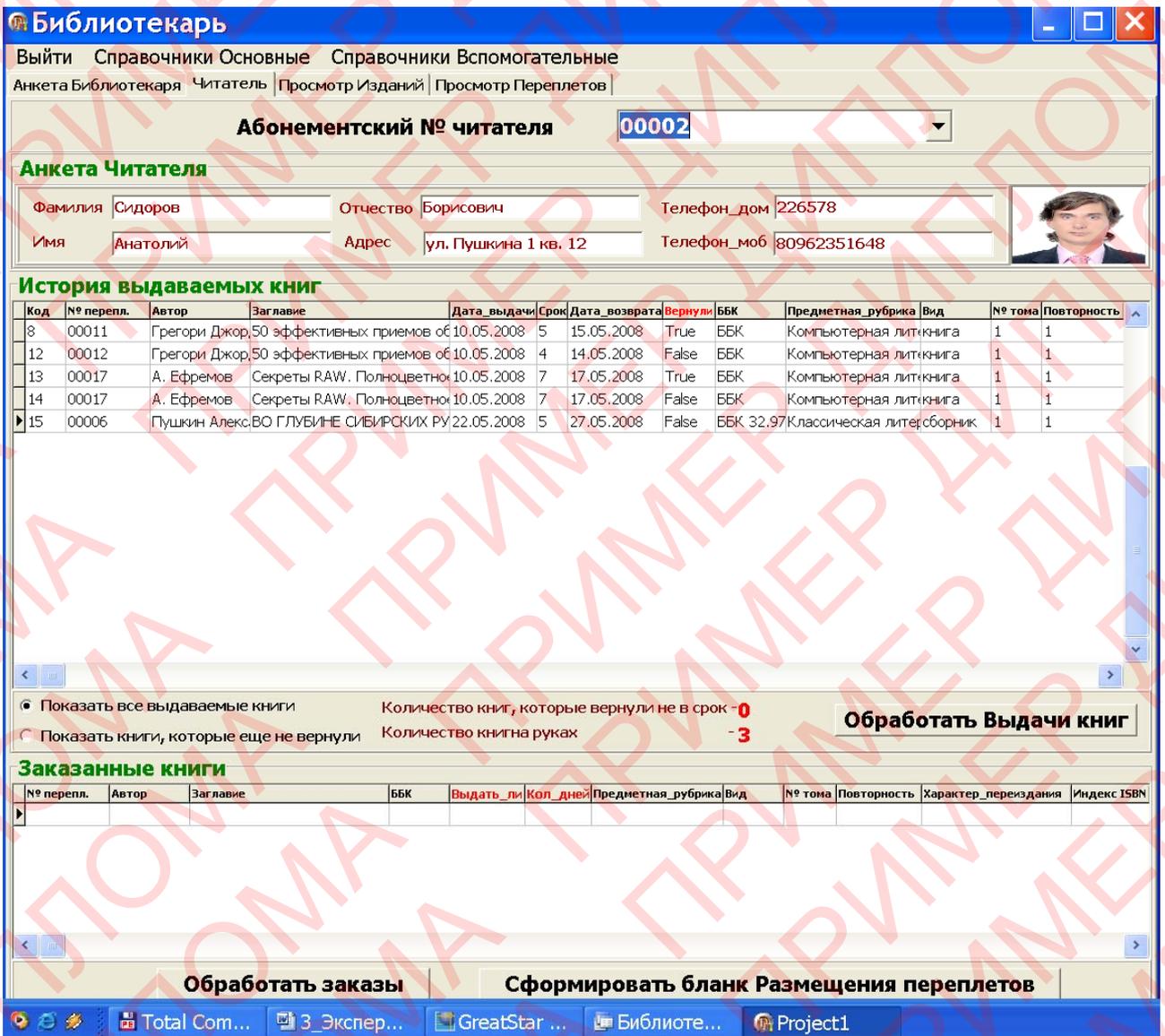


Рисунок 3.12 – Обработанный заказ

Вывод – функция работает правильно.

4. Технологическая часть

4.1. Назначение программы

4.1.1. Функциональное назначение

Функциональным назначением Автоматизированной системы является ввод, хранение, изменения, структурирование, удаление информации, выполнение автоматизированной обработки новых поступлений в библиотеку, поиск и анализ библиотечной информации.

4.1.2. Эксплуатационное назначение

Автоматизированная система должна эксплуатироваться в библиотеках для освобождения сотрудников от ряда рутинных работ по подготовке картотек, изданий, списков, а так же для получения читателем рекомендаций по выбору книг.

Конечными пользователями программы должны являться сотрудники библиотеки.

4.1.3. Состав функций

Программа обеспечивает возможность выполнения перечисленных ниже функций:

- Защищенная паролем авторизация пользователей при запуске программы;

- В зависимости от выбора одной из трех ролей пользователей: администратор, библиотекарь, читатель – дальше открывать разные, соответствующие выбранной роли, главные окна программы.
- Корректный выход из программы, с предупреждающим сообщением.
- Для роли пользователя «администратор» главное окно должно позволять:
 - Вводить, редактировать, удалять основные данные о новых библиотекарях, а также устанавливать им пароли;
 - Вводить, редактировать, удалять основные данные о новых читателях, а также устанавливать им пароли;
- Для роли пользователя «читатель» главное окно должно позволять:
 - Отображать анкету читателя
 - Просмотр книг, которые брал ранее
 - Выбор необходимых изданий, с детальным просмотром всех его данных
 - Автоматизированный поиск изданий по определенным критериям
 - Из списка отобранных изданий, занесение нужной литературы в список заказа
 - Оформить заказ отобранной литературы
- Для роли пользователя «библиотекарь» главное окно должно позволять:
 - Отображать анкету библиотекаря
 - Просмотр всех данных обо всех изданиях
 - Просмотр всех данных обо всех переплетах
 - Выбор читателя по его абонементскому номеру, просмотр его анкеты, списка ранее выдаваемых ему книг и списка заказанных им книг

- Просмотр только записей выданных книг, которых читатель еще не вернул
- Прием сдаваемых книг (обработка списка выдаваемых книг)
- Формирование бланка Размещения переплетов, для быстрой выдачи литературы из хранилища
- Обрабатывать заказы читателя
- Заполнение, редактирование или удаление из списка изданий
- Заполнение, редактирование или удаление из списка переплетов
- Заполнение, редактирование или удаление из списка размещения переплетов
- Заполнение, редактирование или удаление из данных из вспомогательных справочников

4.2. Условия выполнения программы

4.2.1. Минимальный состав технических средств

Для функционирования программы необходим компьютер типа IBM PC со следующей минимальной конфигурацией:

- Процессор Intel 1ГГц;
- Память: 512 Mb;
- Жесткий диск емкостью не менее 20 Гб;
- Дисплей SVGA с соответствующим видеоадаптером;
- Минимальное разрешение экрана 800x600 (рекомендуется 1024x768)
- Клавиатура, указательное устройство “мышь”.

4.2.2. Минимальный состав программных средств

Для работы exe-модуля программы использовать следующие операционные системы: Windows 2000 Professional / Windows XP Professional.

4.3. Требования к персоналу (пользователю)

Минимальное количество пользователей программы, должно составлять не менее 3 человек: администратор и конечные пользователи программы – библиотекарь и читатель.

В перечень задач, выполняемых администратором, должны входить:

- задача поддержания работоспособности технических средств;
- задачи установки (инсталляции) и поддержания работоспособности системных программных средств - операционной системы;
- задача установки (инсталляции) программы;
- задача выдачи пароле пользователям системы (библиотекарям и читателям).

Конечный пользователь программы (библиотекарь или читатель) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

4.4. Выполнение программы

4.4.1 Выполнение (загрузки и запуска) программы

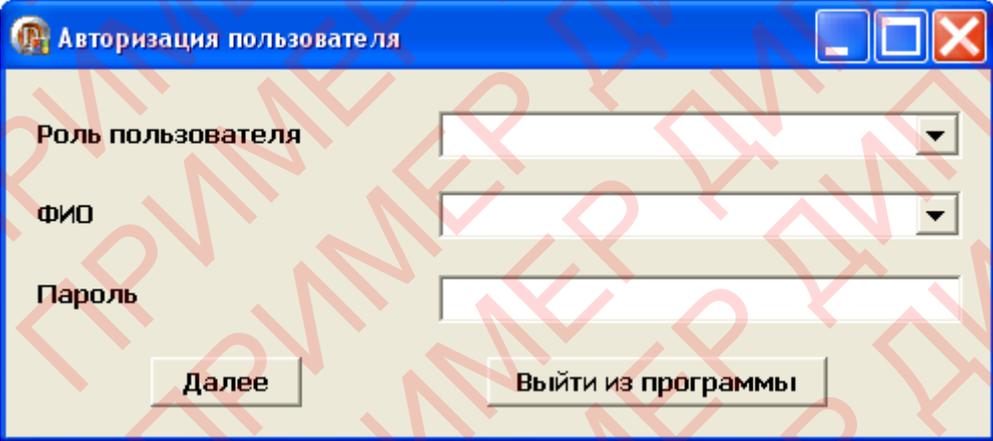
Загрузка и запуск программы осуществляется способами, детальные сведения о которых изложены в Руководстве пользователя операционной системы.

После запуска приложения, файла «СА Библиотека.exe», откроется окно-заставка САБ:



Рисунок 4.1 – Окно заставки САБ

Затем, через несколько секунд, откроется окно для авторизации пользователей:



Авторизация пользователя

Роль пользователя

ФИО

Пароль

Далее

Выйти из программы

Рисунок 4.2 – Окно для авторизации пользователей

4.4.2. Выполнение функций программы

4.4.2.1 Выполнение функции открытия разных главных окон программы, в зависимости от выбранной роли пользователя

Для выполнения указанной функции в окне авторизации пользователь должен вначале выбрать «Роль пользователя»:

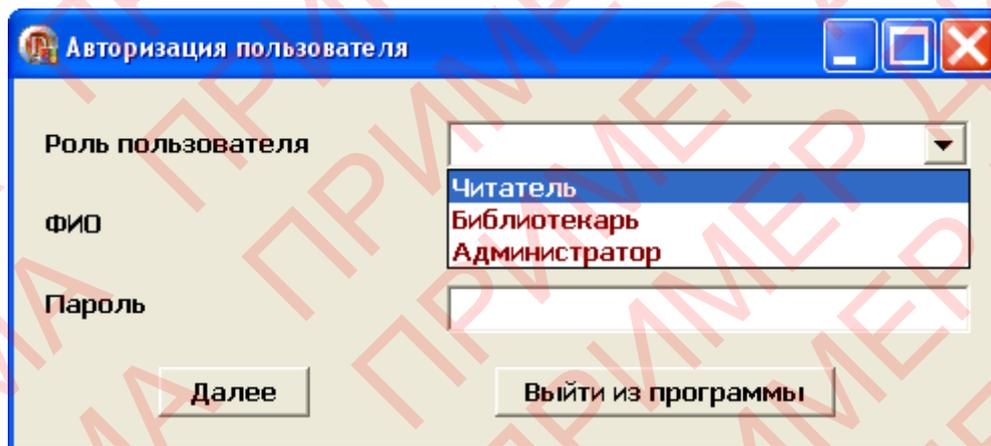


Рисунок 4.3 – Выбор роли пользователя

В САБ по умолчанию задано 3 типа пользователей:

1. Читатель – пользователь, который пришел взять или сдать в библиотеку литературу.
2. Библиотекарь – пользователь, который работает в библиотеке, который выдает читателям и принимает назад литературу.
3. Администратор - пользователь, который администрирует САБ, вводит основные данные о новых читателях и библиотекарях в САБ, а также редактирует или удаляет их.

4.4.2.2 Выполнение функции корректный выход из программы, с предупреждающим сообщением

Для выполнения указанной функции при нажатии на кнопке «Выйти из программы» - выдаться предупреждающее сообщение:

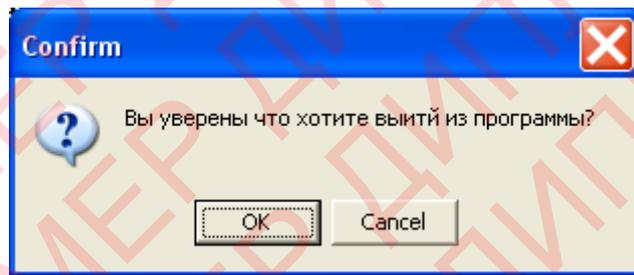


Рисунок 4.4 – Предупреждающей сообщении

Если на него ответить «Ок» - программа закрывается, если «Cancel» - программа продолжит работу.

4.4.2.3 Выполнение функции защищенной паролем авторизации пользователей

Для выполнения указанной функции, если пользователь выбрал роль «Администратор», то второе поле становится недоступным, а в третьем необходимо ввести пароль администратора (111).

4.4.2.4 Для роли пользователя «администратора» выполнение функции ввода, редактирования, удаления основных данных о новых библиотекарях, а также установления им пароли

Для выполнения указанной функции при нажатии на кнопку «Далее», в окне авторизации, если правильно введен пароль, открывается окно «Администрирование». В котором администратор вводит основные данные о новых библиотекарях в САБ, редактирует или удаляет их. А также устанавливает им пароли для авторизации при входе в программу.

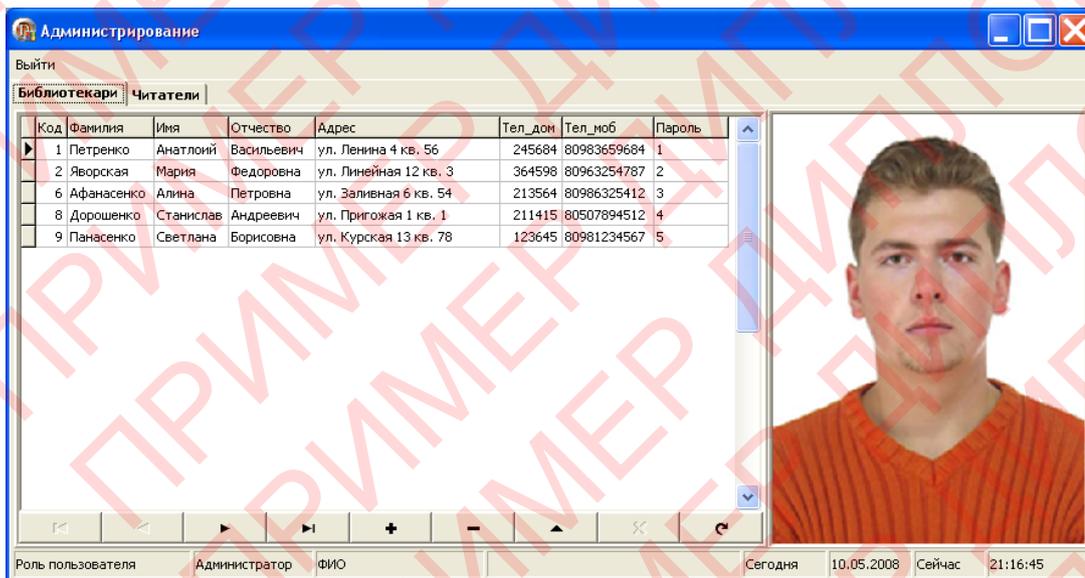


Рисунок 4.5 – Окно Администрирования, вкладка библиотекари

4.4.2.5 Для роли пользователя «администратора» выполнение функции ввода, редактирования, удаления основных данных о новых читателях, а также установления им пароли

Для выполнения указанной функции при нажатии на кнопке «Далее», в окне авторизации, если правильно введен пароль, открывается окно «Администрирование». В котором администратор вводит основные данные о новых читателях в САБ, редактирует или удаляет их. А также устанавливает им пароли для авторизации при входе в программу.

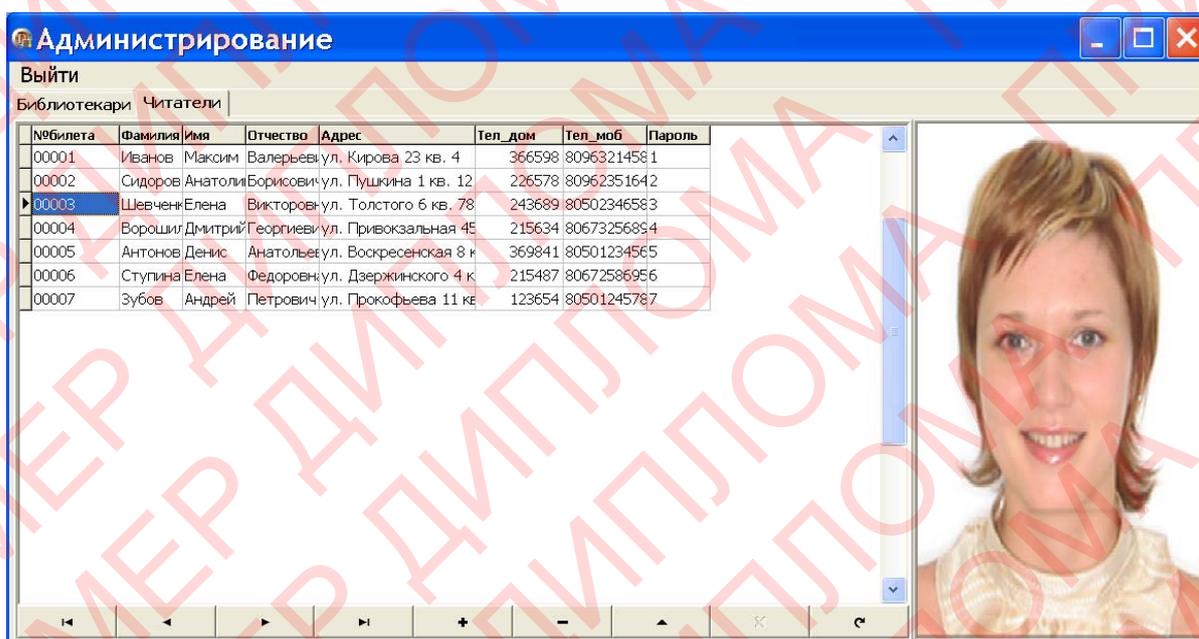


Рисунок 4.6 – Окно Администрирования, вкладка читателя

4.4.2.6 Для роли пользователя «читатель» выполнение функции отображения анкеты читателя

Для выполнения указанной функции, если в окне авторизации пользователь выбрал роль «Читатель», затем он должен выбрать во втором списке № своего абонемента и ввести свой пароль (для 00001 – пароль 1, для 00002 – пароль 2, и т.д.).

При нажатии на кнопке «Далее», если правильно введен пароль, открывается окно «Читатель» с тремя вкладками. На первой – отображается анкета читателя:



The screenshot shows a web application window titled "Читатель". At the top, there are navigation links: "Выйти", "Анкета Читателя", "Заказать книги", and "Выбрать Издания". The main content area contains a form with the following fields:

Абонементский № читателя	00002
Фамилия	Сидоров
Имя	Анатолий
Отчество	Борисович
Адрес	ул. Пушкина 1 кв. 12
Телефон_дом	226578
Телефон_моб	80962351648

To the right of the form is a portrait of a man with dark hair, wearing a suit and tie. At the bottom of the window, there is a status bar with the text: "Роль пользователя Читатель", "Абонементский № читателя 00002", and "Сегодня 10.05.2008".

Рисунок 4.7 – Окно Читатель, первая вкладка

4.4.2.7 Для роли пользователя «читатель» выполнение функции просмотра книг, которые брал ранее

Для выполнения указанной функции на второй вкладке читатель может посмотреть в верхней таблице ведомость литературы, которая выдавалась ему, в которой он может увидеть литературу, которую он еще не сдал назад в библиотеку (см. рис. 4.8).

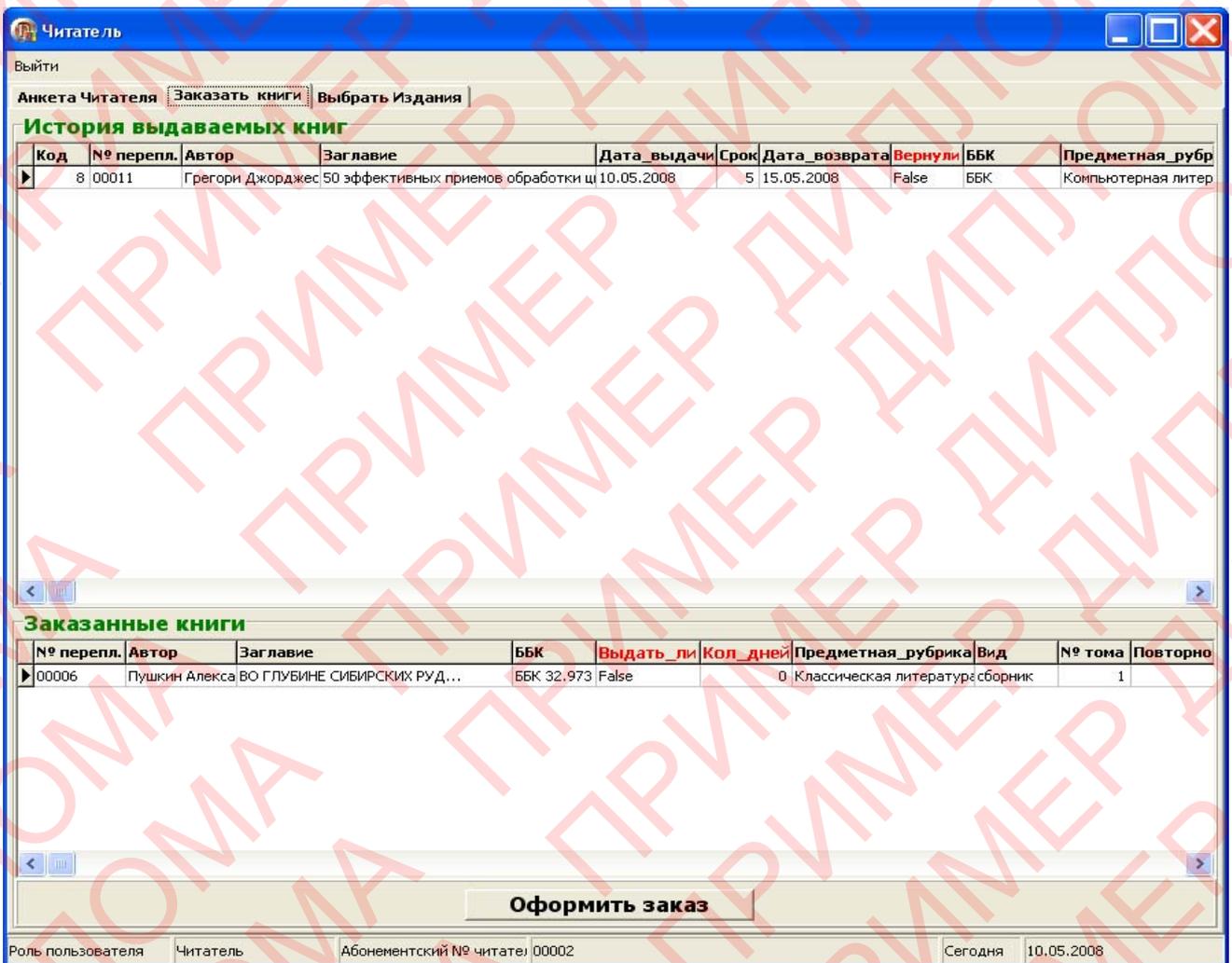


Рисунок 4.8 – Окно Читатель, вторая вкладка

4.4.2.8 Для роли пользователя «читатель» выполнение функции выбора необходимых изданий, с детальным просмотром всех его данных

Для выполнения указанной функции на третьей вкладке читатель может просмотреть полный список всех изданий, которые есть в библиотеке:

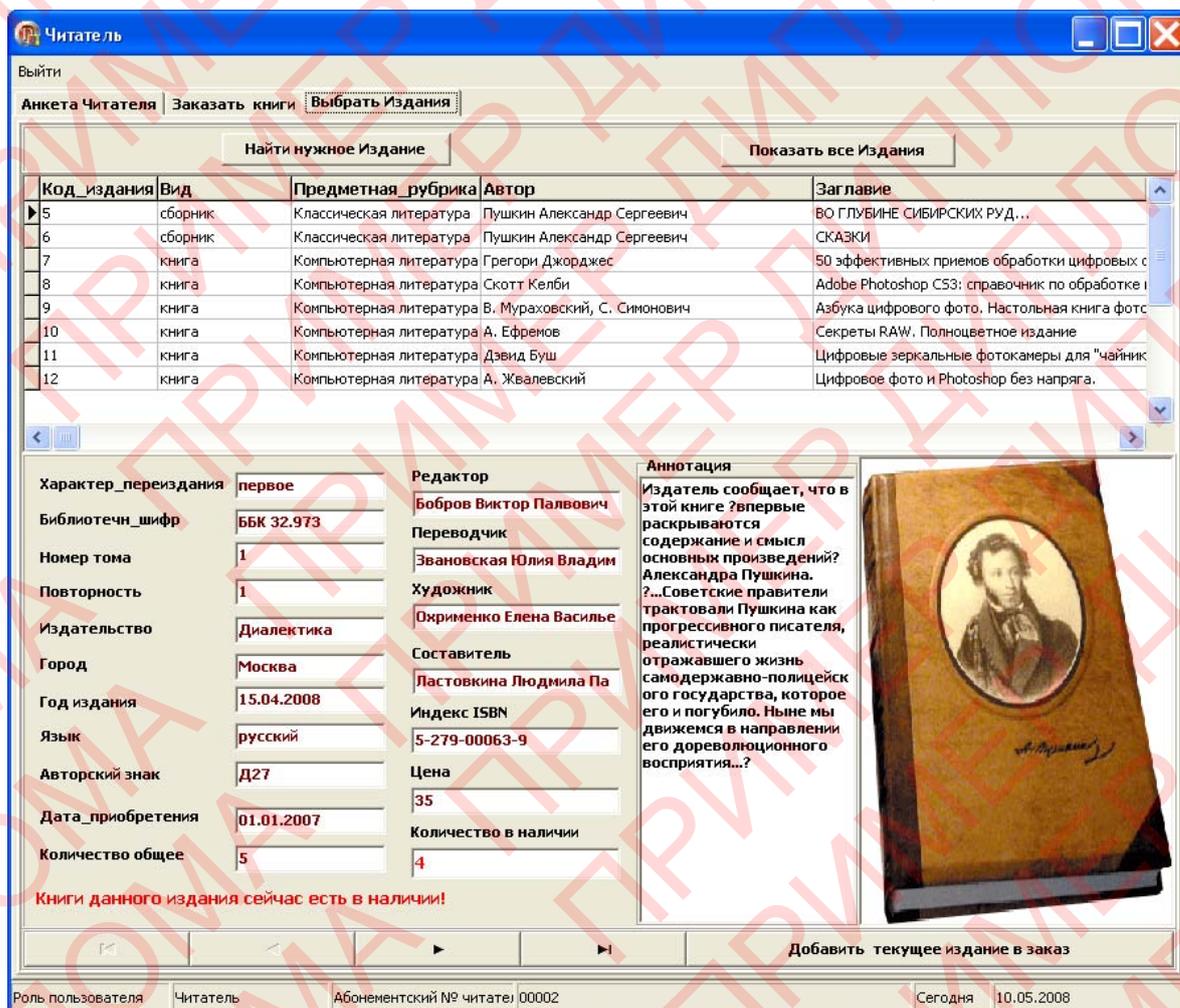


Рисунок 4.9 – Окно Читатель, третья вкладка

4.4.2.9 Для роли пользователя «читатель» выполнение функции автоматизированного поиска изданий по определенным критериям

Для выполнения указанной функции необходимо нажать на кнопке «Найти нужное Издание», воспользовавшись специальной подпрограммой поиска литературы, в специальном открывшемся окне необходимо выбрать нужные критерии поиска и нажать на кнопке «Найти издания»:

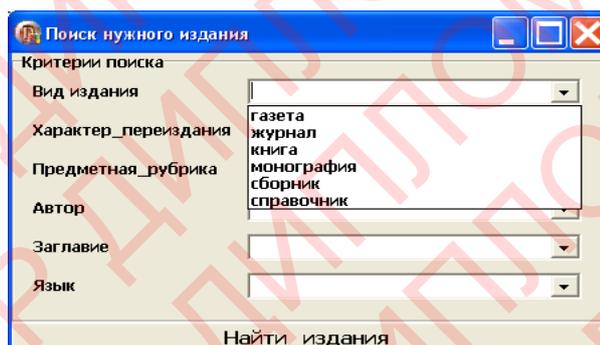


Рисунок 4.10 – Окно поиска изданий

В результате:

- если ничего не найдено – будет выдано соответствующее сообщение с предложением изменить критерии и провести поиск снова;

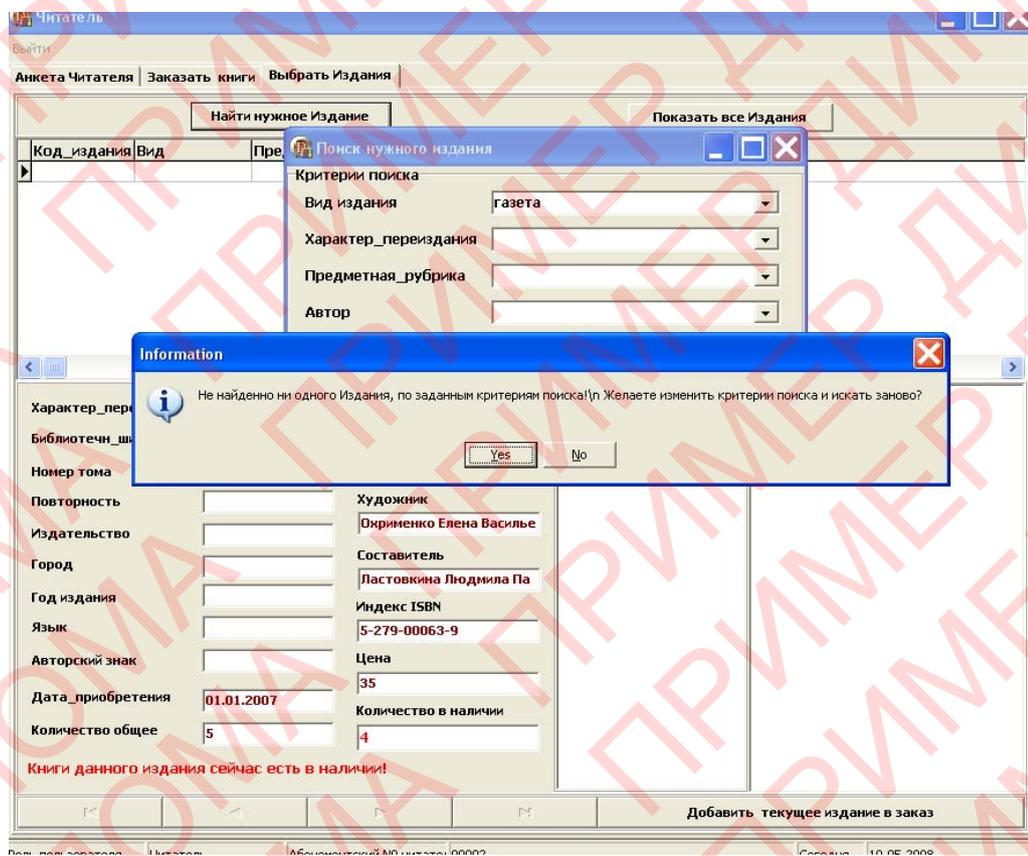


Рисунок 4.11 – Не результативный поиск издания

- если что-то найдено – то результат поиска отобразится в окне (см. рис. 4.12).

4.4.2.10 Для роли пользователя «читатель» выполнение функции занесения нужной литературы в список заказа, из списка отобранных изданий

Для выполнения указанной функции необходимо читателю, среди отобранного, найденного списка, выбрать литературу, которая ему нужна, и занести эту литературу в заказ. Для чего необходимо выбрать по очереди нужное издание и нажать на кнопке «Добавить текущее издание в заказ».

4.4.2.11 Для роли пользователя «читатель» выполнение функции оформление заказа отобранной литературы

Для выполнения указанной функции читателю необходимо перейти на вторую вкладку, в нижней таблице которой будет отображена отобранная для заказа литература (см. рис. 4.13).

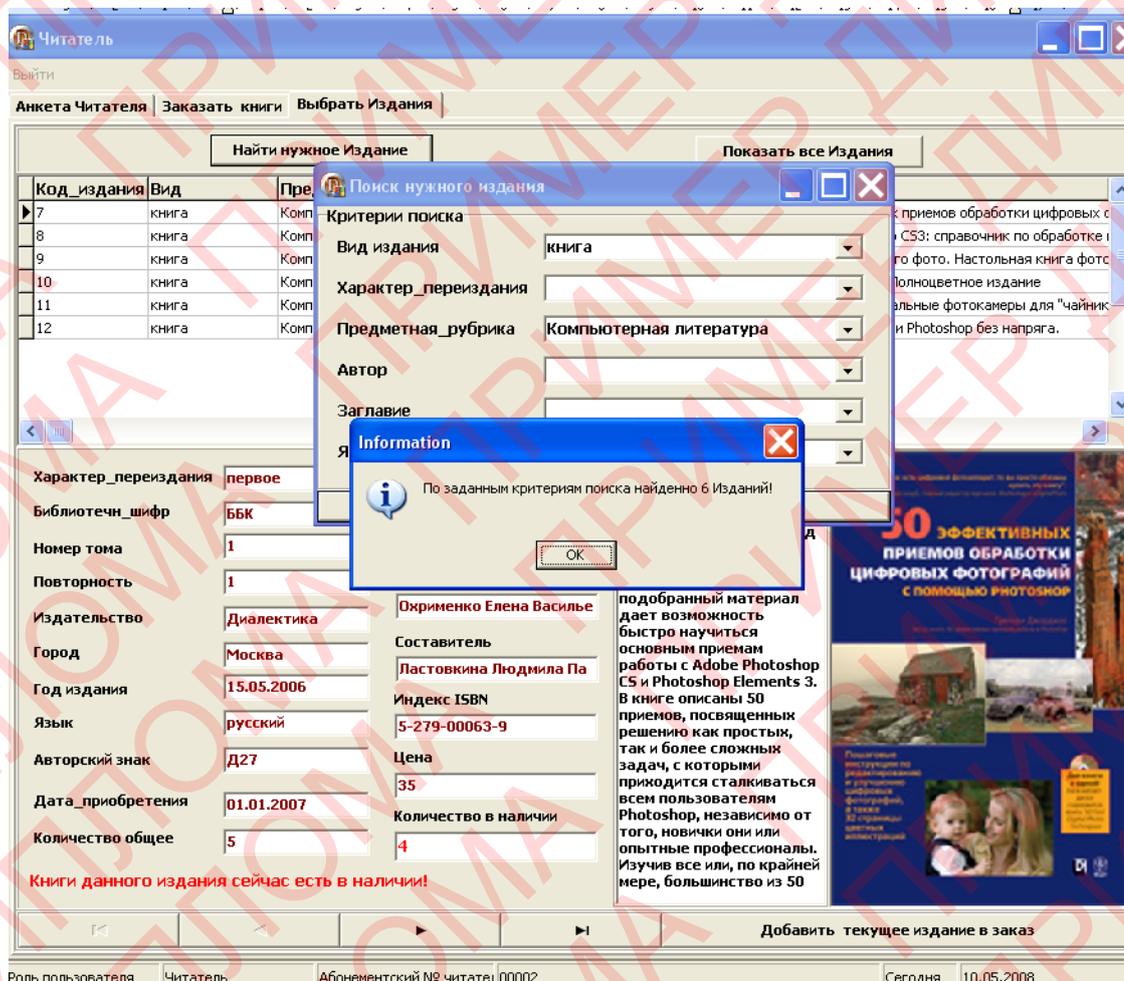


Рисунок 4.12 – Результат поиска

И нажать на кнопке «Оформить заказ».

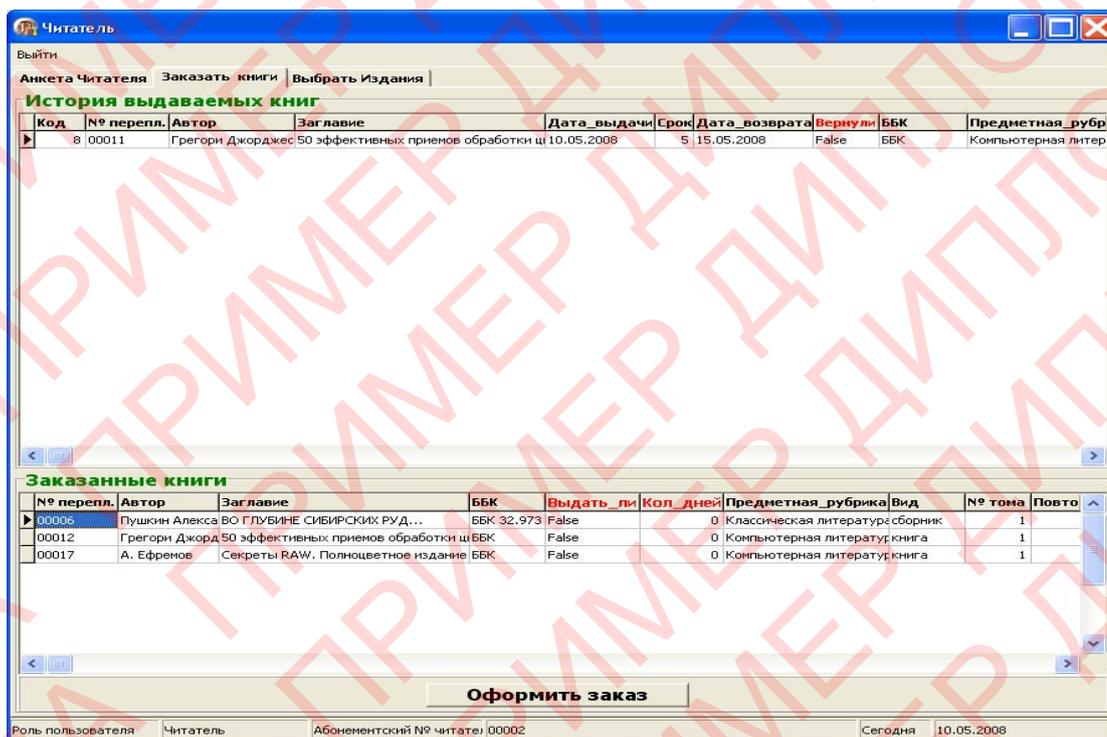


Рисунок 4.13 – Список Заказанных книг

4.4.2.12 Для роли пользователя «библиотекарь» выполнение функции отображения анкеты библиотекаря

Для выполнения указанной функции, если пользователь выбрал роль «Библиотекарь» затем он должен выбрать во втором списке фамилию имя и отчества библиотекаря и ввести соответствующий пароль (для Петренко Анатлоий Васильевич – пароль 1).

При нажатии на кнопке «Далее», если правильно введен пароль, открывается окно «Библиотекарь» с четырьмя вкладками.

На первой – отображается анкета библиотекаря:

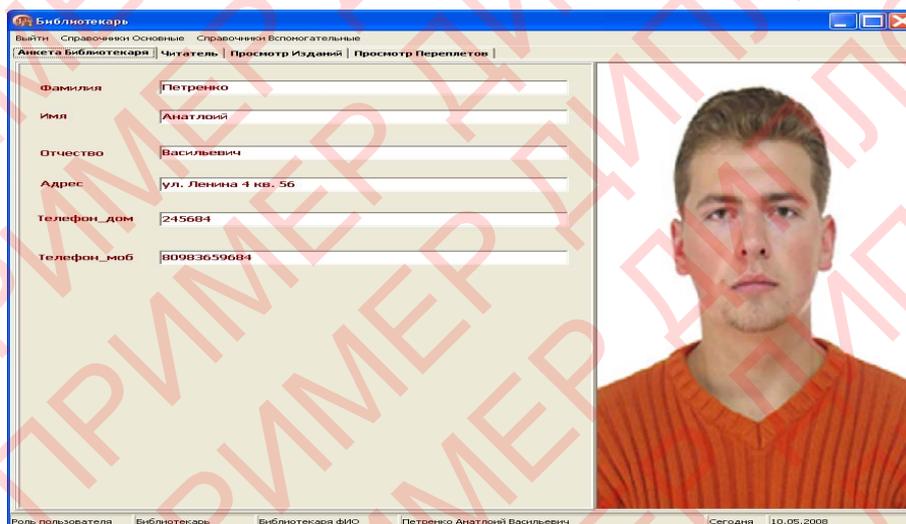


Рисунок 4.14 – Окно Библиотекарь, первая вкладка

4.4.2.13 Для роли пользователя «библиотекарь» выполнение функции просмотра всех данных обо всех изданиях

Для выполнения указанной функции библиотекарю необходимо перейти на третью вкладку, чтобы просмотреть все данные обо всех изданиях, которые есть в библиотеке.

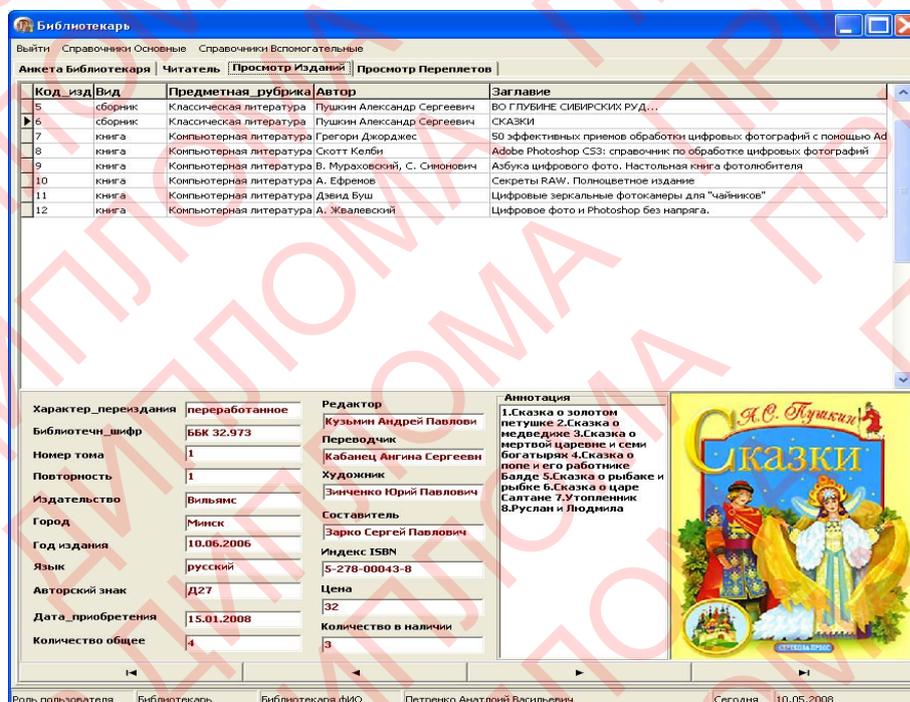


Рисунок 4.15 – Окно Библиотекарь, третья вкладка

4.4.2.14 Для роли пользователя «библиотекарь» выполнение функции просмотра всех данных обо всех переплетах

Для выполнения указанной функции библиотекарю необходимо перейти на четвертую вкладку, где он может просмотреть данные обо всех переплетах, которые есть в библиотеке.

№переплета	Автор	Заглавие	№тома	ББК	Предметная рубрика	Индекс ISBN	Наличие
00001	Пушкин Александр Серге	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.973	Классическая литература	5-279-00063-9	False
00002	Пушкин Александр Серге	СКАЗКИ	1	ББК 32.973	Классическая литература	5-278-00043-8	False
00003	Пушкин Александр Серге	СКАЗКИ	1	ББК 32.973	Классическая литература	5-278-00043-8	True
00004	Пушкин Александр Серге	СКАЗКИ	1	ББК 32.973	Классическая литература	5-278-00043-8	True
00005	Пушкин Александр Серге	СКАЗКИ	1	ББК 32.973	Классическая литература	5-278-00043-8	True
00006	Пушкин Александр Серге	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.973	Классическая литература	5-279-00063-9	True
00007	Пушкин Александр Серге	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.973	Классическая литература	5-279-00063-9	True
00008	Пушкин Александр Серге	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.973	Классическая литература	5-279-00063-9	True
00009	Пушкин Александр Серге	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	ББК 32.973	Классическая литература	5-279-00063-9	True
00010	Скотт Келби	Adobe Photoshop CS3: справочник по обраб	1	ББК	Компьютерная литература	978-5-8459-1364	True
00011	Грегори Джорджес	50 эффективных приемов обработки цифр	1	ББК	Компьютерная литература	5-8459-0939-2	False
00012	Грегори Джорджес	50 эффективных приемов обработки цифр	1	ББК	Компьютерная литература	5-8459-0939-2	True
00013	Грегори Джорджес	50 эффективных приемов обработки цифр	1	ББК	Компьютерная литература	5-8459-0939-2	True
00014	Скотт Келби	Adobe Photoshop CS3: справочник по обраб	1	ББК	Компьютерная литература	978-5-8459-1364	True
00015	В. Мураховский, С. Симо	Азбука цифрового фото. Настольная книг	1	ББК	Компьютерная литература	978-5-91180-951	True
00016	В. Мураховский, С. Симо	Азбука цифрового фото. Настольная книг	1	ББК	Компьютерная литература	978-5-91180-951	True
00017	А. Ефремов	Секреты RAW. Полноцветное издание	1	ББК	Компьютерная литература	978-5-91180-431	True
00018	А. Ефремов	Секреты RAW. Полноцветное издание	1	ББК	Компьютерная литература	978-5-91180-431	True
00019	Дэвид Буш	Цифровые зеркальные фотокамеры для "	1	ББК	Компьютерная литература	978-5-8459-1394	True
00020	Дэвид Буш	Цифровые зеркальные фотокамеры для "	1	ББК	Компьютерная литература	978-5-8459-1394	True
00021	А. Жвалевский	Цифровое фото и Photoshop без напряга.	1	ББК	Компьютерная литература	978-5-388-00122	True
00022	А. Жвалевский	Цифровое фото и Photoshop без напряга.	1	ББК	Компьютерная литература	978-5-388-00122	True
00023	А. Ефремов	Секреты RAW. Полноцветное издание	1	ББК	Компьютерная литература	978-5-91180-431	True

Рисунок 4.16 – Окно Библиотекарь, четвертая вкладка

4.4.2.15 Для роли пользователя «библиотекарь» выполнение функции выбора читателя по его абонементскому номеру, просмотра его анкеты, списка ранее выдаваемых ему книг и списка заказанных им книг

Для выполнения указанной функции библиотекарю необходимо перейти на вторую вкладку (см. рис. 4.17), где он должен вначале выбрать читателя по номеру абонементского билета в верхней части окна. В результате в низу отобразятся:

- анкета читателя;

- историю выдаваемой литературы читателю (в этой таблице библиотекарь может видеть литературу, которая еще на руках у читателя, нажав на соответствующем переключателе под таблицей);
- список заказанной литературы читателем.

4.4.2.16 Для роли пользователя «библиотекарь» выполнение функции просмотра только записей выданных книг, которых читатель еще не вернул

Для выполнения указанной функции библиотекарю необходимо нажать на переключатель «показать книги, которые еще не вернули». В результате в верхней таблице «История выдаваемых книг» отобразятся лишь записи о выдаче литературы, которую читатель еще не вернул в библиотеку.

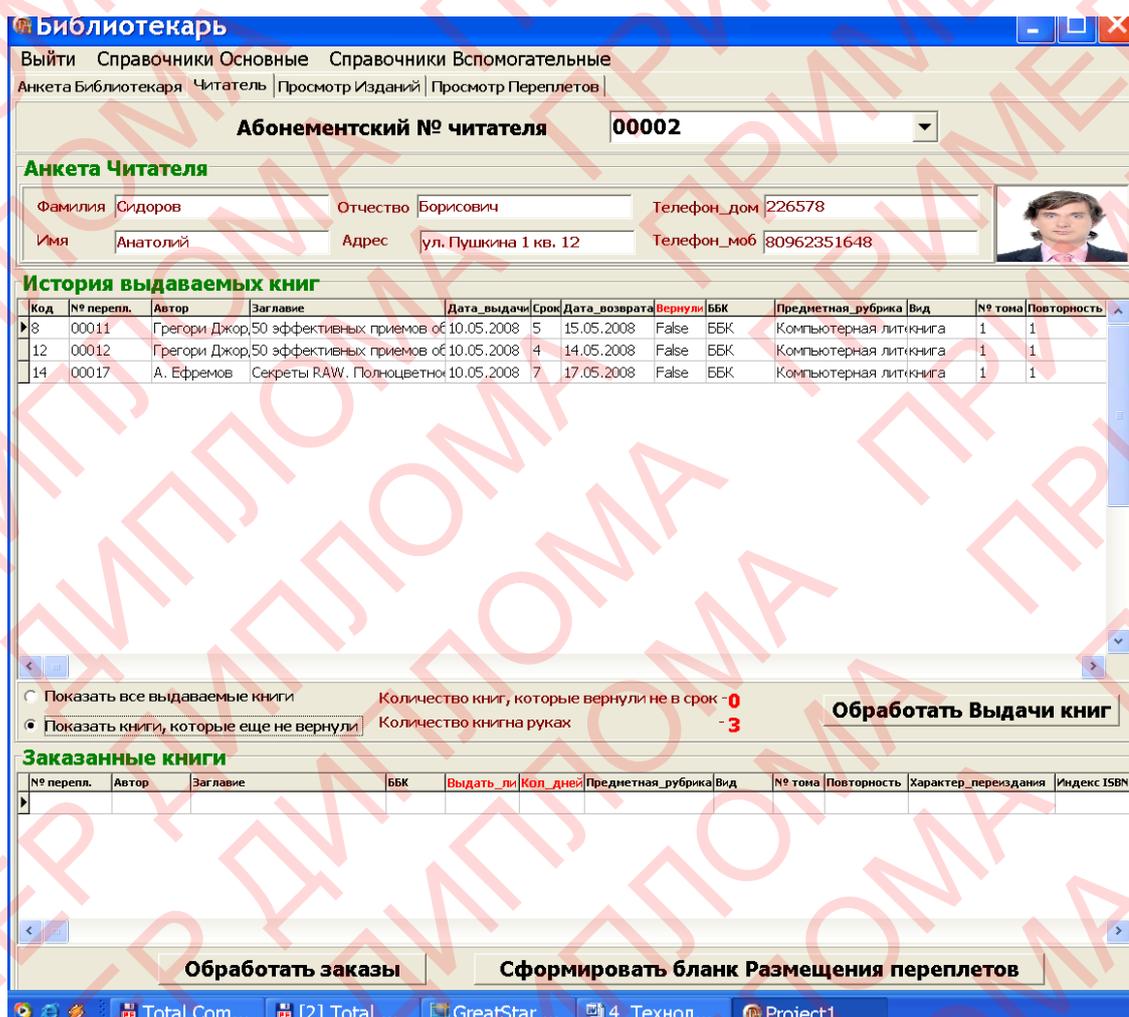


Рисунок 4.17 – Просмотр записей выданных книг, которых читатель еще не вернул

4.4.2.17 Для роли пользователя «библиотекарь» выполнение функции приема сдаваемых книг (обработки списка выдаваемых книг)

Для выполнения указанной функции, если читатель принес сдать литературу в библиотеку, то библиотекарь должен в верхней таблице на второй вкладке напротив соответствующего переплета кликнуть в столбце «Вернули» и нажав правой кнопкой мыши, выбрать из контекстного меню «true». Подобную операцию библиотекарь должен проделать со всей возвращаемой литературой и затем нажать на кнопке «Обработать выдачи КНИГ».

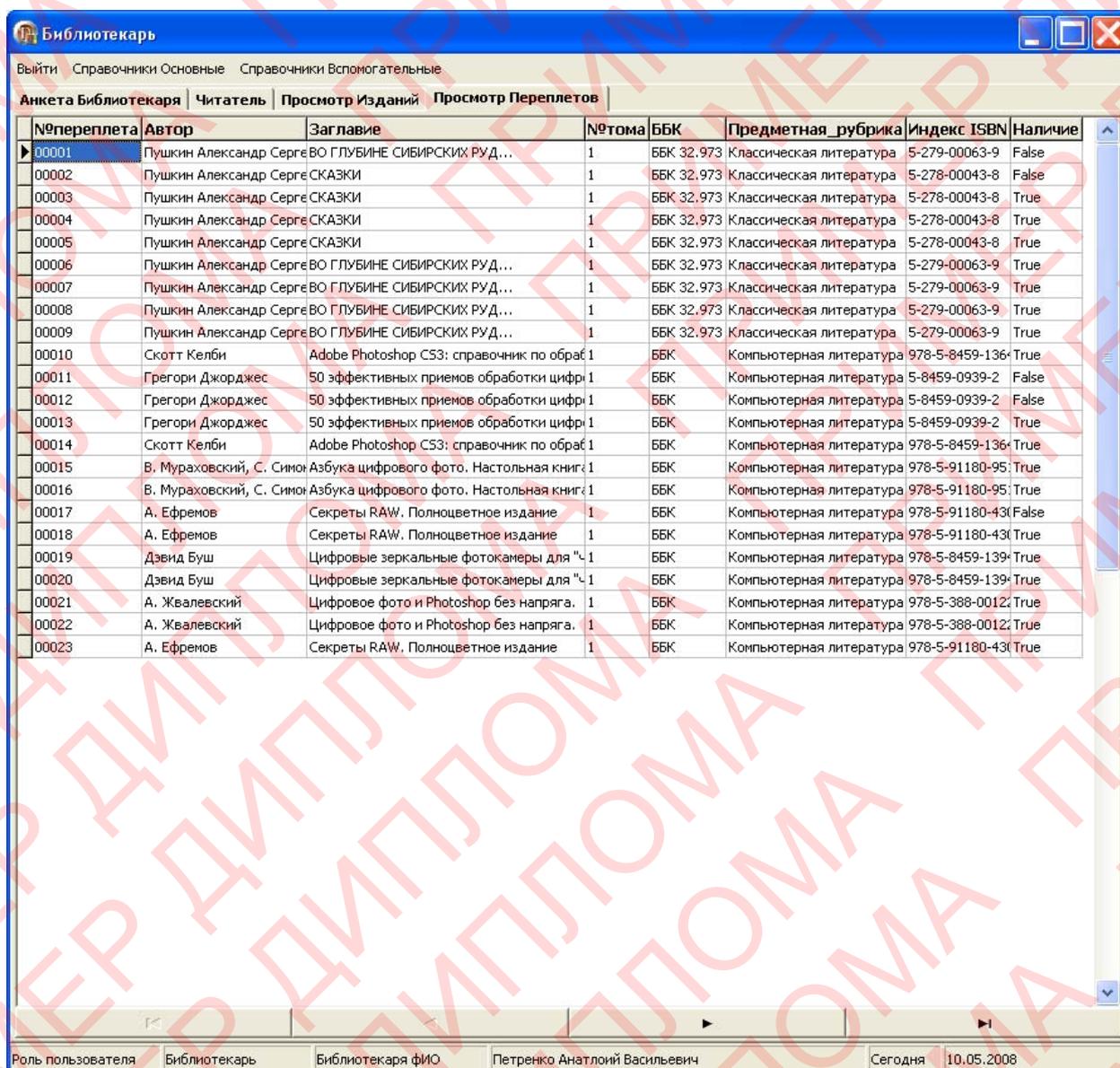
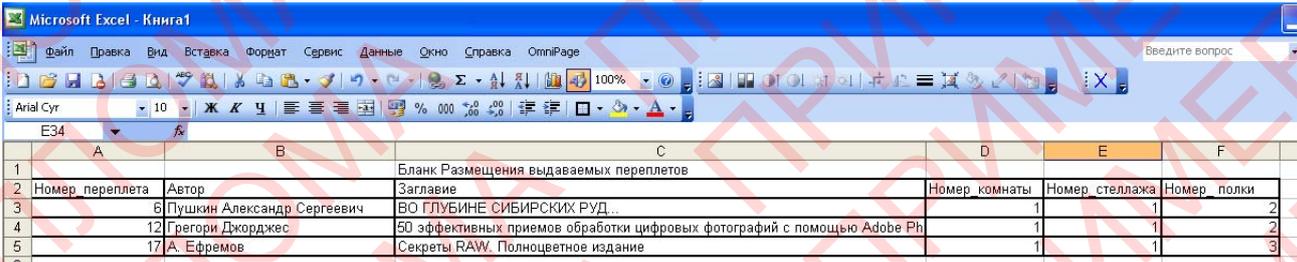


Рисунок 4.18 – Признак отсутствия переплетов

В результате значения полей «Наличие» у соответствующих переплетов примет значение true (см. рис. 4.18) и количество экземпляров соответствующих изданий увеличится.

4.4.2.18 Для роли пользователя «библиотекарь» выполнение функции формирования бланка Размещения переплетов, для быстрой выдачи литературы из хранилища

Для выполнения указанной функции библиотекарь должен нажать на соответствующей кнопке внизу окна. В результате будет создан в программе MS Excel бланк Размещения переплетов (см. рис. 4.19), в котором будут указаны месторасположение каждого заказанного переплета (комната, стеллаж и полка). Этот бланк необходимо распечатать и передать работнику архива, который должен быстро найти указанную в этом бланке литературу и принести ее библиотекарю.



1	A	B	C	D	E	F
			Бланк Размещения выдаваемых переплетов			
2	Номер переплета	Автор	Заглавие	Номер комнаты	Номер стеллажа	Номер полки
3	6	Пушкин Александр Сергеевич	ВО ГЛУБИНЕ СИБИРСКИХ РУД...	1	1	2
4	12	Грегори Джорджес	50 эффективных приемов обработки цифровых фотографий с помощью Adobe Ph...	1	1	2
5	17	А. Ефремов	Секреты RAW. Полноцветное издание	1	1	3
6						

Рисунок 4.19 – Бланк Размещения переплетов

4.4.2.19 Для роли пользователя «библиотекарь» выполнение функции обработки заказа читателя

Для выполнения указанной функции библиотекарь должен в нижнюю таблицу ввести напротив каждого заказанного издания количество дней, на которые выдается литература и нажать на кнопке «Обработать заказы».

В результате этот список опустеет, а заказанная литература перейдет в верхнюю таблицу «выданных книг» (см. рис. 4.20). А так же значения полей «Наличие» у соответствующих переплетов примет значение false (см. рис. 4.18) и количество экземпляров соответствующих изданий уменьшится.

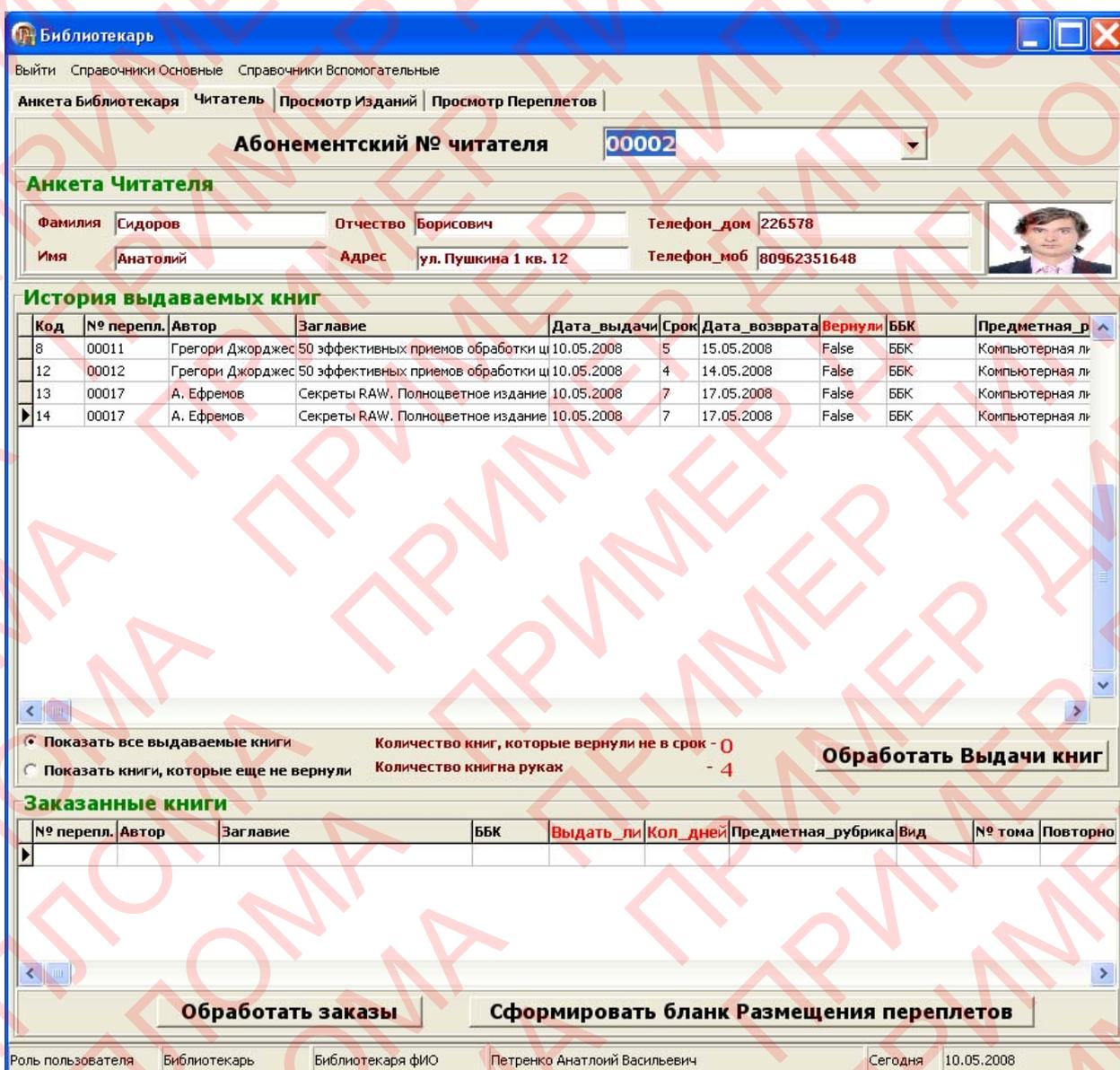


Рисунок 4.20 – Обработанный заказ

4.4.2.20 Для роли пользователя «библиотекарь» выполнение функции заполнения, редактирования или удаления из списка изданий

Для выполнения указанной функции библиотекарь должен в главном меню этого окна программы выбрать команду «Справочники Основные \ Изменение списка Изданий». В результате откроется вспомогательное окно, в котором после редактирования или добавления новой записи – становятся активными кнопки «Выбрать», нажав по которым откроются вспомогательные таблицы из которых необходимо выбрать уже имеющиеся записи или тут же внести их и выбрать (данную операцию необходимо

проделать со всеми полями, напротив которых есть кнопка «Выбрать»), а в поля без кнопки «Выбрать» - необходимо набрать с клавиатуры нужные данные (см. рис. 4.21).

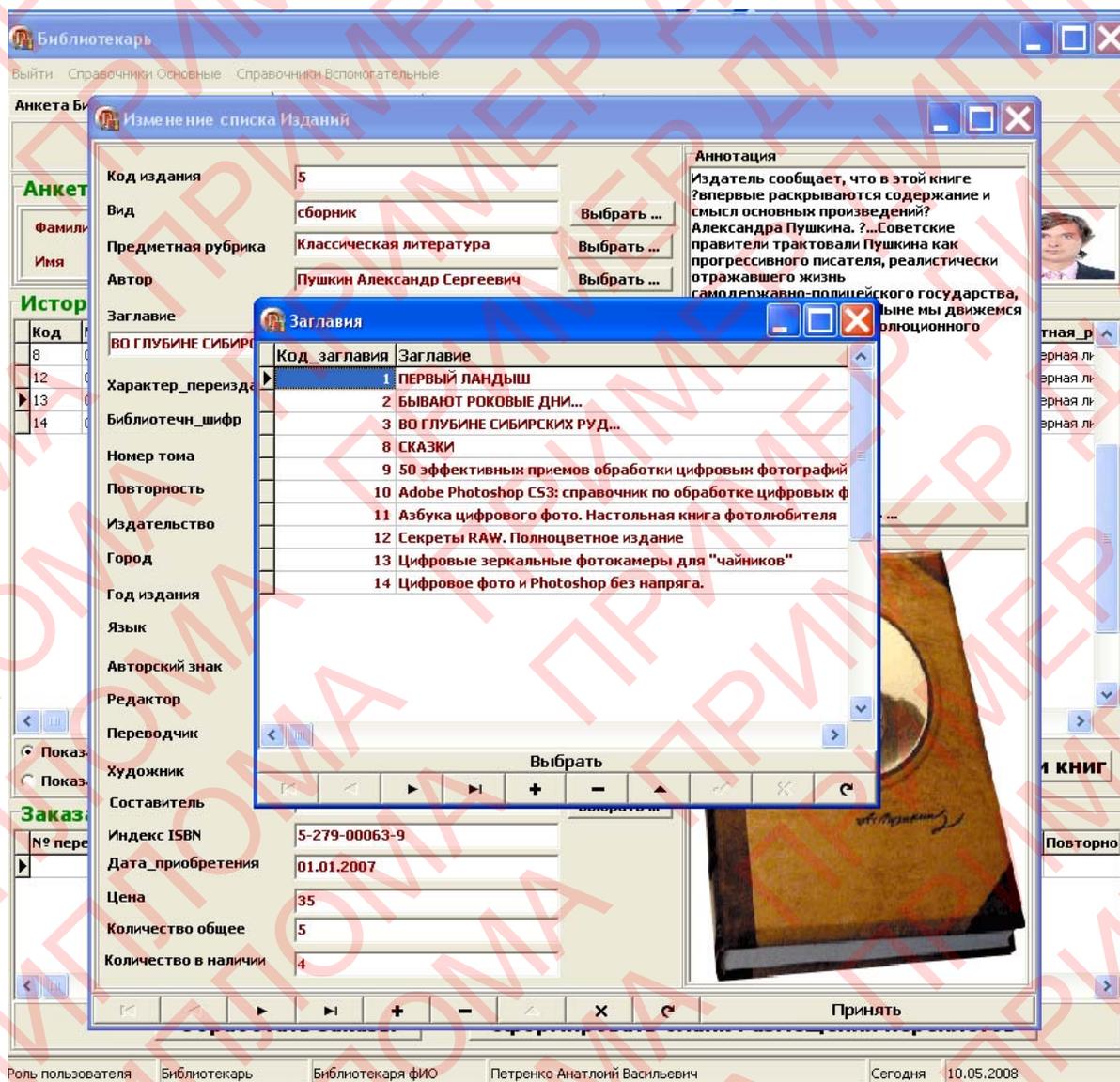


Рисунок 4.21 – Заполнения списка Изданий

4.4.2.21 Для роли пользователя «библиотекарь» выполнение функции заполнения, редактирования или удаления из списка переплетов

Для выполнения указанной функции библиотекарь должен в главном меню этого окна программы выбрать команду «Справочники Основные \ Изменение списка переплетов». В результате откроется вспомогательное окно, в котором после редактирования или добавления новой записи – станет активной кнопка «Выбрать», нажав по которой необходимо будет выбрать из

верхней таблицы уже имеющиеся записи а в поля без кнопки «Выбрать» - необходимо набрать с клавиатуры нужные данные (см. рис. 4.22).

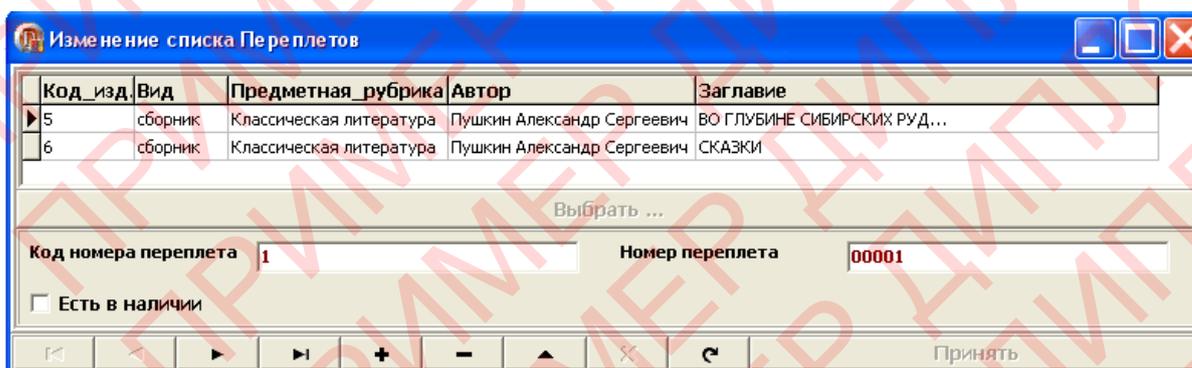


Рисунок 4.22 – Заполнения списка Переплетов

4.4.2.22 Для роли пользователя «библиотекарь» выполнение функции заполнения, редактирования или удаления из списка размещения переплетов

Для выполнения указанной функции библиотекарь должен в главном меню этого окна программы выбрать команду «Справочники Основные \ Изменение списка размещения переплетов». В результате откроется вспомогательное окно, в котором после редактирования или добавления новой записи – станут активными кнопки «Выбрать», нажав по которым необходимо будет выбрать из верхней и боковой таблиц уже имеющиеся записи а в поля без кнопки «Выбрать» - необходимо набрать с клавиатуры нужные данные (см. рис. 4.23).

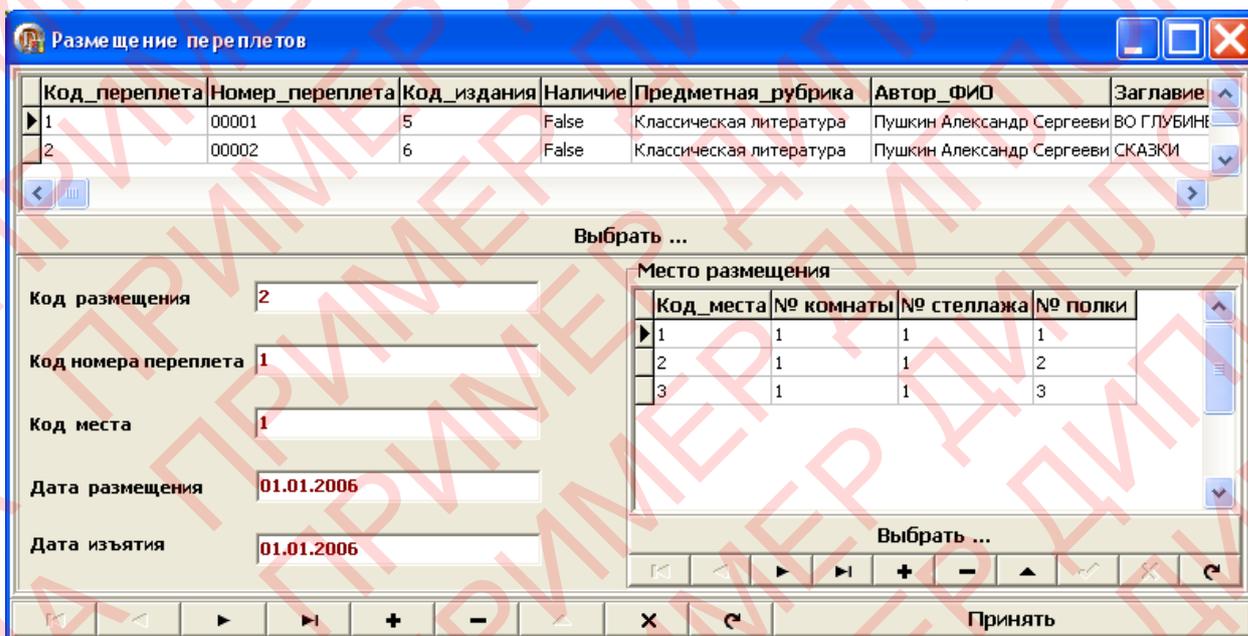


Рисунок 4.23 – Заполнения списка размещения переплетов

4.4.2.23 Для роли пользователя «библиотекарь» выполнение функции заполнения, редактирования или удаления данных из вспомогательных справочников

Для выполнения указанной функции библиотекарь должен в главном меню этого окна программы выбрать из группы «Справочники Вспомогательные» необходимую соответствующую команду. В результате откроется вспомогательное окно в котором можно добавлять, редактировать и удалять данные из вспомогательных справочников.

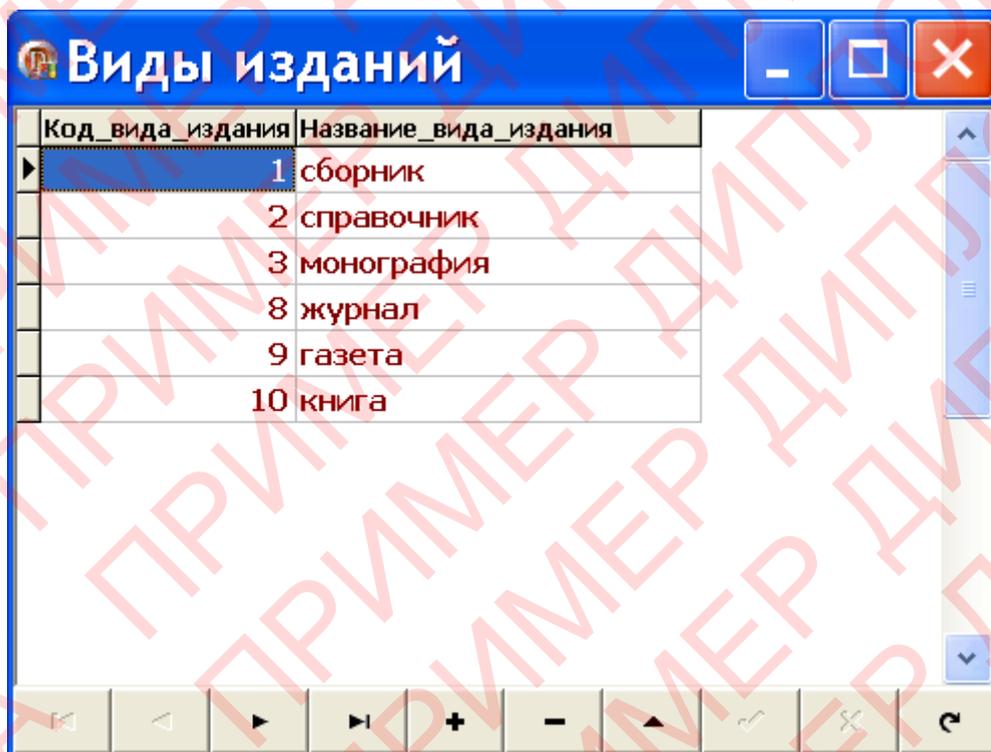


Рисунок 4.24 – Редактирование вспомогательных справочников

В программе имеется двенадцать вспомогательных справочников:

- Виды изданий
- Предметная рубрика
- Издательства
- Редакторы
- Составители
- Характер переиздания
- Языки
- Переводчики
- Художники
- Авторы
- Заглавия
- Аннотации

5. Экономическая часть

5.1 Расчет экономической эффективности программного обеспечения

Эффективность создания и внедрения проектируемой автоматизированной системы библиотекаря должна определяться на основе сравнения с базовым вариантом. В данном случае за базовый вариант принимаем ручную работу библиотекарей.

Главным при разработке программного обеспечения для автоматизации работы библиотекарей, является обеспечение максимальной экономической эффективности, т.е. поредение расчетов с минимальными затратами труда и денежных средств.

Капиталовложения в создание ПП носят единовременный характер:

$$K = K_1 + K_2 + K_3, \quad (5.1)$$

где K_1 - затраты на оборудование, руб.;

K_2 - затраты на лицензионные программные продукты, руб.;

K_3 - затраты на создание программного продукта, руб.

Принимаем $K_1 = 0$, так как оборудование уже было куплено.

Затраты на лицензионные программные продукты для реализации ПП приведены ниже в таблице 5.1.

Таблица 5.1 - Затраты на лицензионные программные продукты

Лицензионный программный продукт	Стоимость, руб.
Borland Developer Studio 2006	70000
Windows XP Pro Rus	4000

Таким образом, затраты на лицензионные программные продукты:

$$K_2 = 70000 + 4000 = 74000 \text{ руб.}, \quad (5.2)$$

Затраты на создание ПП K_3 считаем по формуле:

$$K_3 = Z_1 + Z_2 + Z_3, \quad (5.3)$$

где Z_1 - затраты труда программистов-разработчиков, руб.;

Z_2 - затраты компьютерного времени, руб.;

Z_3 - косвенные (накладные) расходы, руб.

Затраты труда программистов-разработчиков:

$$Z_1 = \sum_{k=1}^K N_k \cdot r_k \cdot T_k \cdot K_{\text{зар}}, \quad (5.4)$$

где N_k - количество разработчиков к-й профессии, чел;

r_k - часовая зарплата разработчика к-й профессии, руб.;

T_k - трудоёмкость разработки для к-го разработчика (количество затраченного разработчиком времени), ч.;

$K_{\text{зар}} = 1,4$ - коэффициент начислений на фонд заработной платы, доли.

Принимаем, $N_k = 1$.

Часовая зарплата разработчика определяется по формуле:

$$r_k = M_k / F_k^{\text{мес}}, \quad (5.5)$$

где M_k – месячная зарплата к-го разработчика, руб.;

$F_k^{\text{мес}}$ - месячный фонд времени его работы, час.

Принимаем месячную зарплату равную 9000 руб, фонд времени работы в среднем 170 часов. Тогда, часовая зарплата разработчика, учитывая (5.5):

$$r_k = \frac{M_k}{F_k^{\text{мес}}} = \frac{9000}{170} = 53 \text{ руб/час.}$$

Трудоёмкость разработки T_k включает время выполнения работ, представленных в таблице 5.3.

Расчет трудоемкости разработки для каждого разработчика осуществляется по формуле:

$$T_k = t_{k1} + t_{k2} + t_{k3} + t_{k4} + t_{k5} = 20 + 100 + 90 + 110 + 75 = 395 \text{ часов,} \quad (5.6)$$

где $t_{k1}, t_{k2}, t_{k3}, t_{k4}, t_{k5}$ - время, затраченное на каждом этапе разработки к-м разработчиком, час.

Затраты труда программиста-разработчика, учитывая (5.4):

$$З_1 = \sum_{k=1}^K N_k \cdot r_k \cdot T_k \cdot K_{\text{зар}} = 1 \cdot 53 \cdot 395 \cdot 1,4 = 29309 \text{ руб.}$$

Затраты компьютерного времени:

$$З_2 = C_k \cdot F_0, \quad (5.7)$$

где C_k - стоимость компьютерного часа, руб.;

$F_0 = 395 \text{ ч}$ - затраты компьютерного времени на разработку программы, час.

Стоимость компьютерного часа исчисляется по формуле:

$$C_k = C_A + C_3 + C_{\text{ТО}} \quad (5.8)$$

где C_A - амортизационные отчисления, руб.;

C_ϕ - энергозатраты, руб.;

C_{To} - затраты на техобслуживание, руб.

Амортизационные отчисления:

$$C_A = C_i \cdot N_A / F_{\text{год}}, \quad (5.9)$$

где $C_i=20000$ руб. - балансовая стоимость i -го оборудования, которое использовалось для создания ПП, руб.;

N_A - годовая норма амортизации i -го оборудования, доли;

$F_{\text{год}}$ - годовой фонд времени работы i -го оборудования, час.

Таблица 5.2 - Этапы и содержание работ

Этапы работ	Содержание работ
1. Техническое задание	Краткая характеристика программы; основание и назначение разработки; требования к программе и программной документации; стадии и этапы разработки программы; порядок контроля и приёмки выполнения.
2. Эскизный проект	Предварительная разработка структуры входных и выходных данных; уточнение метода решения задачи; разработка и описание общего алгоритма решения; разработка технико-экономического обоснования и пояснительной записки.
3. Технический проект	Уточнение структуры входных и выходных данных, определение формы их представления; разработка подробного алгоритма; разработка структуры программы; окончательное определение конфигурации технических средств

4. Рабочий проект	Описание программы на выбранном языке; отладка; разработка методики испытаний; проведение предварительных испытаний (тестирование); корректировка программы; разработка программной документации.
5. Внедрение	Подготовка и передача программы для сопровождения; обучение персонала использованию программы; внесение корректировок в программу и документацию.

Для выбранного оборудования принимаем одинаковую норму амортизации и годовой фонд времени работы $N_A = 0,25$, $F_{год} = 2000ч$.

Учитывая (5.9) получим:

$$C_a = C_i \cdot N_A / F_{год} = 20000 \cdot 0,25 / 2000 = 2.5 \text{ руб.}$$

Энергозатраты составят:

$$C_э = P_э \cdot C_{кВт} = 0,4 \cdot 0,75 = 0,3 \text{ руб.}, \quad (5.10)$$

где $P_э$ - расход электроэнергии, потребляемой компьютером, кВт/ч;

$C_{кВт}$ - стоимость 1 кВт/ч электроэнергии, руб.

Затраты на техобслуживание:

$$C_{ТО} = r_{ТО} \cdot \lambda, \quad (5.11)$$

где $r_{ТО}$ - часовая зарплата работника обслуживающего оборудование, руб;

принимаем аналогично (5.5) $r_{ТО} = 4500 / 170 = 26 \text{ руб./час}$;

λ - периодичность обслуживания:

$$\lambda = N_{ТО} / F_{мес} = 1 / 166,7 = 0,006, \quad (5.12)$$

где N_{TO} - количество обслуживаний оборудования в месяц, принимаем $N_{TO} = 1$;

$F_{мес} = F_{год} / 12 = 2000 / 12 = 166,7$ - месячный фонд времени работы оборудования, час.

Затраты на техобслуживание, учитывая формулу (5.11):

$$C_{ТО} = 26 * 0,006 = 0,156 \text{ руб.}$$

Определим стоимость компьютерного часа:

$$C_k = C_a + C_э + C_{то} = 2,5 + 0,3 + 0,156 = 2,95 \text{ руб.}$$

Таким образом, по формуле (5.7) определим затраты компьютерного времени:

$$Z_2 = 2,95 * 395 = 1165,25 \text{ руб.}$$

Косвенные расходы Z_3 определяются по формуле:

$$Z_3 = C_1 + C_2 + C_3 \quad (5.13)$$

где C_1 - расходы на содержание помещений (составляют 2 - 2,5 % от стоимости здания), руб.;

C_2 - расходы на освещение, отопление, охрану и уборку помещения (составляют 0,2 - 0,5 % от стоимости здания), руб.;

C_3 - прочие расходы (стоимость различных материалов, используемых при разработке проекта, услуги сторонних организаций и т.п. - составляют 1 - 2 % от стоимости вычислительной техники).

Рассматриваемое помещение зал библиотеки (в котором планируется проведение проектировочных работ) представляет собой комнату, расположенную на третьем этаже 2-ом этажного кирпичного здания. Параметры помещения: ширина комнаты - 5 м, длина - 7 м, высота - 3,5 м, общая площадь помещения составляет 35 м², общий объем помещения - 122,5 м³. Таким образом, учитывая стоимость 1 м² - 125 руб., общая стоимость помещения составляет 4375 руб.

Затраты на содержание помещений C_1 составляют $4375 \cdot 0,02 = 87,5$ руб., затраты на освещение, отопление охрану и уборку помещений C_2 - $4375 \cdot 0,004 = 17,5$ руб., прочие расходы составляют (принимая стоимость вычислительной техники 20000 руб.) $C_3 = 20000 \cdot 0,02 = 4000$ руб.

Таким образом, косвенные расходы:

$$Z_3 = C_1 + C_2 + C_3 = 87,5 + 17,5 + 4000 = 4105 \text{ руб.}$$

Определим затраты на создание ПП:

$$K_3 = Z_1 + Z_2 + Z_3 = 29309 + 1165,25 + 4105 = 34579,25 \text{ руб.}$$

Капитальные затраты на выполнение и реализацию ПП составят:

$$K = K_1 + K_2 + K_3 = 0 + 74000 + 34579,25 = 108579,25 \text{ руб.}$$

Т.к. объектом автоматизации является автоматизация деятельности библиотеки, то источником экономии может служить снижение затрат на выполняемые ими работы.

Годовая экономия от снижения затрат на выполняемые работы библиотекарями определяется следующим образом:

$$\Delta \Gamma = C_p \cdot T_p - C_a \cdot T_a, \quad (5.14)$$

где C_p , C_a — трудоемкость работы библиотекарей в ручном и автоматизированном вариантах, час;

T_p , T_a — часовая себестоимость работы библиотекарей в ручном и автоматизированном вариантах, руб.;

Себестоимость работы библиотекарей в ручном варианте:

$$C_p = C_p^1 + C_p^2, \quad (5.15)$$

где C_p^1 — затраты на оплату труда библиотекарей, руб.;

C_p^2 — косвенные расходы, руб.

Определяем затраты на оплату труда персонала библиотеки:

$$C_p^1 = \sum_{k=1}^n N_k \cdot R_k \cdot k_{зар} \quad (5.16)$$

где $N_k = 7$ — количество библиотекарей к-й профессии, участвующих в ручном варианте работы;

R_k — часовая зарплата одного работника библиотекарей к-й профессии, руб.;

$k_{зар} = 1,4$ — коэффициент начислений на фонд зарплаты;

$k = 1$ — число различных профессий библиотекарей.

Часовая зарплата библиотекаря рассчитывается следующим образом:

$$R_k = 7500 / 170 = 44 \text{ руб.},$$

где $Mk = 7500$ руб. — месячный оклад библиотекаря, руб.;

$F_k^{мес} = 170$ ч — месячный фонд времени работы библиотекаря (при ручном варианте разработки), час.

Получаем затраты на оплату труда персонала по формуле (5.16):

$$C_p^1 = 7 \cdot 44 \cdot 1,4 = 61,6 \text{ руб.}$$

Косвенные расходы C_p^2 рассчитываются аналогично формуле (5.16).

Принимаем, что при автоматизированном и ручном вариантах затраты на содержание помещений и расходы на освещение, отопление охрану и уборку помещений совпадают, а прочие расходы составляют 1%–2 % от фонда заработной платы.

Фонд зарплаты составляет: $C_p^1 = 61,6$ руб.

Затраты на содержание помещений C_1 составляют 0,1 руб./час, затраты на освещение, отопление охрану и уборку помещений C_2 – 17,5 руб./час, прочие расходы составляют 0,25 руб./час.

Таким образом, косвенные затраты:

$$C_p^2 = 61,6 + 17,5 + 0,25 = 79,35 \text{ руб.}$$

Себестоимость работы библиотекарей в ручном варианте составит:

$$C_p = 61,6 + 79,35 = 140,95 \text{ руб.}$$

Себестоимость работы библиотекарей в автоматизированном варианте:

$$C_A = C_A^1 + C_A^2 + C_A^3 \quad (5.17)$$

где C_A^1 — затраты на оплату труда персонала, руб.;

C_A^2 — стоимость компьютерного времени, руб.;

C_A^3 — косвенные расходы, руб.

Затраты на оплату труда персонала с учетом (5.5) и (5.16)

При месячной заработной плате 4500 руб.: $R = 4500/170 = 26$ руб./час.

Тогда, затраты на оплату труда:

$$C_{A1} = 1 \cdot 26 \cdot 1,4 = 36,4 \text{ руб./час.}$$

Стоимость компьютерного времени C_A^2 определяется по формуле (5.8) и составляет 17,5 руб.

Косвенные расходы C_A^3 рассчитываются по формуле (5.13) и составляют:

$$C_A^3 = 36,4 + 17,5 + 0,25 = 54,15 \text{ руб.}$$

Себестоимость проектирования в автоматизированном варианте:

$$C_A = C_A^1 + C_A^2 + C_A^3 = 36,4 + 17,5 + 54,15 = 108,05 \text{ руб.}$$

Исходя из опытно-статистического анализа, можно сделать вывод, что время обработки библиотекарем одного заказа читателя в ручном режиме, составляет около 20 минут (1/3 часа), а прогнозируемое время обработки библиотекарем одно заказа читателя в автоматизированном варианте – 5 минуты (1/12 часа).

Тогда годовая экономия, согласно формуле (5.14):

$$\mathcal{E}_T = 1200 \cdot (140,95 \cdot 1/3 - 108,05 \cdot 1/12) = 45576 \text{ руб./год.}$$

Годовой экономический эффект:

$$\mathcal{E}_\phi = \mathcal{E}_z - E_n \cdot K \quad (5.18)$$

где $E_n = 0,2$ - коэффициент, показывающий, какая часть капиталовложений затрачена за год.

$$\Xi_{\phi}=45576 - 0,2 * 108\ 579,25 = 23860,15 \text{ руб.}$$

Эффективность внедрения программного продукта

$$\Xi = \frac{\Xi_{\phi}}{K} \quad (5.19)$$

$$\Xi=(23860,15 / 108579,25)*100\% = 22 \%$$

Срок окупаемости капиталовложений – период времени, в течение которого окупаются затраты на ПО, определяется по формуле:

$$T_p = \frac{K}{\Xi_{\Gamma}} \quad (5.20)$$

$$T_p=108579,25 / 45576 \approx 2,36 \text{ года}$$

При эффективном использовании капиталовложений расчётный срок окупаемости T_p должен быть меньше нормативного $T_H = 2,4$:

$$T_p < T_H;$$
$$2,36 < 2,4.$$

5.2. Выводы по экономической эффективности создания программного обеспечения

Данный экономический расчет показывает, что разработка и использование программного продукта является экономически оправданным и целесообразным. Об этом свидетельствует следующее:

- годовая экономия текущих затрат при внедрении автоматизированной системы библиотекаря составит 45 576 руб.;
- экономическая эффективность составляет 23 860,15 руб.;
- эффективность внедрения программного продукта 22 %;
- срок окупаемости капиталовложений в ПП составит порядка 2,36 года.

6. Охрана труда и окружающей среды

6.1 Обеспечение условий труда на рабочем месте библиотекаря

Все здания и помещения, в которых проводится эксплуатация ЭВМ, должны соответствовать требованиям ГСанНиП 3.3.2 - 007 – 98, СН 912 –78, СНиП II.01.02 – 85, СНиП II.05.02 – 89.

Эти стандарты устанавливают нормы площади и объема помещений (площадь - не менее 6,0 м², объем - не менее 20,0 м³) с учетом максимального количества лиц, которые одновременно работают в смене.

Рассматриваемое помещение библиотеки (в котором планируется применение автоматизированной системы библиотекаря) представляет собой комнату, расположенную на втором этаже 4-х этажного кирпичного здания.

Параметры помещения: ширина комнаты – 7 м, длина – 8 м, высота – 3,5 м, общая площадь помещения составляет 56 м², общий объем помещения – 196 м³. В помещении расположено 8 рабочих мест, т.е. на одного человека приходится 7 м², что соответствует норме. Объем помещения, приходящийся на одного человека 24,5 м³, что также не противоречит нормам. В помещении имеется два окна размером 2,5х3 м, выходящие на восток.

Напряжение источников питания в помещении библиотеки – 220В.

В помещении располагается 8 компьютеров, примерно такой конфигурации: ПК IP4 2400 MHz/512 DDRII/160 Gb/256 MB PCI-E GeForce 8600GT, монитор ЖК 17" Samsung 740FN, TCO'03.

В помещении находится два лазерный принтера HP LaserJet 1000, HP LaserJet 4V и сканер.

Для снижения шума в помещениях, компьютеры установлены на специальные амортизационные прокладки, предусмотренные нормативными документами.

Уровень шума в помещении не превышает 50 дБ, что соответствует требованиям ГОСТ 12.1.003-83, так как принтеры и системные блоки компьютеров, находящиеся в помещении, работают практически бесшумно (шум не более 20 дБ).

Организация рабочих мест осуществляется на основе современных эргономических требований. Рабочее место для выполнения работ в положении сидя оборудовалось в соответствии с требованиями ГОСТ 12.2.032-78 и требованиям технической эстетики, с учетом психологических особенностей человека и его антропометрических данных.

При организации условий труда необходимо учитывать воздействие на работающих опасных и вредных производственных факторов, которые могут привести к травме или другому внезапному резкому ухудшению здоровья и заболеванию или снижению работоспособности.

В соответствии с ГОСТ 12.0.003 – 91 все возникающие в производственных условиях опасные и вредные производственные факторы делятся на следующие группы: физические; химические; психофизиологические и биологические.

В помещении на библиотекаря могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

К психологически вредным факторам, воздействующим на библиотекаря в течение его рабочей смены, можно отнести следующие:

- нервно - эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Все опасные и вредные производственные факторы регулируются соответствующими нормативными документами, устанавливающими допустимые и оптимальные значения по каждому из параметров. Параметры ионизирующего излучения мониторов компьютеров находятся в допустимых пределах и регулируются стандартами ТСО-*

Все дисплейные терминалы и другое оборудование, применяемое в данном помещении, имеет гигиенический сертификат, соответствует требованиям действующих стандартов, правилам национальных стандартов и имеет в паспорте соответствующую отметку об этом.

6.2. Анализ состояния охраны труда в помещении

При приеме на работу с каждым работником библиотеки проводится инструктаж по технике безопасности - вводный. Затем проводится инструктаж на рабочем месте с выдачей библиотекарю памятки и личной карточки обучения в области охраны труда. Спустя 90 дней с работником проводится тематический инструктаж руководителем подразделения. В последующем, по мере надобности проводятся инструктажи ответственными лицами за охрану труда и техники безопасности. Это происходит при вводе в эксплуатацию новой техники или при вводе новых нормативных актов по охране труда, при изменении условий труда, при выполнении ремонтных работ. Работа работников библиотеки относится к категории 1 – легкая физическая работа, которая выполняется сидя.

Электробезопасность в помещении библиотеки обеспечивается техническими способами и средствами защиты, а так же организационными и техническими мероприятиями.

Согласно ОНТ – 2486 здание, в котором располагается библиотека, имеет II степень огнестойкости и относится к категории В. Согласно нормативным требованиям помещение оборудовано автоматической противопожарной сигнализацией и средствами огнетушения.

Требования пожарной безопасности соблюдаются согласно инструкции библиотеки.

В помещении библиотеки обеспечивается:

- выполнение установленного противопожарного режима;
- поддержка средств противопожарной защиты и связи в исправленном состоянии;
- слежение за исправностью приборов отопления, вентиляции, электроустановок, а также принятие мер по избеганию неисправностей, которые могут привести к пожару.

Персональные компьютеры и другие электроустановки по окончании рабочего дня отключаются от сети.

Из средств пожаротушения в помещении находятся огнетушители: углекислотный ОУ-5 и порошковые (ОП-2, ОП-5), которые позволяют тушить пожары в помещениях с вычислительной техникой.

6.3. Анализ вентиляции, искусственного и природного освещения

Воздухообмен при естественной вентиляции происходит вследствие разности температур воздуха в помещении и наружного воздуха, а также в результате действия ветра.

Разность температур воздуха внутри (более высокая температура) и снаружи помещения, а следовательно, и разность плотностей вызывают поступление холодного воздуха в помещение и вытеснение из него теплого воздуха. При действии ветра с заветренной стороны зданий создается пониженное давление, вследствие чего происходит вытяжка теплого или загрязненного воздуха из помещения; с наветренной стороны здания создается избыточное давление и свежий воздух поступает в помещение на смену вытягиваемому воздуху. Работа ряда вытяжных вентиляционных устройств в сильной степени также зависит от обдува их ветром.

Естественная вентиляция производственных помещений может быть *неорганизованной и организованной*.

При неорганизованной вентиляции поступление и удаление воздуха происходит через неплотности и поры наружных ограждений (инфильтрация), и через окна, форточки, специальные проемы (проветривание).

Организованная (поддается регулировке) естественная вентиляция производственных помещений осуществляется аэрацией и дефлекторами.

Для вентиляции помещения используется естественная неорганизованная вентиляция (через окна) – проветривание и кондиционирование воздуха.

6.3.1. Расчет достаточности вентиляции в помещении

Согласно СНИП 2.09.04-87 в помещениях для обслуживания ЭВМ объем рабочего помещения, который приходится на каждого рабочего, должен составлять не меньше 40 м^3 .

Размеры исследуемого помещения: $7 \times 8 \times 3,5 \text{ м}$.

Размер секции окна для проветривания: $2,5 \times 0,5 \text{ м}$.

Объем помещения составит $V_n=7*8*3,5=196 \text{ м}^3$ и на каждого человека приходится $24,5 \text{ м}^3$.

Так как фактическое значение объема на одного человека ниже установленного для нормальной работы в помещении необходимо обеспечить постоянный воздухообмен с помощью вентиляции не меньше, чем $L^1=30 \text{ м}^3/\text{ч}$ на каждого человека. Таким образом, необходимый воздухообмен $L_n, \text{ м}^3/\text{ч}$, определяется по формуле:

$$L_n=L^1*n, \quad (6.1)$$

где n – максимально возможное количество людей в помещении, включая читателей и библиотекарей.

Пусть в помещении одновременно могут находиться не больше 13 читателей и 5 библиотекарей, тогда $n=18$ и $L_n=30*18=540 \text{ м}^3/\text{ч}$.

Определим фактический воздухообмен, который осуществляется в помещении с помощью природной вентиляции как неорганизованно через различные неплотности оконных и дверных проемов, так и организовано через секцию окна в оконном проеме или специальные вентиляционные каналы. В данном случае – секции окна для проветривания.

Фактический воздухообмен $L_\phi, \text{ м}^3/\text{ч}$, определим по формуле:

$$L_\phi=\mu*F*V*3600, \quad (6.2)$$

где μ - коэффициент расхода воздуха, который принимает значение в рамках 0,1-0,8 (возьмем среднее значение 0,55);

F – площадь форточки, м^2 ;

V – скорость выхода воздуха через секцию окна для проветривания, $\text{м}/\text{с}$.

Скорость выхода воздуха через секцию окна для проветривания определяется по формуле:

$$V = \sqrt{\frac{2g * H_2}{\gamma_c}}, \quad (6.3)$$

где g – ускорение свободного падения, $g=9,8$ м/с²;

H_2 – тепловой напор, под действием которого будет производиться выход воздуха из форточки, который можно вычислить по формуле:

$$H_2 = h_2 (\gamma_{вн} - \gamma_c), \quad (6.4)$$

где h_2 – высота от плоскости равных давлений до центра форточки, обратно пропорциональная квадрату площади сечения окна ;

$\gamma_{вн}$, γ_c – соответственно объемный вес воздуха снаружи помещения и внутри его, кгс/м³;

В общем случае объемный вес воздуха определяется по формуле:

$$\gamma = 0,465 * P_{\delta} / T, \quad (6.5)$$

где P_{δ} – барометрическое давление, мм рт. ст. (принимается 750 мм рт. ст.);

T – температура воздуха, °К (согласно ГОСТ 12.1.005-88 для холодного периода года температура в помещении должна находиться в пределах 19°С - 21°С, или 272°К-274°К; для воздуха снаружи помещения температура устанавливается по СНИП 2.04.05-91 для зимы не ниже -11°С, или 242°К).

По данным термометра, $T_{вн}=247^{\circ}\text{К}$, $T_c=272^{\circ}\text{К}$. Тогда очевидно, что $\gamma_{вн} \neq \gamma_c$, а значит:

$$h_2 = 1 / (2,5 * 0,5)^2 = 0,64 \text{ 1/м}^4,$$

Для теплого периода года:

$$\gamma_{вн} = 0,465 * 750 / 272 = 1,28 \text{ кгс/м}^3,$$

$$\gamma_c = 0,465 * 750 / 278 = 1,24 \text{ кгс/м}^3,$$

$$H_2 = 0,64(1,28 - 1,24) = 0,0256 \text{ кгс*м}$$

$$V = \sqrt{\frac{2 * 9,8 * 0,0256}{1,28}} = 0,62 \text{ м/с},$$

$$L_{\phi} = 0,55 * (2,5 * 0,5) * 0,62 * 3600 = 1534,5 \text{ м}^3/\text{ч}$$

Для холодного периода года:

$$\gamma_c = 0,465 * 750 / 242 = 1,44 \text{ кгс/м}^3,$$

$$\gamma_{\text{вн}} = 0,465 * 750 / 272 = 1,28 \text{ кгс/м}^3,$$

$$H_2 = 0,64(1,44 - 1,28) = 0,1024 \text{ кгс*м}$$

$$V = \sqrt{\frac{2 * 9,8 * 0,1024}{1,28}} = 1,25 \text{ м/с},$$

$$L_{\phi} = 0,2 * (2,5 * 0,5) * 1,25 * 3600 = 1125 \text{ м}^3/\text{ч}$$

Сравнивая фактический и необходимый воздухообмен, можно сделать вывод, что вентиляция эффективна ($L_{\phi} > L_n$) и нет необходимости в дополнительных мерах по его улучшению.

Для вентиляции помещения используется естественная неорганизованная вентиляция (через окна) и организованная вентиляция – проветривание и кондиционирование воздуха.

Значение скорости передвижения воздуха находится в допустимых пределах.

Освещение рабочего места - важнейший фактор создания нормальных условий труда. Освещение помещений и рабочих мест с ПК осуществляется в соответствии со СНиП II.4.79.

6.3.2. Анализ освещенности помещения

Так как при работе библиотекаря необходима точность выполнения работ, то к освещению предъявляются специфические требования. Наиболее удобным здесь является комбинированное освещение (СНиП II.4.79).

В целях создания равномерного естественного освещения помещений следует ориентировать оконные проемы на северную и северо-восточную стороны горизонта. Так как окна помещения выходят на юго-восток, то для окон предусмотрены солнцезащитные регулируемые устройства типа жалюзи.

Стандартом предусмотрены следующие нормы параметров искусственного освещения для помещений библиотек:

- освещенность – 400-500 лк;
- показатель дискомфорта – не более 40 лк;
- коэффициент пульсации освещенности – не более 10%.

Для искусственного освещения помещения зала библиотеки используются люминесцентные лампы. Общее количество ламп - $6 \times 4 = 24$.

6.3.3. Расчет естественного освещения

Нормируемое значения коэффициента естественной освещенности E_n (%) с учетом характера зрительной работы, светового климата, солнечности климата, зависящее от географического расположения здания, определяется по формуле:

$$e_{n, I, II, IV, Y}^{III} = e_n^{III} \cdot m \cdot c, \quad (6.6)$$

где - $e_n^{III} = 1,5$ - значение КЕО (%) при рассеивании света от небосвода, определяемое с учетом характера зрительной работы (работа средней точности IV разряда при боковом естественном освещении);

$m = 0,9$ – коэффициент светового климата (без учета прямого солнечного света), в зависимости от расположения здания;

$c = 0,8$ – коэффициент солнечности климата (с учетом прямого солнечного света), в зависимости от пояса светового климата и ориентации здания относительно сторон горизонта (для IV световом пояса при световых проемах ориентированных на восток).

Полученный коэффициент округлен до десятых.

$$e'' = 1,5 \cdot 0,9 \cdot 0,8 = 1,08 \approx 1,1.$$

Далее производится расчет фактического значения КЕО:

При боковом освещении КЕО рассчитывается по формуле:

$$e_{\phi} = \frac{100 \cdot S_0 \cdot \tau_0 \cdot r_1}{S_n \cdot \eta_0 \cdot k_3 \cdot k_{30}}, \quad (6.7)$$

где $S_0=7,5$ - площадь окон в рассматриваемом помещении, м²;

τ_0 - общий коэффициент светопропускания, определяемый по формуле:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \cdot \tau_5, \quad (6.8)$$

где $\tau_1=0,8$ - коэффициент светопропускания материала, (для стекла оконного листового двойного);

$\tau_2=0,75$ - коэффициент, учитывающий потери света в переплетах светопроема, (для деревянных спаренных переплетов);

$\tau_3=1$ - коэффициент, учитывающий потери света в несущих конструкциях (при боковом освещении);

$\tau_4=1$ - коэффициент, учитывающий потери света в солнцезащитных устройствах (при отсутствии солнцезащитных устройств);

$\tau_5=1$ - коэффициент, учитывающий потери света в защитной сетке, устанавливаемой под фонарем (при боковом освещении);

$$\tau_0 = 0,8 \cdot 0,75 \cdot 1 \cdot 1 \cdot 1 = 0,6;$$

$k_3=1,3$ - коэффициент запаса;

$r_1=2,9$ - коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию, (отношение глубины помещения 8 м к высоте от уровня условно рабочей поверхности до верха окна 2 м $8/2=4$; отношение расстояния расчетной точки от наружной стены к глубине помещения $3,5/8=0,43$; отношение длины помещения к его глубине $8/8=1$; и средневзвешенном коэффициенте отражения от потолка, стен, пола $\rho_{cp}=0,4$);

$S_n=56$ площадь пола помещения, м²;

$\eta_0=21$ - световая характеристика окон;

$k_{зд}=1$ - коэффициент, учитывающий затенение окон противостоящими зданиями.

$$e_{\phi} = \frac{100 \cdot 7.5 \cdot 0,6 \cdot 2,9}{56 \cdot 21 \cdot 1,3 \cdot 1} = 0,935 \approx 1,0.$$

При сравнении полученного фактического значение с нормируемым получаем $e_{\phi} \approx e_n=1,0$.

Следовательно естественное освещение помещения соответствует СНИП П-4-79.

6.3.4. Расчет искусственного освещения

Задачей расчета является определение освещенности осветительных ламп и сравнение с нормативным значением.

Для расчета общего равномерного освещения при горизонтальной рабочей поверхности основным является *метод светового потока* (коэффициента использования), учитывающий световой поток, отраженный от потолка и стен.

Для искусственного освещения помещения отдела САПР используются люминесцентные лампы типа ЛД80. Световой поток, излучающий такой лампой равен 4070 лм. Общее количество ламп - $6 \times 4 = 24$.

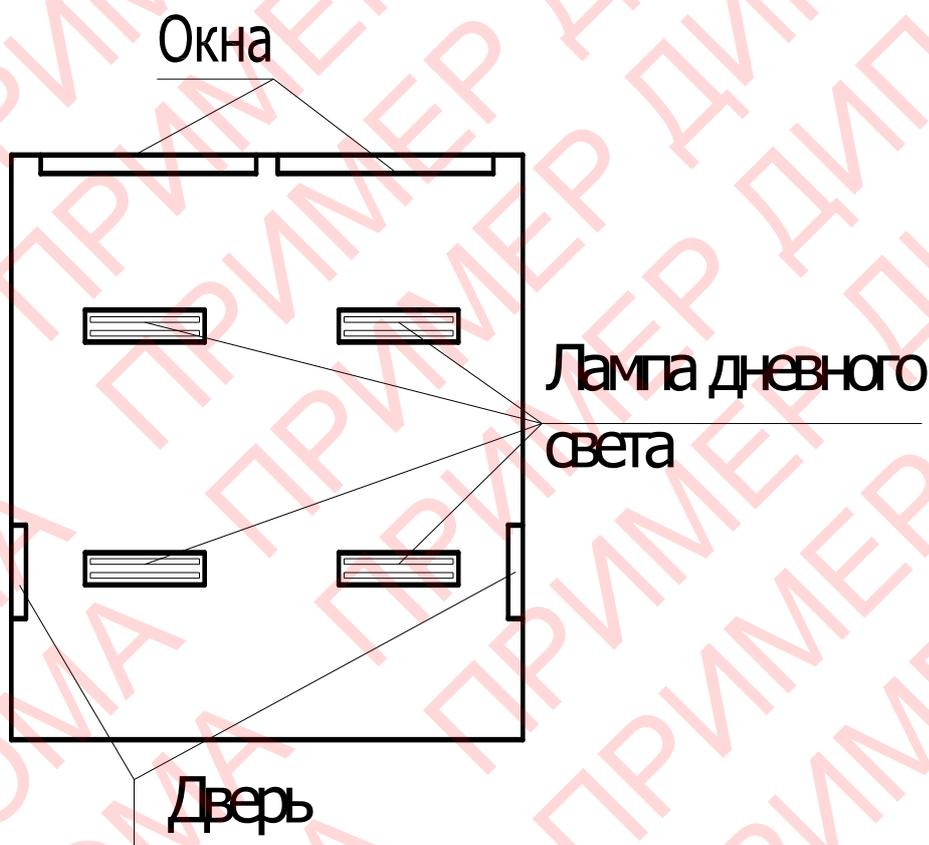


Рисунок 6.1 Схема расположения ламп

Световой поток F_n (лм) группы ламп светильника при люминесцентных лампах рассчитывают по формуле:

$$F_n = \frac{E_n \cdot S \cdot z \cdot k}{N \cdot \eta} \quad (6.9)$$

Выразим значение E из формулы для расчета фактической освещенности:

$$E_\phi = \frac{F_n \cdot N \cdot \eta}{S \cdot z \cdot k} \quad (6.10)$$

где E_n - нормированная минимальная освещенность, ($E_n = 300$ лк);

S - площадь освещаемого помещения, м²;

Z - коэффициент минимальной освещенности, равный отношению $\frac{E_{cp}}{E_{min}}$, значения которого обычно находятся в пределах 1,1—1,5 (в среднем 1,2);

k - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (принимается в пределах 1,3-1.5);

N - число светильников в помещении;

η - коэффициент использования светового потока ламп, зависящий от к.п.д. и кривой распределения силы света светильника, коэффициента отражения потолка (p_n) и стен (p_c), высоты подвеса светильников и размеров помещения. $\eta=0,45$

$$E_{\phi} = \frac{4070 \cdot 24 \cdot 0.45}{56 \cdot 1,2 \cdot 1,5} = 436 \text{ лк}$$

$$E_{\phi} (436 \text{ лк}) > E_n (300 \text{ лк});$$

Сравнивая полученное значение с нормативным можно сделать вывод о достаточности искусственного освещения в помещении библиотеки.

6.4. Мероприятия по обеспечению благоприятных условий труда

В комплекс необходимых мероприятий по улучшению условий труда входит рациональная организация рабочего места библиотекаря, а также соблюдение Санитарных правил и норм и соблюдение режима труда и отдыха.

Однако необходимо предусмотреть больше перерывов, например, 15 минутные перерывы через каждые 2 часа.

В помещениях с ЭВМ должны быть медицинские аптечки первой помощи. В рассматриваемом центре таковых не имеется. Рекомендуется снабдить отдел аптечками ПМП.

В целях предотвращения пожара необходимо проводить с работниками противопожарный инструктаж, на котором ознакомить их с правилами противопожарной безопасности, а также обучить использованию первичных средств пожаротушения.

Рекомендуется снабдить отдел образцами современной офисной мебели, разработанными ведущими фирмами-производителями.

Соблюдение Санитарных правил и норм, режима труда и отдыха в конструкторском бюро приведет к улучшению условий труда.

6.5. Микроклимат рабочей зоны

Зал библиотеки – это помещение, где выполняются легкие физические работы, поэтому должны соблюдаться следующие требования в соответствии с ГОСТ 12.1.005-88 ССБТ:

- оптимальная температура воздуха – 22⁰С (допустимая - 20-24⁰С) в соответствии с ГОСТ12.1.014-79 и ГОСТ 12.1.005-88,
- оптимальная относительная влажность - 45-60%,
- скорость движения воздуха не более 0.1 м/с.

Для создания и автоматического поддержания в помещении независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Системы вентиляции, отопления и

кондиционирования воздуха выполняются в соответствии с главой СНиП II.33.75.

Таблица 6.1

Сводные значения параметров микроклимата

Параметр	Норма	Фактическое значение
Температура воздуха летом, t °С	18-23	20-22
Температура воздуха зимой, t °С	19-24	19-21
Относительная влажность, %	45-60	50-55
Скорость движения воздуха, м/с	0,1-0,25	0,1-0,2

6.6. Выводы

В помещении библиотеки выполнены все требования к микроклимату, пожаробезопасности и электробезопасности в соответствии с нормативными документами, соблюдаются режимы труда и отдыха.

Все дисплейные терминалы и другое оборудование, применяемое в данном помещении, имеет гигиенический сертификат, соответствует требованиям действующих стандартов, правилам национальных стандартов и имеет в паспорте соответствующую отметку об этом.

Выводы

В последние годы тема построения интегрированных информационных систем управления все больше актуализируется как современными тенденциями в информационно-технологической сфере, так и острой потребностью самих организаций повысить эффективность своей деятельности в части управления внутренними организационными процессами.

В связи с общим стремительным развитием информационных технологий и появлением ряда новых методологических подходов к построению (развертыванию) АИСУ и оценке их эффективности по-иному расцениваются и требования к функциональным возможностям комплексных систем, к интеграции различных компонентов.

В силу того, что традиционно вопросами построения ИС занимаются специалисты в области информационных технологий, до недавнего времени задача воспринималась с точки зрения технической и технологической. С учетом опыта последних лет сегодня с уверенностью можно сказать, что это задача — прежде всего методологическая (концептуальная), где одних знаний в области прикладных информационных технологий недостаточно.

При проектировании и построении АИСУ в организации должна учитываться не только точка зрения пользователя и программиста. По-настоящему эффективной и полезной будет система, в которой реализованы современные методики управления, учитывающие все требования всех пользователей системы.

Поэтому любой современной организация, тем более такой как библиотека, в которой хранятся огромные массивы информации, необходима автоматизированная система, состоящей из базы данных и программы управляющей нею, выполняющей функции хранения, администрирования, управления данными.

В данной работе была спроектирована и разработана в соответствии с современными принципами и технологиями автоматизированная система Библиотеки, которая будет хранить и предоставлять по запросам различную библиотечную информацию, иметь подпрограмму, позволяющую читателю быстро найти нужную ему литературу.

Разработанная система выполняет автоматизированную обработку всей библиотечной информации, упрощает ее поиск и анализ, освобождая сотрудников библиотеки от ряда рутинных работ по подготовке картотек, изданий, списков, а так же для получения читателям рекомендаций по выбору книг.

Программа, управляющая базой данных, имеет понятный простому пользователю интерфейс и работает в диалоговом режиме, используя различные формы, поля и списки.

Список использованной литературы:

1. Черняк А., Метельский Ю. Математическое программирование. Алгоритмические методы. – М.: "Высшая школа", 2007 г. - 352 с.
2. Окулов С. Программирование в алгоритмах (3-е издание). – М.: "Бином. Лаборатория знаний", 2007 г. - 383 с.
3. Князева М. Алгоритмика: от алгоритма к программе. Учебное пособие. - М.: "Кудиц-Образ", 2006 г. - 192 с.
4. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования (2-е издание). – М.: "Инфра-М", 2006 г. - 432 с.
5. Кулаичев А. П. Методы и средства комплексного анализа данных. – М.: "Форум Инфра-М", 2006 г. - 512 с.
6. Левчук Е.А. Технологии организации, хранения и обработки данных. Учебное пособие. – М.: "Вышэйшая школа: Минск", 2005 г. - 239 с.
7. Кузнецов С. Основы баз данных. – М.: "Бином. Лаборатория знаний Интуит", 2007 г. - 484 с.
8. Джон Дей, Крейг Ван Слайк, Рэймонд Фрост. Базы данных. Проектирование и разработка. - М.: "НТ Пресс", 2007 г. - 592 с.
9. Энди Опель. Изучаем SQL. – М.: "НТ Пресс", 2007 г. - 320 с.
10. Полякова Л. Основы SQL (2-е издание). – М.: "Бином. Лаборатория знаний", 2007 г. - 224 с.
11. Голицына О.Л., Партыка Т.Л., Попов И.И. Системы управления базами данных. Учебное пособие. – М.: "Инфра-М", 2006 г. - 432 с.
12. Джен Харрингтон. Проектирование объектно-ориентированных баз данных. – М.: "ДМК Пресс", 2001 г. - 272 с.
13. Кошелев В. Access 2007. Эффективное использование. - М.: "Бином. Лаборатория знаний", 2008 г. - 592 с.
14. Вирджиния Андерсен. Microsoft Office Access 2003. – М.: "АСТ", 2007 г. - 572 с.

15. Кен Блюттман. Access. Трюки. – М.: "Питер", 2006 г. - 332 с.
16. Сапков В. Информационные технологии и компьютеризация делопроизводства (2-е издание). – М.: "Академия", 2007 г. - 288 с.
17. Джим Кью, Марио Джеанини. Объектно-ориентированное программирование. Просто и понятно. – М.: "Питер", 2005 г. - 240 с.
18. Иванова Г., Пугачев Е., Ничушкина Т. Объектно-ориентированное программирование (3-е издание). – М.: "МГТУ им. Н. Э. Баумана", 2007 г. - 368 с.
19. Колесов Ю., Сениченков Ю. Моделирование систем. Объектно-ориентированный подход. Учебное пособие. – М.: "ВНУ", 2006 г. - 192 с.
20. Иван Хладни. Внутренний мир Borland Delphi 2006. – М.: «Вильямс», 2006 г. – 768 с.
21. Хавьер Пашеку. Программирование в Borland Delphi 2006 для профессионалов. – М.: «Вильямс», 2006 г. – 944 с.
22. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем.
23. Калянов Г.Н. CASE: структурный системный анализ (автоматизация и применение). М.: ЛОРИ. 1996.