

Содержание:

ВВЕДЕНИЕ

В современном обществе информационные технологии все активнее и глубже проникают во все сферы человеческой деятельности и становятся их неотъемлемой частью. Информацию рассматривают как один из основных ресурсов развития, а информационные системы и технологии как одно из средств повышения производительности и эффективности работы людей.

Традиционно пользователи привыкли к мысли, что в библиотеке можно получить любую информацию. Теперь, в условиях информационного перенасыщения, библиотеке все сложнее выполнять основные функции: фондообразование, информационно-библиографическое и абонементное обслуживание. Улучшить сбор, хранение и обеспечение доступа к информации библиотека сможет только при условии изменения ее технического оснащения.

На практике это означает выполнение автоматизированной обработки новых поступлений в библиотеку; освобождение сотрудников от ряда рутинных работ по подготовке картотек, изданий, списков, заказов, писем, отчетной документации; создание базы данных о поступлениях; осуществление операций по созданию и копированию тематических архивов литературы.

Благодаря автоматизации можно выполнять предметный поиск информации по запросам читателей, обслуживание баз данных информационных и периодических изданий библиотеки, ведение массива библиографических описаний журнальных статей, поиск записей по ключевым словам, создание электронных справочников, контроль за выданной литературой: учет читателей и их формуляров; автоматическая запись в формуляр читателя выданной литературы; контроль срока возврата книг. А главное - обеспечение читателям своей библиотеки выхода в отечественное и мировое информационное пространство.

Автоматизировав процессы, можно устранить необходимость тратить значительное количество времени, требующегося для проделывания всех ежедневных операций вручную, тем самым, более эффективно использовать труд сотрудников отдела и организации в целом.

В настоящий момент готовые системы на российском рынке, автоматизирующие рабочие места, в том числе и рабочего места сотрудника библиотеки, либо обладают явно излишним функционалом, либо требуют индивидуального подхода, что существенно увеличивает финансовые затраты на внедрение.

В связи с вышеизложенным, было принято решение разработать собственную программу автоматизации рабочего места библиотекаря организации, в частности школьной библиотеки.

Целью настоящего исследования является разработка автоматизированного рабочего места библиотекаря, которая позволит сократить время выполнения функций, устранить случайные ошибки при расчетах, автоматизировать большую часть информационных процессов, и повысить учётно-аналитические возможности.

В рамках дипломного проекта были поставлены и решены следующие задачи:

- изучить и проанализировать соответствующую литературу;
- разработать подробное техническое задание, в котором будет прописан функционал и системные требования к программному продукту;
- выбрать среду программирования путем выявления преимуществ и недостатков;
- спроектировать и разработать программный продукт;
- произвести отладку и тестирование программного продукта.

Объектом исследования является школьная библиотека Муниципального бюджетного образовательного учреждения «Мичуринская средняя общеобразовательная школа» (МБОУ «Мичуринская средняя общеобразовательная школа»), п. Лесные Поляны, пушкинского района.

Предметом исследования является работа школьного библиотекаря.

Разработка базы данных предполагает, что основные операции по накоплению, хранению и переработке информации будут возлагаться на вычислительную технику, специалист же будет выполнять только определенную часть ручных операций и операций, требующих творческого подхода при подготовке управленческих решений.

При подготовке проекта изучена классическая и современная литература в области ИС, включая работы Шрайберга Я.Л., Земскова А.И.

Различные варианты классификации информационных систем у Ю.А.Шафрина Е.В., В.Н. Петрова, И.С. Пилко, Ф.С.Воройского. Некоторые тенденции инновационного развития библиотек и библиографической деятельности в целом, рассмотрены в работах Е.Ю. Качановой, Ю.Н.Столярова.

В дипломном проекте применялись общие методы исследования - системный подход, наблюдение и анализ библиотечно-информационной практики, метод обобщения; классификационная группировка и метод сравнительного анализа.

Практическая значимость проведенного исследования состоит в том, что полученные результаты исследования могут быть полезны для библиотек, формирующие направления использования новых информационных систем.

1. Технико-экономическая характеристика предметной области и предприятия

1.1 Характеристика предприятия и его деятельности

Мичуринская СОШ функционирует с 1964 года. Она осуществляет учебно-воспитательный процесс в соответствии с уровнями общеобразовательных программ 3 ступеней образования: начальное общее образование, основное общее образование, среднее (полное) общее образование. На сегодняшний день в школе обучается 1002 учащихся, работает 76 педагогов. В числе работающих 4 заслуженных учителя Российской Федерации и 5 педагогов имеют нагрудный знак «Отличник народного просвещения», 7 – «Почетный работник общего образования».

Для организации учебно-воспитательного процесса в Мичуринской СОШ имеются 38 учебных кабинета. 3 компьютерных класса, 2 спортивных зала, зал корригирующей гимнастики, бассейн, 2 медицинских кабинета. 3 кабинета для проведения опытно-экспериментальных работ по предметам, мастерские технического и обслуживающего труда, 2 кабинета ОБЖ, психолога, кабинет социального педагога, актовый зал и столовая на 200 посадочных мест, библиотека с читальным залом. Состав библиотечного фонда: учебно-методическая, художественная литература и медиапособия – 11099 экземпляров, словари,

справочники – 424 экземпляра. Имеется доступ в Интернет. С 2004 года действует школьный сайт.

Школа укомплектована 41 классом-комплектом, наполняемость классов составляет не менее 25 человек. Режим работы 2-сменный, 6 дней в неделю, в первых классах – 5 дней. Продолжительность пребывания учащихся в школе 1 смены - с 8.00 до 13.00, 2 смены - с 14.00 до 19.00. В школе 1 ступени после уроков организована группа продленного дня, которую посещают все учащиеся 1-4 классов. Учебная нагрузка учащихся не превышает предельно допустимой нормы.

Продолжительность перемен между уроками 10 - 20 минут. Для учащихся 1-4 классов организовано проведение динамических пауз на свежем воздухе.

Организованным горячим питанием охвачено 100% учащихся. Дополнительным образованием охвачено 750 учащихся.

В 1964-90 учебном году в школе началась работа по обучению первоклассников методом решения учебных задач математике, построенной на основе психологической теории содержательного обобщения. С 1991-92 учебного года технология развивающего обучения (система Д.Б.Эльконина – В.В.Давыдова) используется в преподавании предметов литературы, математики, русского языка, естествознания, ИЗО в начальной школе. На втором этапе инновационной деятельности (сентябрь 1994 – июнь 2003 года) осуществлялась работа по обучению методам решения учебных задач математике и русскому языку младших подростков. В целях обеспечения преемственности в обучении учащихся I и II ступеней Мичуринская СОШ сотрудничает с Открытым институтом развивающего обучения и участвует в разработке содержания математики и предметов естественно-научного цикла основной школы. В ходе инновационной деятельности педколлективом школы создан механизм эффективного функционирования технологии развивающего обучения в школе 1 ступени и создается в школе 2 ступени.

1.2 Организационная структура управления предприятием

Управление школой осуществляется в соответствии с Законом Российской Федерации «Об образовании» и Уставом школы на принципах демократичности, открытости, приоритета общечеловеческих ценностей, охраны жизни и здоровья человека, свободного развития личности.

Система управления МКОУ СОШ осуществляется с учетом социально-экономических, материально-технических и внешних условий в рамках существующего законодательства РФ.

В основу положена пятиуровневая структура управления

Первый уровень структуры – уровень директора (по содержанию – это уровень стратегического управления). Директор школы определяет совместно с Управляющим советом школы стратегию развития школы, представляет её интересы в государственных и общественных инстанциях. Несет персональную юридическую ответственность за организацию жизнедеятельности школы, создает благоприятные условия для развития школы.

На втором уровне структуры (по содержанию – это тоже уровень стратегического управления) функционируют как традиционные субъекты управления: педагогический совет, общественные учительские организации, совет школы, родительский комитет.

Высшим коллективным органом управления школой является Общее собрание собрания коллектива школы, включающее представителей педагогического коллектива, родителей, общественности и учащихся, которое решает вопросы организации внешкольной и внеклассной работы, детского питания, развития материальной базы школы, принимает участие в развитии учебного заведения, регулирует трудовые, социально-экономические и профессиональные отношения между руководителем и работником.

Педагогический совет – коллективный орган управления школой, который решает вопросы, связанные с реализацией программы развития, рассматривает проблемы, подготовленные научно-методическим советом, администрацией школы, несет коллективную ответственность за принятые решения.

Членами Педагогического Совета являются все учителя и воспитатели школы, включая совместителей. Председателем Педагогического Совета является директор школы. Он назначает своим приказом секретаря Педагогического Совета сроком на один год.

Педагогический Совет собирается не реже четырех раз в году. Ход педагогических советов и решения оформляются протоколами. Протоколы хранятся в школе постоянно.

Педагогический Совет определяет:

а) порядок проведения промежуточных аттестаций для учащихся не выпускных классов;

б) порядок приема в школу;

в) условный перевод обучающихся, имеющих академическую задолженность по одному предмету, в следующий класс;

г) оставление на повторный год обучения, перевод в общеобразовательный класс при наличии более трех-четырёх удовлетворительных оценок по предметам, или перевод на семейное образование (по усмотрению родителей) учащихся, имеющих академическую задолженность по двум или более предметам по результатам учебного года;

д) перевод в следующий класс обучающихся, освоивших курс в полном объеме.

Третий уровень структуры управления (по содержанию – это уровень тактического управления) – уровень заместителей директора. Этот уровень представлен методическим советом.

Методический совет – коллегиальный совещательный орган, в состав которого входят руководители ШМО, творческих групп. Возглавляет МС заместитель директора по учебно-воспитательной работе. МС руководит работой творческих групп учителей, методическими объединениями, инновационной деятельностью коллектива. МС подотчетен педагогическому совету, несет ответственность за принятые решения и обеспечивает их реализацию.

Заместитель директора по учебно-воспитательной работе осуществляет управление функционированием школы: контролируют выполнение государственных стандартов образования, отслеживает уровень сформированности общеучебных умений и навыков, необходимых для продолжения образования.

Заместитель директора по воспитательной работе организует внеурочную воспитательную работу с детьми, работу органов ученического самоуправления. Контролируют состояние воспитательной работы в школе, отслеживают уровень воспитанности учащихся, работает с детьми, требующими особого педагогического внимания, отвечает за связь с внешкольными учреждениями.

Четвертый уровень организационной структуры управления – уровень учителей, функциональных служб (по содержанию – это уровень оперативного управления), структурных подразделений школы.

Методические объединения – структурные подразделения методической службы школы, объединяют учителей одной образовательной области. Руководитель ШМО выбирается из состава членов ШМО и утверждается директором школы. МО ведет методическую работу по предмету, организует внеклассную деятельность учащихся, проводит анализ результатов образовательного процесса. ШМО имеет право выдвигать предложения по улучшению процесса образования, получать методическую помощь научных консультантов, согласует свою деятельность с методическим советом школы и в своей работе подотчетно ему.

Творческая группа учителей – временная форма педагогического коллектива, работающего в режиме развития. Создается для решения определенной учебной или воспитательной проблемы, может объединять учителей одного или различных предметов. В группе выбирается руководитель, организующий разработку данной проблемы. По итогам работы готовятся рекомендации по использованию созданного опыта. Подотчетна МС.

Медицинское обслуживание обеспечивается медицинской сестрой, специально закрепленной органом здравоохранения за школой. Медицинская служба осуществляет постоянное наблюдение за состоянием здоровья и физическим развитием учащихся. Совместно с педагогическим коллективом несет ответственность за проведение лечебно-профилактических мероприятий, соблюдение санитарно-гигиенических норм, режима и качества питания обучающихся.

Пятый уровень организационной структуры – уровень учащихся. По содержанию – это тоже уровень оперативного управления, но из-за особой специфичности субъектов, этот уровень скорее можно назвать уровнем соуправления. Иерархические связи по отношению к субъектам пятого уровня предполагают курирование, помощь, педагогическое руководство как создание условий для превращения ученика в субъект управления. Школьный ученический совет самоуправления – осуществляет деятельность по всем направлениям воспитательной работы в школе, помогает в проведении всех внеклассных и общественных мероприятий, осуществляет шефство над начальной школой, способствует организации учебно-воспитательного процесса.

В школе действует общешкольный родительские комитет. Он содействует объединению усилий семьи и школы в деле обучения и воспитания детей. Оказывают помощь в определении и защите социально не защищенных учащихся.

Избранные представители классных родительских комитетов составляют общешкольный родительский комитет, который избирает председателя комитета, секретаря.

Родительский комитет школы:

- обсуждает кандидатуры и утверждает списки учащихся, которым необходимо оказать материальную помощь в любой форме;
- принимает решение по вопросу охраны Учреждения и другим вопросам жизни Учреждения, которые не оговорены и не регламентированы Уставом Учреждения.

Общее руководство Учреждением осуществляет выборный орган – Управляющий совет Учреждения (далее – Совет), который создается с использованием процедур выборов, назначения и кооптации.

Полномочия Совета:

- разрабатывает и утверждает совместно с директором Учреждения концепцию развития Учреждения, разработанные долгосрочные образовательные программы;
- согласовывает режим занятий обучающихся по представлению педагогического совета, в том числе продолжительность учебной недели (пятидневная или шестидневная), время начала и окончания занятий;
- принимает решения по другим важнейшим вопросам деятельности Учреждения, не отнесенным к компетенции директора, иных органов управления Учреждения;
- согласовывает компонент учебного плана Учреждения государственного образовательного стандарта общего образования, профили обучения и вносит предложения администрации Учреждения;
- согласовывает решение о введении (отмене) единой в период занятий формы одежды для обучающихся;
- рассматривает жалобы и заявления обучающихся, родителей (законных представителей) на действия (бездействие) педагогического и административного персонала Учреждения;

- содействует привлечению внебюджетных средств для обеспечения деятельности и развития Учреждения;
- заслушивает отчет руководителя Учреждения по итогам учебного и финансового года;
- рассматривает вопросы создания здоровых и безопасных условий обучения и воспитания в Учреждении.



Рис 1. Структура управления школой

1.3 Библиотека как объект автоматизации. Организация и ведение учета документов в библиотеке

Библиотека представляет собой учреждение, содержащее интеллектуальные ценности общества и обеспечивающая использование этих ценностей для целей образования, научных исследований и культурного развития личности. Всевозрастающие информационные потоки выражались в сложной и трудоемкой их ручной обработке. Данный процесс способствовал появлению вычислительной технике в библиотеках.

Учет библиотечного фонда является частью системы управления экономическими процессами; это важнейший элемент сохранности библиотечного фонда. Учетные документы, в которых регистрируются прием, выбытие и движение фонда являются юридическими документами.

Учет библиотечного фонда – это и технологический процесс организации их. Основная цель учета – формирование точных и достоверных данных обо всех видах документов, обеспечение контроля за их движением и сохранностью в соответствии с действующими нормативными документами [21, с. 48].

Школьная библиотека является структурным подразделением образовательного учреждения, участвующим в учебно-воспитательном процессе в целях обеспечения права участников образовательного процесса на бесплатное пользование библиотечно-информационными ресурсами. Руководствуется в своей деятельности федеральными законами, указами и распоряжениями Президента РФ, постановлениями и распоряжениями Правительства РФ и исполнительных органов субъектов РФ, решениями соответствующего органа управления образованием, уставом общеобразовательного учреждения, положением о библиотеке.

Деятельность библиотеки осуществляется на основе библиотечно-информационных ресурсов в соответствии с учебным и воспитательным планами образовательного учреждения, программами, проектами и планом работы библиотеки.

Повседневно взаимодействуя с учащимися, административно-педагогическим коллективом, родителями, библиотека выполняет следующие функции:

образовательную – поддерживает и обеспечивает образовательные цели, сформулированные в концепции школы, осуществляет свою деятельность в соответствии с основными направлениями развития образования в школе;

информационную – предоставляет возможность использования информации вне зависимости от ее вида, формата, носителя;

культурную – организывает мероприятия, формирующие культурное и социальное самосознание, содействующие эмоциональному развитию учащихся.

Цели библиотеки соотносятся с целями общеобразовательного учреждения: формирование общей культуры личности обучающихся на основе усвоения обязательного минимума содержания общеобразовательных программ, их

адаптация к жизни в обществе, создание основы для осознанного выбора и последующего освоения профессиональных образовательных программ, воспитание гражданственности, трудолюбия, уважения к правам и свободам человека, любви к окружающей природе, Родине, семье, формирование здорового образа жизни.

Основные задачи библиотеки

- Обеспечение участникам учебно-воспитательного процесса доступа к информации, знаниям, идеям, культурным ценностям посредством использования библиотечно-информационных ресурсов на различных носителях.
- Формирование у читателей навыков независимого библиотечного пользователя.
- Воспитание культурного и гражданского самосознания, помощь в социализации обучающегося, развитии его творческого потенциала.
- Совершенствование и освоение новых библиотечных технологий.
- Расширение ассортимента библиотечно-информационных услуг.
- Выявление информационных потребностей и удовлетворение образовательных и индивидуальных потребностей пользователей библиотеки.
- • Формирование комфортной библиотечной среды.

Основные направления работы

- Планирование и отчетность. Выполнение средних показателей.
- Организация книжного фонда. Учет. Сохранность.
- Комплектование.
- Справочно-библиографическая работа.
- Библиотечно-информационное обслуживание. Массовая работа.
- Повышение квалификации. Самообразование. Участие в профессиональных мероприятиях. Взаимодействие с другими библиотеками, организациями.

Основные контрольные показатели работы:

- Книжный фонд, учебный фонд (количественный и качественный состав).
- Массовая работа (количество и охват).
- Библиотечные уроки (количество и охват).
- Библиотечное обслуживание (процент охвата).
- Книговыдача.

- Читаемость.
- Посещаемость.
- Обращаемость фонда.
- Книгообеспеченность.

1.4 Необходимость автоматизации рабочего места сотрудника библиотеки

Современная библиотека напрямую связана с развитием информационной техники; информационные технологии позволяют быть ей современной, актуальной и оперативной в предоставлении информации. Информационные технологии в библиотечном деле базируются на основе аппаратных средств и программных продуктов.

Современное состояние образования предъявляет новые требования к функционированию школьной библиотеки. В настоящее время она рассматривается как открытая система, обменивающаяся информацией с окружающей средой. Изменение информационных потребностей пользователей, внедрение автоматизированных технологий кардинально меняют политику, направления и возможности библиотечного обслуживания.

Внедрение компьютерной техники, позволившие переводить традиционные картотеки в электронную форму и создавать на их основе проблемно-ориентированные базы данных, работы по созданию ЭК, последовательное и целенаправленное приобретение баз данных на CD-ROM, а также первый опыт работы в международных информационных сетях через Интернет требуют поиска новых форм организации информационно-библиографического обслуживания в библиотеке.

На практике это означает выполнение автоматизированной обработки новых поступлений в библиотеку; освобождение сотрудников от ряда рутинных работ по подготовке картотек, изданий, списков, заказов, писем, отчетной документации; создание базы данных о поступлениях; осуществление операций по созданию и копированию тематических архивов литературы.

Благодаря автоматизации, можно выполнять предметный поиск информации по запросам читателей, обслуживание баз данных информационных и периодических

изданий библиотеки, ведение массива библиографических описаний журнальных статей, поиск записей по ключевым словам, создание электронных справочников, контроль за выданной литературой: учет читателей и их формуляров; автоматическая запись в формуляр читателя выданной литературы; контроль срока возврата книг. А главное - обеспечение читателям своей библиотеки выхода в отечественное и мировое информационное пространство [20, с. 138].

Какие же функции библиотеки целесообразно автоматизировать? Программное обеспечение в первую очередь должно реализовать следующие функции:

- обработку, хранение библиографической и фактографической информации, ее поиск по любым элементам записей и их сочетаниям;
- поддержку иерархических классификаций;
- подготовку материалов для библиографических изданий, подборок материалов в виде списков, фактографических и библиографических записей, отсортированным по индексам какой-либо классификации и алфавиту;
- вывод данных о занятости экземпляра издания, осуществление заказа единицы хранения читателем непосредственно при работе с ЭК с автоматическим формированием читательского требования;
- фиксацию выдачи и возврата единиц хранения с параллельным изменением данных об их местонахождении и сведений в читательском формуляре о полученных документах;
- поддержку использования при поиске нормативных записей об индивидуальных авторах и коллективах;
- наличие справочных текстов для ситуаций, которые могут вызвать затруднения у пользователя;
- ввод записей о заказываемых документах и учет поступления их в библиотеку;
- выдачу результатов поиска на экран и на принтер в принятой форме;
- загрузку данных из текстовых файлов и выгрузку записей из базы данных в текстовые файлы;
- защиту базы данных от несанкционированного доступа;
- восстановление базы данных в случае аварийной ситуации.

Современные масштабы и темпы внедрения средств автоматизации управления с особой остротой ставит задачу проведения комплексных исследований, связанных со всесторонним изучением и обобщением возникающих при этом проблем как практического, так и теоретического характера.

1.4 Анализ современных средств автоматизации библиотечных ИС

1.4.1 Общий обзор программных комплексов

Автоматизированные библиотечные информационные системы (АБИС) - системы планирования ресурсов для библиотеки, которые используются для отслеживания библиотечных фондов, от их заказа и приобретения до выдачи посетителям библиотек.

Для перехода библиотек на новый уровень необходимо осуществить комплекс мер по внедрению аппаратно-технического и программного обеспечения. Причем выбор и внедрение программного обеспечения являются наиболее важной составляющей процесса автоматизации библиотек и их объединения, так как именно от нее зависит эффективность вложения как начальных средств, так и последующих, еще более значительных, затрат на создание электронных информационных массивов [5,с.173].

Основным программным обеспечением для библиотек являются автоматизированные информационно-библиотечные системы (АИБС). Применение библиотечных форматов увеличивает скорость обслуживания, предотвращает дублирование работы и помогает библиотекам лучше распорядиться библиотечными ресурсами. Средства каталогизации позволяют обрабатывать и описывать любые виды изданий, включая нетрадиционные, такие, как аудио и видеоматериалы, компьютерные файлы и программы, картографические материалы, ноты и т.д. Пользователю предлагается список готовых сценариев ввода, т.е. входных экранных форм для создания различных типов библиографических описаний. При этом у него есть возможность менять их в самых широких пределах в соответствии с собственными требованиями.

На рынке программного обеспечения существует несколько информационных систем, обеспечивающих комплексную автоматизацию в библиотеках. Рассмотрим основные из них.

АИБС «МАРК-SQL» является системой третьего поколения, разрабатывалась на основе принципов открытых систем и обеспечивает комплексную автоматизацию всех библиотечных процессов: комплектование литературы, создание и ведение электронного каталога, систематизацию, обработку поступающих изданий,

справочно-информационное обслуживание, обслуживание читателей, учет библиотечных фондов. С помощью этой программы собственные информационные ресурсы библиотек могут быть интегрированы в мировую информационную среду.

АИБС «МАРК-SQL» состоит из пяти АРМов: «Администратор», «Каталогизатор», «Комплектование», «Поиск» и «Абонемент».

Для описания документов в АИБС «МАРК-SQL» используется USMARC-формат [5,с.19].

С помощью различных настроек в системе обеспечивается описание различных типов документов. Во всех АРМах предоставляется возможность различного представления документов на экране. Имеется возможность создавать собственные экранные формы представления документов.

В системе поддерживается штрихкодовая технология при создании библиографической записи для идентификации экземпляра, ведении базы данных читателей, при регистрации читателей, при операциях выдачи/возврата книг

Существуют и другие библиотечные программы. Например, система автоматизации библиотек ИРБИС представляет собой типовое интегрированное решение в области автоматизации традиционных библиотечных технологий и предназначена для использования в библиотеках любого уровня и профиля. В настоящее время ИРБИС применяется в сотнях библиотек самого разного масштаба и разного профиля.

Отличительной особенностью ИРБИС является то, что она ориентирована на использование в библиотеках, не располагающих штатом специалистов высокого уровня ни в области библиотечного дела, ни в области вычислительной техники. Система располагает широким набором инструментальных средств, которые позволяют подготовленному пользователю перестраивать ее практически в неограниченных пределах: начиная с изменения входных экранных форм, структуры данных и форматов их представления и кончая созданием собственных оригинальных модулей и приложений. Система ориентирована на работу в локальных вычислительных сетях любого типа без ограничения количества пользователей при условии, что клиентской платформой является Windows.

В состав системы входят модули поддержки Интернет-технологий, а именно: Веб-ИРБИС – решение, обеспечивающее доступ к базам данных ИРБИС через WWW-сервер. Система полностью совместима с международными библиографическими форматами UNIMARC и USMARC на основе средств двухсторонней конверсии

данных. ИРБИС также поддерживает Российский коммуникативный формат RUSMARC [26,с.20].

В системе реализованы все основные библиотечные технологии, включая технологии комплектования, систематизации, каталогизации, читательского поиска, книговыдачи и администрирования, на основе взаимосвязанного функционирования пяти типов автоматизированных рабочих мест: «Комплектатор», «Каталогизатор», «Читатель», «Книговыдача», «Администратор».

ИРБИС позволяет создавать и поддерживать любое количество баз данных, составляющих электронный каталог или представляющих собой проблемно-ориентированные библиографические базы данных. Система предлагает технологию автоматического формирования словарей, на основе которых реализуется быстрый поиск по любым элементам описания и их сочетаниям. Предлагаются средства, позволяющие пользователю формировать список видов поиска по собственному усмотрению.

ИРБИС поддерживает традиционные «бумажные» технологии, предоставляя возможности для получения широкого спектра выходных форм: начиная с каталожных карточек и кончая оригинал-макетами таблиц и указателей [5, с.146]. ИРБИС обладает широкими возможностями для адаптации к условиям работы конкретной библиотеки, т.е. при передаче и установке системы может производиться ее настройка в соответствии со специфическими требованиями пользователя.

АБИС «Руслан» создана в Центре «Открытые библиотечные системы» Санкт-Петербургского государственного технического университета. Система разработана с использованием самых современных технологий построения распределенных информационных систем, открытых стандартов на протоколы взаимодействия и форматы данных.

Система является модульной, масштабируется в зависимости от потребностей организации. В процессе эксплуатации системы можно постепенно, без замены компонентов, наращивать число узлов в сети, создавать новые центры, практически не ограничено увеличивать объемы каталогов и количество пользователей и организаций. и пр.) Система поддерживает обработку библиографических и авторитетных записей в формате RUSMARC со всеми возможностями и особенностями формата, заложенными его разработчиками, включая механизмы связывания записей, автоматической трассировки ссылок,

подключения авторитетных файлов, настраиваемые таблицы значений индикаторов и подполей, контроль структурной и логической целостности записи и многое другое. Это позволяет качественно создавать собственные записи и эффективно использовать базы данных библиографических и авторитетных записей ведущих российских библиографирующих агентств.

«Руслан» - российская библиотечная система, учитывающая специфику технологий российских библиотек, но вместе с тем, отвечающая мировым требованиям и спецификациям. В системе поддерживаются и постоянно развиваются новейшие библиотечные технологии, внедряемые в ведущих мировых библиотечных организациях (корпоративная каталогизация, ведение каталогов электронных ресурсов, открытый доступ к электронным каталогам и полнотекстовым коллекциям, штрих-кодирование единиц хранения и читателей, электронный заказ и доставка документов.

Автоматизированная информационная библиотечная система «Библиотека 2000» подходит для использования в библиотеках любого типа. Основные характеристики системы: создание и ведение электронных каталогов; ввод, хранение, коррекция информации в электронных каталогах; поиск по любому критерию. Система обеспечивает поиск данных в электронном каталоге по любому набору поисковых элементов. Кроме простого поискового интерфейса, обеспечена возможность составления сложных поисковых запросов. Автоматическое создание полного комплекта каталожных карточек. В комплект поставки включены типовые документы.

Поддержка электронного каталога авторитетных записей. Проверка на дублирование. Доступ к каталогам через Internet/Intranet. Открытая система настройки Система не имеет деления на отдельные АРМы. В стандартный комплект поставки включены все основные функции, обеспечивающие реализацию технологии работы библиотек: администрирование, каталогизация, комплектование, книговыдача, обслуживание читателей. Работа организуется путем разграничения прав доступа.

Все эти программы имеют полную совместимость по данным. Программы нашли широкое применение при создании электронных коллекций.

1.4.2 Применение средств сканирования в автоматизации библиотечного документооборота

Современные технологии с применением сканирования разделяются на две группы: технологии штрих-кодирования и технологии радиочастотной идентификации (RFID) контроля перемещения документов в библиотеке.

Технология штрихового кодирования состоит из трех этапов: создание штрихового кода, нанесение его на этикетку, считывание штрихового кода специальной аппаратурой.

Штрих-код представляет собой закодированную определенным образом последовательность из цифр и букв (может быть что-либо одно). Разработка штрих-кода происходит с помощью специальных программ. Создается своя уникальная последовательность, которая является своеобразным паспортом помеченного документа. Чаще всего символ штрихового кода отображается в виде светлого прямоугольника, внутри которого расположены темные штрихи.

Нанесение штрихового кода на этикетки может быть реализовано в библиотеке с помощью как самых обычных, так и специализированных термо- и термотрансферных принтеров. На этикетке может быть не только штриховой код, но и самая разнообразная информация (например, название библиотеки, ее отдела, к которому приписан документ). Организовать печать можно непосредственно в том месте, где будут наклеиваться этикетки: например в отделе обработки.

Чтобы просто считать штрих-код, необходим сканер штрих-кодов, подключенный к компьютеру и любая программа, принимающая и отображающая ввод с клавиатуры. Сканер штрих-кода освещает штрих-код, принимает отраженный световой сигнал, преобразует его в последовательность цифр, и посылает цифровые символы в программу. Таким образом, осуществляется автоматизированный ввод идентификатора предмета. Сам штрих-код вместе с описанием вносится в базу данных. Далее можно совершать необходимые действия: фиксировать операции с книгами, оборот книг в библиотеке и т.д.

В современных АИБС информация электронного каталога технологически связана с регистрационными базами данных читателя и позволяет «привязать» выдаваемые и возвращаемые книги к данному читателю, контролируя период отсутствия документа в фонде. Все это играет немаловажную роль в учете, а следовательно, и в сохранности библиотечных фондов. У библиотекаря же появится возможность автоматизировать ежедневные многократные процессы регистрации выдачи и возврата книг.

RFID метка для библиотек специально разработана для маркировки объектов фондов хранения, включая книги, CD/DVD диски и видеокассеты.

Библиотечные RFID метки имеют клеящуюся поверхность, легко наносятся на предметы. Метки могут иметь различную форму: квадратные, прямоугольные и круглые (для книг, журналов, CD дисков, кассет). Библиотечные RFID метки наклеены на каждую единицу библиотечного фонда (книгу, журнал, брошюру), покрыты защитной бумажной этикеткой (белой или с нанесенным логотипом штрих - кодом) (рисунок 1.1.).



Рис 1.1. Библиотечная RFID метка

Использование RFID упрощает обработку данных, существенно, облегчая ежедневный труд библиотечных работников, и позволяет организовать станции самообслуживания, удобные для читателей (выдача и возврат материалов). Объединяя идентификационную и противокражную функции, библиотечные RFID метки, уменьшает затраты времени на каждом этапе процесса, от создания каталога до операций выдачи и возврата материалов.

Обеспечивая возможность выдавать и принимать несколько книг одновременно, RFID технология избавляет библиотечных работников от многократно повторяемых действий, что значительно ускоряет процесс обслуживания читателей.

Сочетание с имеющимися противокражными системами дает возможность постепенного перехода на законченное RFID решение (идентификация + защита).

С использованием технологии RFID проводить инвентаризации становится существенно легче, для этого не обязательно закрывать библиотеку, поскольку теперь инвентаризация займет примерно в 20 раз меньше времени, чем с использованием технологии штрих-кода.

Сотруднику библиотеки достаточно пройти вдоль полок со специальным ридером для инвентаризации, чтобы собрать информацию с меток.

Затем собранные данные автоматически загружаются в базу данных библиотеки. Существует версия оборудования с технологией Wi-Fi, в этом случае считанные

данные передаются непосредственно в РС или на карманный компьютер.

Ридер для инвентаризации позволяет библиотечным работникам легко и быстро идентифицировать книги на полках.

Ридер подсоединяется к компьютеру, который сохраняет и выводит на дисплей данные, относящиеся к считываемому предмету.

Далее информация передается в базу данных библиотеки через специальное устройство или бесконтактным способом. В настоящее время доступна версия оборудования с технологией Wi-Fi, что позволяет передавать данные с меток в РС в процессе проведения инвентаризации.

Нет необходимости идентифицировать книги одну за другой, это можно делать одновременно, кроме того, не нужно брать их с полки. Благодаря высокой скорости считывания и возможности долгой работы в автономном режиме требуется значительно меньше времени и человеческих ресурсов для проведения инвентаризации. Поскольку процесс инвентаризации значительно упрощен, его можно проводить чаще.

Технология RFID имеет преимущества перед штриховым кодированием, например, с ее помощью можно существенно сократить время поиска заставленных книг. Для того чтобы найти нужную книгу, необходимо пройти вдоль стеллажей с ручным считывателем, и как только он «увидит» ее, вы услышите звуковой сигнал. С помощью того же ручного считывателя можно проводить инвентаризацию, и она будет занимать у вас в 20 раз меньше времени, чем с штрих-кодом. Считывателем проводится вдоль стеллажей, и вся информация о книгах, присутствующих на них, дистанционно передается в компьютер.

При внедрении данной системы происходит увеличение и упрощение скорости обслуживания читателей, точное планирование закупок книг для пополнения библиотечного фонда, предоставляет широкие возможности по интеграции с существующими АБИС.

Прежде всего, организация должна определить, а что же собственно ожидается от новой системы: какие функциональные области она должна охватывать, какую техническую платформу использовать, какие отчеты готовить?

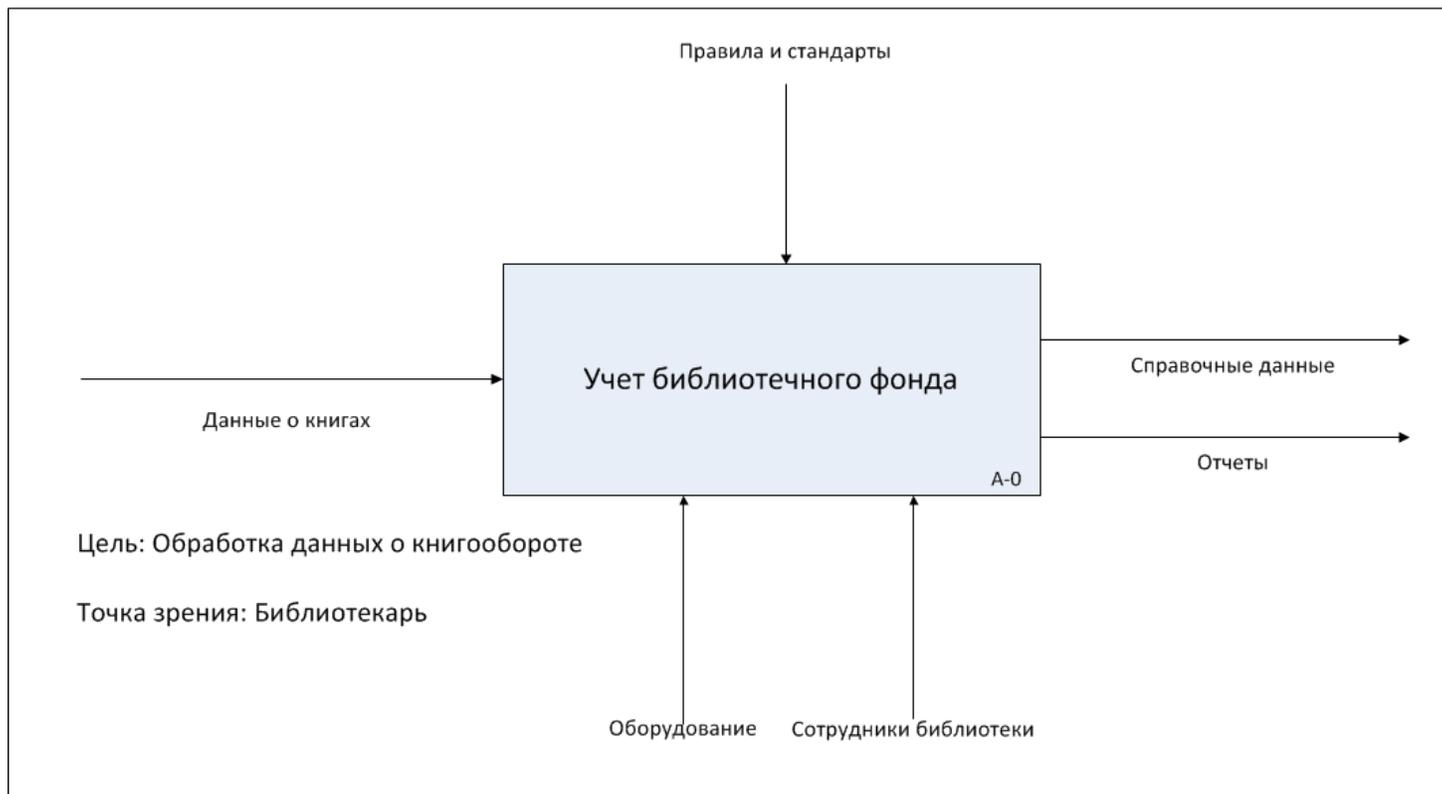
Любая из систем – лишь механизм для повышения эффективности управления, принятия правильных стратегических и тактических решений на основе

своевременной и достоверной информации, выдаваемой компьютером.

Так что оптимальным решением будет разработка собственной программы рабочего места сотрудника библиотеки.

2. Информационное обеспечение задачи

Информационная модель и её описание



Главной целью деятельности библиотеки является предоставление преподавателям и учащимся доступ к литературе разных видов и жанров.

Рис 2.1 Деятельность библиотеки

2.1 Основные положения учета библиотечного фонда

Учет библиотечного фонда отражает поступление документов в фонд, выбытие из фонда, величину всего библиотечного фонда и служит для государственного статистического учета, отчетности библиотеки, планирования ее деятельности,

обеспечение сохранности фондов, контроля за наличием и движением документов.

К ведению учета библиотечного фонда предъявляются следующие требования:

1. полнота и достоверность учетной информации;
2. документированное оформление каждого поступления в фонд и выбытия из фонда;
3. совместимость приемов и форм учета;
4. соответствие показателей учета библиотечного фонда аналогичным показателям государственной библиотеки [19, с. 15].

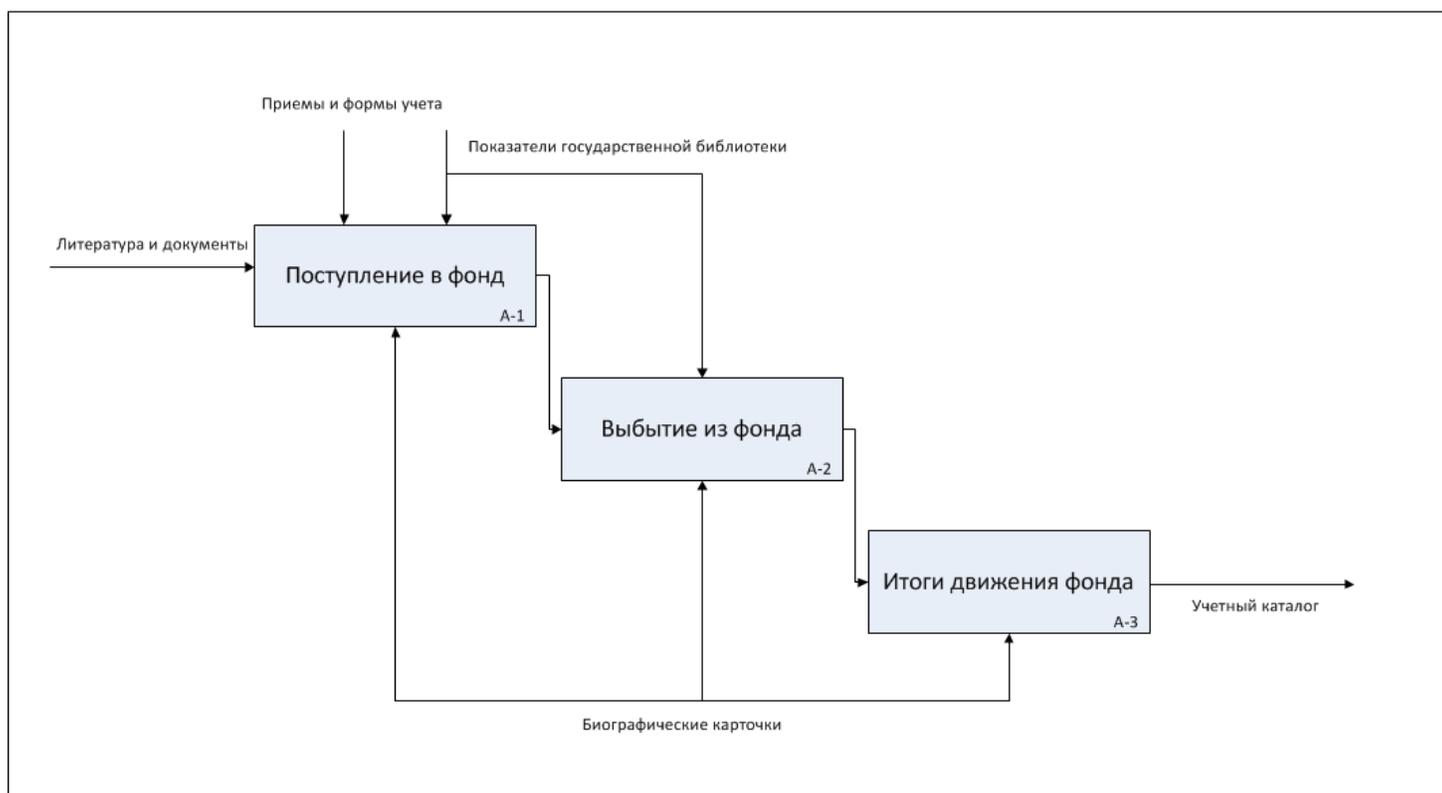


Рис 2.2 Учет библиотечного фонда

Основными функциями учета являются информативная, управленческая и контрольная. В каждом виде учета в соответствии с его назначением приоритетной становится одна из функций. Так, с определенной долей условности информативная и управленческая функции осуществляются с помощью суммарного учета, контрольная – индивидуального и стоимостного. В условиях рыночных отношений возрастает роль стоимостного учета, стало быть, усиливаются контрольные функции учета.

Учет библиотечного фонда, то есть процесс регистрации документов, прибывающих в фонд или исключаемых из него, ведется в двух формах: статистической и бухгалтерской.

Статистический учет — это составление количественной характеристики объема, состава и движения (постоянного изменения) библиотечного фонда в процессе его формирования и функционирования. Он проводится библиотекой по особым учетным формам, которые предусматривают возможность расширения сведений о фонде, чтобы можно было использовать дополнительные данные для его изучения.

Сравнение и анализ дополнительных количественных сведений учета библиотечного фонда с другими показателями работы позволяют сделать выводы о качестве и перспективах деятельности библиотеки в целом. Учет дает возможность установить такие показатели работы, как книгообеспеченность, обращаемость, читаемость и всего фонда, и его частей [18, с. 53].

Правильная организация учета способствует также охране библиотечного фонда — государственной и общественной собственности, имеющей огромную культурную и материальную ценность.

Бухгалтерский учет — это финансовый контроль бухгалтерии библиотеки или учреждения над стоимостным выражением БФ, то есть за расходованием средств на комплектование, организацию и использование фонда. В стоимость изданий и материалов не включаются накладные расходы (на библиотечную обработку, почтовые и транспортные).

Существует два вида учета: суммарный и индивидуальный.

Суммарный учет дает полную количественную характеристику фонда. Суммарный учет — это учет документов партиями, которые поступают или выбывают по одному сопроводительному документу (счет, накладная, акт и т.д.)

В случае отсутствия сопроводительного документа составляется акт приема.

Формой суммарного учета является «Книга суммарного учета библиотечного фонда».

Она состоит из трех частей:

1. Часть 1 — «поступление в фонд». — В нем ведется запись: дата и номер записи, источник поступления, номер сопроводительного документа, количество

поступивших документов (по видам и содержанию), стоимость приобретенных документов. Нумерация записей о поступлении ежегодно начинается с № 1, о выбытии — также с № 1 и идет в возрастающем порядке из года в год вне зависимости от причин выбытия. Литература, не подлежащая записи в ИК (временного хранения), записывается в части I книги суммарного учета без указания ее стоимости и распределяется по содержанию. Не подлежат учету и не включаются в фонд материалы служебного пользования (программные продукты, являющиеся рабочим инструментом персонала библиотеки и программистов материалы, приобретаемые для оформления библиотеки, других подсобных работ, не связанных с комплектованием библиотечного фонда).

2. Часть 2 – «выбытие из фонда» - здесь собираются все сведения о выбывших из библиотеки документах. Они выбывают по актам с указанием причин выбытия: ветхость, утеря читателями, морально устаревшие и т.д.
3. Часть 3 – «итоги движения фонда». Эта графа заполняется в конце года, когда приняты все партии документов, проведены все акты на списание.

Поступление CD-ROM, мультимедиа и т.д. отражаются в первой части книги суммарного учета, выбытие – во второй, итоги движения – в третьей части.

Все периодические издания подлежат суммарному учету.

Индивидуальный учет – учет каждого экземпляра документа, поступившего или выбывшего из фонда. Учету подлежат все виды документов на любых носителях информации. Индивидуальный учет – это регистрация отдельных изданий (по названиям) и экземпляров документов для сохранности БФ. Для того чтобы выполнить все ее требования, следует прежде всего определить единицу учета разных видов документов, одинаковую для всех библиотек. Понятие «единица учета» заложена в ГОСТ 7.20—2000 и «Инструкции об учете БФ» (2000 год). Это книги, брошюры ноты, карты, изоиздания (отдельный лист, тетрадь, атлас, карта, альбом), научно-технические документы (НТД) (стандарты, технические условия (ТУ), описания изобретений к авторским свидетельствам) — экземпляр; диафильмы — катушка; диапозитивы (слайды) — комплект; грампластинка — диск; магнитные фонограммы — катушки, бобина, кассета, рулон; фильмокопии — фильм. Но если последние являются приложением к производству печати, то индивидуальному учету не подлежат. Учет CD-ROM, мультимедиа, дискет ведется в любой форме индивидуального учета.

Индивидуальный учет по названиям осуществляется с целью представление о содержательном потенциале библиотечного фонда, количестве названий документов. Он позволяет определить, насколько многоаспектно отражены в фонде те или иные темы, вопросы, проблемы, то есть выявить многообразие его наполнения информацией. Существует несколько форм индивидуального учета: учет часто называют библиографическим он осуществляется в виде каталогов (алфавитный каталог (АК), систематический каталог (СК), нумерационный; для учета документов — со сквозной нумерацией: ГОСТы, описания изобретений, ценники, прейскурант и др.). В них на каждое название заводится отдельная карточка. На серийные, продолжающиеся издания, бюллетени ведутся учетно-регистрационные картотеки и каталоги [14, с. 50].

Учетный каталог совмещает учет библиотечного фонда по названиям и по экземплярности. На каждое название заполняется карточка с полным библиографическим описанием и указанием цены. Карточке и документу присваивается очередной порядковый номер. Издание объемом до 48 страниц регистрируется на карточке, но не нумеруется. На обороте карточки фиксируется распределение экземпляров документа по подразделениям фонда библиотеки. Дается один общий инвентарный номер для всех экземпляров документ» и всех типов многотомного издания [14, с. 53].

Карточки расставляются в алфавитном порядке. Каждая карточка учетного каталога и общая опись инвентарных номеров регистрируются в специальных журналах. Затем нумератором на обороте титульного листа книги, на книжных формулярах и на карточке учетного каталога проставляется инвентарный номер.

На каждый вид издания существует своя форма карточки. Учетные каталоги хранятся в запирающихся шкафах или служебном помещении, которое опечатывается. Карточки в ящиках закреплены стержнями, доступ к каталогам имеет ограниченный круг сотрудников.

Несколько по-иному ведется *учет периодики*. Текущие поступления журналов и газет регистрируются в учетной картотеке. В конце года подсчитывается общее количество поступивших номеров журналов за год и составляется акт, на основании которого производится запись в книге суммарного учета. Акты эти в бухгалтерию не сдаются, так как там есть подписные квитанции, подтверждающие расходование средств на подписку. Газеты регистрируются в картотеке, хранению подлежат лишь местные газеты, центральные — исходя из наличия площади.

Учет библиотечного фонда учебников существенно отличается от учета другой литературы. Особые затруднения вызывает учет книг для чтения, словарей, сборников задач, дидактических материалов. К школьным учебникам относят орфографические словари, математические таблицы, сборники задач и упражнений, практикумы. Учет экспериментальных и пробных учебников осуществляется в книге суммарного учета и учетных каталогах [21, с. 64].

Регистрация многоэкземплярных (дублетных) изданий проводится либо по способу кавычек (вместо повторной записи названия проставляются кавычки), либо по способу крайних номеров (на одной строке записываются через тире первый и последний по числу дублетов номера). Номера оставляются на одной строке перед названием документа, всего на страницу заносится 20 записей. В пределах этой нумерации каждому экземпляру присваивается индивидуальный номер. Такой способ экономичен и широко применяется для регистрации учебной и многоэкземплярной литературы. Инвентарный номер является сугубо индивидуальным и никогда никаким другим документам не присваивается даже в случае, когда все экземпляры выбывают из библиотеки.

В последние годы функцию инвентарного номера с успехом выполняет штриховой бар-код, применяемый при автоматизированной обработке и выдаче документа.

Таким образом, ведение учета библиотечного фонда обусловлено рядом обязательных требований. Он должен быть полным и достоверным и отвечать показателям государственной статистики. Полнота и достоверность учета обеспечиваются документальным оформлением каждого поступления в фонд и каждого выбытия из него.

Прием новых поступлений.

Прием происходит по счету (накладной), который прилагается к партии документов и обязательно содержит их описание. Если документы поступают без описи (например, в дар), то библиотекарь составляет акт на их получение, который передается вместе с другими документами на следующие операции. Все поступившие документы сверяются с картотекой комплектования и каталогами (на предмет дублетности) и маркируются штампом или штрих-кодом, удостоверяющими, что они стали собственностью библиотеки [21, с. 73].

При сверке возле каждого названия в описи ставится пометка. Далее просматривается каждая отдельная книга, для того чтобы выяснить, нет ли, среди них бракованных экземпляров. При обнаружении в принимаемой партии

документов недостачи, дефектных экземпляров или замены одних документов другими, их заменяют в источнике комплектования, откуда поступила партия, составив предварительно акт в двух экземплярах — для библиотеки и для книготоргового учреждения. На документы, поступившие без сопроводительного документа, составляется акт из двух частей: текстовой и приложения — описи документов. Если последних в партии не более пяти, их список включают в текст акта. Нумерация актов о приеме начинается ежегодно с № 1. Если поступили документы, которые не были заказаны библиотекой, их сверяют с алфавитным каталогом и выясняют, нужны они библиотеке или нет. После проверки с сопроводительными документами (если нет неточностей) все экземпляры документов штампелюются. Штемпель ставится на титульном листе и на странице 17. Для особо ценных и редких изданий может применяться книжный знак или штрих-код [16, с. 56].

После этого сопроводительный документ передается в бухгалтерию.

Учет выбывающих документов.

Выбытие документов и библиотечного фонда оформляется актами. В каждом акте фиксируются документы, которые исключаются только по одной причине: ветхость, излишняя дублетность, утеря читателями и т.д. В акт могут включаться документы различного вида, но списываемые по одной причине. При выбытии документов длительного хранения к акту прилагается список, который не может быть заменен книжными формулярами. После утверждения акта, данные, содержащиеся в нем, фиксируются во второй части «Книги суммарного учета» библиотечного фонда».

При утере документов пользователями, составляется акт (на основании «тетради учета книг, принятых от читателей взамен утерянных»). Содержание этого акта записывается в первой части «Книги суммарного учета»

Акт о списании недостающей литературы выявленной при проверке книжного фонда утверждается генеральным директором.

Переоценка библиотечного фонда.

Списание документов, состоящих на бухгалтерском учете, производится с применением переоценочных коэффициентов. Работа по переоценке фонда – это новое направление в работе библиотек. Библиотечные фонды переоцениваются по годам поступления документов на основании записей его индивидуальному и суммарному учету и бухгалтерских документов. Основное условие переоценки –

соответствие стоимости бухгалтерского учета данные первичного библиотечного учета. Исходные данные для переоценки – балансовая стоимость фонда по данным бухгалтерского учета. При переоценке каждого издания за основу принимается первоначальная цена, затем определяется восстановительная стоимость на день утраты (цена приобретения, умноженная на коэффициент соответствующий году.

Проверка библиотечного фонда.

В соответствии с Законом РФ «О бухгалтерском учете» проводится обязательная проверка книжного фонда с целью определения наличия зарегистрированных в учетной документации изданий, а также установления их соответствия учетным документам.

Во время проверки анализируется правильность ведения учетной документации, устраняются ошибки, допущенные в период между проверками. Проверки могут быть периодически, регулярные, и т.д. Проверке подлежат все виды документов, электронные издания, всем документам, которым присвоен инвентарный номер.

Основанием для проверки является приказ генерального директора. Создается комиссия, разрабатывается план проверки, сроки.

В подготовительный период подвергается проверке правильность оформления учетных документов (книги суммарного учета, инвентарной книги, накладных, актов и т.д.). На этом этапе должны быть выявлены и исправлены все ошибки в ведении учетных документов.

Проверка фонда может проходить по одному из способов проверки: (сверка с инвентарной книгой, при помощи контрольных талонов, по топографическому каталогу [20, с. 105].

Завершающий этап – подведение итогов и оформление соответствующей документации – акт (с объяснительной запиской) и список недостающих документов.

Документы, в которых осуществляется учет библиотечного фонда, являются финансовыми документами и должны храниться в сейфах или специальных шкафах.

2.2 Анализ информационного обеспечения библиотеки

Движение документов в библиотеке с момента их создания или получения до завершения исполнения или отправки образует документооборот. Физическое перемещение информации от одного сотрудника предприятия к другому или от одного подразделения к другому называется информационным потоком.

Основными направлениями работы любой библиотеки являются: комплектование и организация книжного фонда, обслуживание читателей.

Проведем анализ информационного обеспечения конкретной библиотеки - МБОУ «Мичуринская средняя общеобразовательная школа».

Библиотека в школе существует с момента открытия школы. Располагается библиотека на 2-ом этаже.

Услугами библиотеки пользуются учителя, ученики и работники школы.

Объём библиотечного фонда: 4117 экз. В том числе:

- учебной литературы – 3130 экз.
- художественной литературы – 987 экз.
- Электронные издания и карты

В структуру школьной библиотеки входят:

- абонементский отдел выдачи книг на дом;
- читальный зал (на 6 посадочных мест) - место, где учащиеся имеют возможность подготовить рефераты, доклады, домашние задания, почитать любимые журналы, ознакомиться с новой выставкой;
- книгохранилище – место, где хранится литература.

Читателями школьной библиотеки могут быть все учащиеся школы и родители, учителя и работники школы.

Рассмотрим наиболее трудоемкие бизнес-процессы в библиотеке МБОУ «Мичуринская средняя общеобразовательная школа» с точки зрения обработки информации.

1. Работа с книгами

1. Поступление книг.

2. Списание книг.

3. Инвентаризация книг.

1. Работа с читателями.

1. Заведение / продление формуляра.

2. Выдача / прием книг

3. Отчетность

Рассмотрим более подробно каждую группу процессов.

Организация книжного фонда включает вопросы учёта, расстановки, хранения литературы и доставки её читателю. Правильная организация фонда облегчает читателю пользование литературой, библиотекарю — быстрое выполнение читательских требований, а также обеспечивает сохранность фондов как общественной собственности.

1. Работа с книгами.

Учет библиотечного фонда должен отражать поступление, выбытие изданий, общую численность и распределение его по видам и содержанию. Учету в библиотеке подлежат все виды изданий и материалов в соответствии с ГОСТ 7.20–80. Основной единицей учета изданий в соответствии с ГОСТ 7.20–80 является экземпляр.

При поступлении партии книг в библиотеке выполняются следующие операции:

1. Поступающие издания принимаются, по сопроводительному документу (счет, копия счета, накладная, акт), за исключением журналов и других изданий, получаемых по подписке. При обнаружении в принимаемой партии недостачи замены одних книг другими или дефект изданий составляется акт в двух экземплярах. Один экземпляр акта пересылается в организацию, от которой поступили книги, с требованием дослать недостающие или заменить дефектные экземпляры, другой — остается в библиотеке.

2. Делается запись в Книге суммарного учета библиотечного фонда – одной строкой для партии книг. Суммарный учет дает общие сведения о фонде библиотеки: его объеме (количестве экземпляров), составе по содержанию отраслей знаний, видам изданий, языкам, стоимости книг, а также содержит данные о движении фонда. Приложение 1.

3. Делается запись в Книге учета библиотечного фонда – одна строка для каждой книги. Здесь же проставляются присвоенные инвентарные номера. Присваивание выполняется вручную.

4. Делается запись в инвентарной книге. При безинвентарном учёте литературы, поступающей в библиотеку (газеты, журналы и книги, содержащие менее 48 стр.) составляется учетная карточка, которая регистрируется в журнале, журнал заполняется вручную.

4. Заполняются регистрационные карточки для картотеки (вручную или с помощью компьютера) на все группы экземпляров. Приложение 2.

5. Карточки регистрируются в Журнале регистрации карточек учетного каталога. Приложение 3.

6. Печать списка поступивших книг.

При выбытии книг в библиотеке МБОУ «Мичуринская средняя общеобразовательная школа» выполняются следующие операции:

1. На книги, выбывшие из библиотеки по разным причинам, сначала делается запись по каждой выбывшей книге в тетради (копится определенное количество – более 100 экз.).

2. Составляется акт в 3-х экземплярах. В каждый акт вносятся книги, исключаемые только по какой-либо одной причине (по ветхости, как устаревшие по содержанию и др. причины). Акт утверждается директором МБОУ «Мичуринская средняя общеобразовательная школа».

На основе учетных документов осуществляется контроль за сохранностью фонда, проверка и передача его от одного работника другому, ведется статистическая отчетность.

Инвентаризация основных средств в библиотеке МБОУ «Мичуринская средняя общеобразовательная школа» проводится один раз в год, а библиотечных фондов

- один раз в пять лет.

При инвентаризации библиотечного фонда каждый экземпляр книги, подлежащий инвентаризации пересчитывается и сравнивается с записью в инвентарной книге. Инвентарная книга — важнейший документ библиотеки, который хранится наравне с денежными документами. Приложение 4.

2. Работа с читателями.

При обслуживании читателей библиотеки МБОУ «Мичуринская средняя общеобразовательная школа» осуществляются следующие операции:

1. При поступлении нового читателя заполняется читательский формуляр. При записи нового читателя в формуляр включаются следующие реквизиты:

- фамилия;
- имя;
- отчество;
- адрес: улица, дом, квартира;
- контактный телефон

2. При выдаче книги выполняется запись в формуляре.

3. При приеме книги делается отметка о возврате в формуляре.

4. При продлении срока делается отметка о продлении в формуляре

5. Каждое обращение читателя отмечается в тетради регистрационного учета вручную, в конце смены подсчитывается число обращений.

При этом возникает необходимость скорейшего оснащения библиотеки современной компьютерной техникой и организации системы повышения квалификации библиотечных работников. Причем такая система должна быть, не только постоянно действующей, но и многоступенчатой, поскольку персонал библиотеки должен пройти путь от традиционных библиотекарей до информационных консультантов, способных правильно ориентировать своих читателей в многообразии информационных ресурсов, доступных по сети Интернет, а также обеспечивать им получение запрошенной информации с требуемой полнотой и точностью. Первостепенное значение при этом приобретает наполненность сети российскими информационными ресурсами.

Важно подчеркнуть, что обеспечить современные потребности, библиотека сможет только при условии комплексной автоматизации технологических процессов своей деятельности: ведения каталогов, обработки новых поступлений, регистрация читательских запросов и др. [20, с.136]. Именно компьютерная база автоматизированной системы создает предпосылки подключения библиотеки к информационным сетям, ее полноправного участия в формировании информационного пространства. В реализации данного направления принципиально возможны два пути. Первый связан с проектированием, разработкой и единовременным внедрением системы поддержки библиотечной деятельности, отражающей все аспекты требуемых информационных технологий. Второй - это постепенный охват автоматизацией сфер деятельности библиотеки и эволюция предоставляемых услуг.

Таким образом, совершенствование ИС значительно облегчит и качественно улучшит многие процессы библиотечной деятельности.

3. Программное обеспечение задачи

3.1 Техническое задание

Разрабатываемый АРМ предназначен для ввода, хранения и обработки информации о библиотечных изданиях (монографиях, справочниках, сборников статей и т.п.), информации о месте положения отдельных экземпляров (переплёт) каждого издания, сведений о книгообороте и данных о читателях. Реализация такого АРМ позволит значительно облегчить работу сотрудников библиотеки: Поможет быстро и оперативно получать требуемую информацию; Облегчит ввод данных; Позволит быстро получать отчёты; Позволит быстро и точно найти или подобрать читателю информацию об издании.

Программный продукт должен обеспечивать работу в рамках архитектуры «Клиент-сервер».

Требования к надежности

Разрабатываемый продукт должен выдавать сообщения о возникающих ошибках при неверном задании исходных данных, поддерживать диалоговый режим в рамках предоставляемых пользователю возможностей.

Разрабатываемый продукт должен корректно реагировать на удаление, изменение данных, т.е. обеспечивать целостность данных. В случаях неправильных действий со стороны пользователя система должна выдавать соответствующие предупреждения.

Конечный пользователь не должен иметь свободный доступ к исходным файлам системы. Для модификации программой части системы пользователю необходимо обращаться к разработчику.

Требования к составу и параметрам технических средств

Разрабатываемая система предназначена для использования на персональных компьютерах типа IBM с центральным процессором Intel Pentium III 1000ЕВ или мощнее (рекомендовано Intel Pentium IV), ОЗУ объемом не менее 256 Мб, свободным дисковым пространством не менее 100Мб. Обязательным условием является наличие манипулятора типа «мышь», SVGA-адаптера и монитора с разрешением не менее 1024x768. Дополнительно необходимо наличие принтера для возможности распечатки отчетов.

Требования к информационной и программной совместимости

Приложение должно быть разработано с применением среды разработки Delphi 7 и СУБД FireBird версии не ниже v2.5.1.

Приложение должно работать в операционных системах MS Windows XP и Windows 7. Отчеты, создаваемые приложением, должны быть совместимы с MS Excel.

Требования к программной документации

Разрабатываемый проект должен сопровождаться документацией, в составе которой будут учтены следующие пункты:

- техническое задание;
- описание программы;
- описание применения;
- листинги программ, результаты работы.

Стадии и этапы разработки

АРМ должно разрабатываться в следующем порядке:

- анализ предметной области;

- разработка технического задания;
- разработка функциональной модели;
- разработка инфологической модели;
- разработка программных модулей продукта;
- тестирование и отладка системы;
- оформление документации.

Порядок контроля и приемки

В целях контроля за правильностью работы системы, должно проводиться тестирование, в котором должны быть учтены все возможные сочетания исходных данных, отражающие работу всех частей системы. Пояснительная записка должна соответствовать ГОСТ. Для тестирования системы должны быть подготовлены тестовые примеры, охватывающие все функциональные возможности разрабатываемой системы. Правильность работы определяется по результатам выполнения этих примеров.

Приемка программного продукта должна проводиться при представлении работоспособности системы при различных входных данных, при правильном выполнении всех функций и при наличии полной документации к программе.

3.2 Обоснование выбора среды разработки

В настоящее время существует множество СУБД, таких как FoxPro, SQLServer, MS Access, Firebird и т.д.

FoxPro — один из диалектов языка программирования xBase, применяемый в одноименном программном пакете. Как язык программирования, в основном применяется для разработки файл-серверных реляционных СУБД, хотя существует, за счет гибких и богатых средств языка, возможность разработки и других классов программ. В настоящее время используется в среде разработки Microsoft Visual FoxPro.

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

персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Microsoft Access - реляционная СУБД корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных.

В данной работе используется клиент-серверная технология. Для разработки серверной части была выбрана система управления базами данных (СУБД) Firebird-2.5.1, языком программирования клиентского приложения выбран язык программирования Delphi-7. Основанием для этого выбора послужило следующее.

Во-первых, Firebird зарекомендовала себя как достаточно надежная СУБД, хорошо восстанавливается после сбоев. Язык программирования PSQL наиболее легкий и логичный, по сравнению, например, с Oracle PL/SQL или Microsoft T-SQL. В то же время PSQL позволяет выполнять сложные запросы, включая рекурсивные.

Во-вторых, Firebird является бесплатным продуктом.

В-третьих, Delphi относится к классу инструментальных средств ускоренной разработки программ, который включает в себя полный набор визуальных инструментов для скоростной разработки приложений (RAD - rapid application development), поддерживающей разработку пользовательского интерфейса. При этом Delphi обладает достаточной мощностью и строгой типизированностью, в отличие, например, от Visual Basic.

В-четвертых, в Delphi встроены средства работы с Interbase/Firebird, что позволяет обойтись без BDE, то есть увеличивается скорость работы приложений. В то же время использование IBx-компонентов упрощает процесс программирования, по сравнению с ADO и DBExpress.

Преимущества разработки информационных систем на их основе заключаются в следующем.

SQL-серверы прямо ориентированы на создание интегрированных, многопользовательских систем, имея в своем распоряжении развитые словари данных.

Средства разработки для этих СУБД оптимизированы в отношении коллективной разработки сложных систем в рамках единой стратегической линии.

Развитый механизм обработки транзакций позволяет обеспечить целостность данных при одновременной работе многих пользователей.

Использование единого языка доступа к данным (SQL) позволяет упростить переход от одной СУБД к другой.

Обеспечивается масштабируемость разрабатываемых систем. Поддерживается возможность работы, как в локальной, так и в глобальной сетях.

Используемая в разработке СУБД FireBird (созданная компанией IBPhoenix, группой разработчиков знаменитой Borland InterBase) в полной мере удовлетворяет всем перечисленным требованиям. FireBird представляет собой полнофункциональный SQL-сервер.

Сервер баз данных - это программный процесс, который выполняется на узле сети, где расположен главный компьютер и физически расположена сама база данных. Процесс сервера - единственный процесс на любом узле, который может исполнять прямые операции ввода-вывода для файлов базы данных.

Серверный процесс является полностью сетевым, он поддерживает запросы на подключение от других узлов сети и тот же самый протокол FireBird прикладной программы, что и клиентские процессы. Несколько клиентов могут быть связаны с многопоточным процессом сервера одновременно.

Сервер регулирует доступ к отдельным записям данных в пределах базы данных и обеспечивает монопольный доступ к записям, когда клиенты выдают запросы на изменение данных в записях.

- Отличительными качествами FireBird являются:
- высокая производительность и надежность сервера при минимальных требованиях к техническим средствам;
- поддержка стандарта SQL-92, обеспечивающая переносимость приложений;
- продукт является бесплатным и доступным для скачивания;
- простота установки и поддержки сервера. Удобный и не требующий специальной подготовки механизм администрирования базой данных;
- большая популярность в учебных и государственных учреждениях;
- отличные справочные руководства.

Все это, делает FireBird прекрасным выбором для реализации корпоративных систем малого и среднего масштаба (с количеством пользователей в несколько

десятков). При реализации очень крупных проектов (с сотнями или более пользователей) стоит, наверное, рассмотреть более мощные серверы - типа Oracle или Informix.

Системные требования FireBird.

FireBird работает на различных платформах, включая Microsoft Windows и разные версии операционной системы UNIX.

Память: минимум 256 Мб (для сервера рекомендуется 512). Процессор: 586 или больше рекомендуется для мультиклиентского сервера. Компиляторы: Microsoft Visual C++ 4.2 и Borland C++ 5.0, C++ Builder, Delphi.

Как мы видим, требования к аппаратному обеспечению не превышают требований, предъявляемых операционной системой.

В версии 1.5 добавлены:

- ряд изменений языка запросов и хранимых процедур, позволяющих повысить производительность труда программиста;
- улучшена работа оптимизатора плана выполнения запросов и процедур, что увеличивает производительность работы операторов.

Таким образом, все указанные достоинства позволяют выбрать FireBird 2.5.1 в качестве СУБД.

Обоснование выбора системы программирования Delphi-7.

Главные достоинства:

- простота разработки приложений для баз данных;
- совместимость с InterBase / FireBird;
- большое количество литературы и интернет-ресурсов.

Системные требования к Delphi-7 заметно выше, чем к FireBird, но и они укладываются в требования для нормальной работы Microsoft Windows, Microsoft Office.

3.3 Описание серверной части

В состав базы данных входят домены, генераторы, таблицы, первичные и внешние ключи, индексы, хранимые процедуры, триггеры.

Список таблиц, генераторов, триггеров, первичных ключей приведен в таблице 2.1. Перечень доменов приведён в таблице 2.2. Структура таблиц приведена в таблицах 2.3 – 2.7. Список хранимых процедур приведен в таблице 2.8. Полный сценарий создания базы данных приведен в приложении 5.

Схема базы данных приведена на рисунке 2.1.

Таблица 2.1.

Таблицы, генераторы, первичные и внешние ключи, триггеры

Название таблицы	Генератор	Триггер	Первичный ключ	Внешний ключ
BOOK	GEN_BOOK_ID	BOOK_BI	PK_BOOK	
BOOKMOVE	GEN_BOOKMOVE_ID	BOOKMOVE_BI	PK_BOOKMOVE	FK_BOOKMOVE_1
PUBLISHER	GEN_PUBLISHER_ID	PUBLISHER_BI	PK_PUBLISHER	
READER	GEN_READER_ID	READER_BI	PK_READER	
READING	GEN_READING_ID	READING_BI	PK_READING	FK_READING_1, FK_READING_2

Таблица 2.2.

Домены (пользовательские типы данных)

Название домена	Тип	Назначение
-----------------	-----	------------

T_AMOUNT	INTEGER	Количество
T_AUTHOR	VARCHAR(150)	Автор
T_BOOKKIND	VARCHAR(30)	Вид издания
T_BOOKNAME	VARCHAR(100)	Название книги
T_BOOKNUM	VARCHAR(10)	Инвентарный номер
T_CLASS	AS VARCHAR(5)	Класс
T_DATE	DATE	Дата
T_DIR	CHAR(1)	Признак: поступление '+'/списание '-'
T_DOC	VARCHAR(50)	Документ
T_FIRSTNAME	VARCHAR(50)	Имя
T_ID	INTEGER	Идентификатор строки
T_ISBN	VARCHAR(20)	ISBN
T_LASTNAME	VARCHAR(50)	Фамилия
T_MIDNAME	VARCHAR(50)	Отчество
T_NAME_PUBLISHER	VARCHAR(100)	Издательство

T_NUMBERS	VARCHAR(200)	Список (диапазон) инв. номеров
T_TICKET	VARCHAR(10)	Читательский билет
T_YEAR	INTEGER	Год

Таблица 2.3.

Таблица BOOK («Книги»)

Название	Тип	Назначение	Ограничения
BOOK_ID	T_ID	ИД строки	NOT NULL. Первичный ключ
NAME	T_BOOKNAME	Название книги	
AUTHOR	T_AUTHOR	Автор	
ID_PUBLISHER	T_ID	ИД издательства	Внешний ключ
YEAROF	T_YEAR	Год издания	
KIND	T_BOOKKIND	Вид издания	
ISBN	T_ISBN	ISBN	
AMOUNT	T_AMOUNT	Количество	
NUMBERS	T_NUMBERS	Инвентарные номера	

Таблица 2.4.

Таблица BOOKMOVE («Поступление/выбытие»)

Название	Тип	Назначение	Ограничения
BOOKMOVE_ID	T_ID	ИД строки	NOT NULL. Первичный ключ
BOOK_ID	T_ID	ИД Книги	Внешний ключ
DAYOF	T_DATE	Дата	
DOC	T_DOC	Документ	
DIR	T_DIR	Поступление/выбытие	
AMOUNT	T_AMOUNT	Количество	
NUMBERS	T_NUMBERS	Инв. номера	

Таблица 2.5.

Таблица PUBLISHER («Издательства»)

Название	Тип	Назначение	Ограничения
ID_PUBLISHER	T_ID	ИД строки	Not null. Первичный ключ
NAME_PUBLISHER	T_NAME_PUBLISHER	Название издательства	

Таблица 2.6.

Таблица READER («Читатели»)

Название	Тип	Назначение	Ограничения
READER_ID	T_ID	ИД строки	Not null. Первичный ключ
LASTNAME	T_LASTNAME	Фамилия	
FIRSTNAME	T_FIRSTNAME	Имя	
MIDNAME	T_MIDNAME	Отчество	
TICKET	T_TICKET	Билет	
DAYOF	T_DATE	Дата выдачи	
CLASS	T_CLASS	Класс	

Таблица 2.7.

Таблица READING («Чтение»)

Название	Тип	Назначение	Ограничения
READING_ID	T_ID	ИД строки	Not null. Первичный ключ
READER_ID	T_ID	ИД читателя	Внешний ключ
BOOK_ID	T_ID	ИД книги	Внешний ключ
DAYGET	T_DATE	Дата выдачи	

DAYVALID	T_DATE	Выдано до
DAYRETURN	T_DATE	Дата возврата
BOOKNUM	T_BOOKNUM	Инв. номер

Таблица 2.8.

Хранимые процедуры, связанные с таблицей Book

Таблицы	Название	Назначение	Аргументы	Возвращаемые значения
Book	Book_Del	Удаление строк	BOOK_ID	
Book	Book_Ins	Вставка строки	name, author, id_publisher, yearof, kind, isbn, amount, numbers	BOOK_ID
Book	Book_Upd	Модификация строк	Book_id, name, author, id_publisher, yearof, kind, isbn, amount, numbers	
Book, Reading	Reading_SEL1	Выборка	reader_id	reading_id, book_id, dayget, dayvalid, dayreturn, booknum, bookname, author

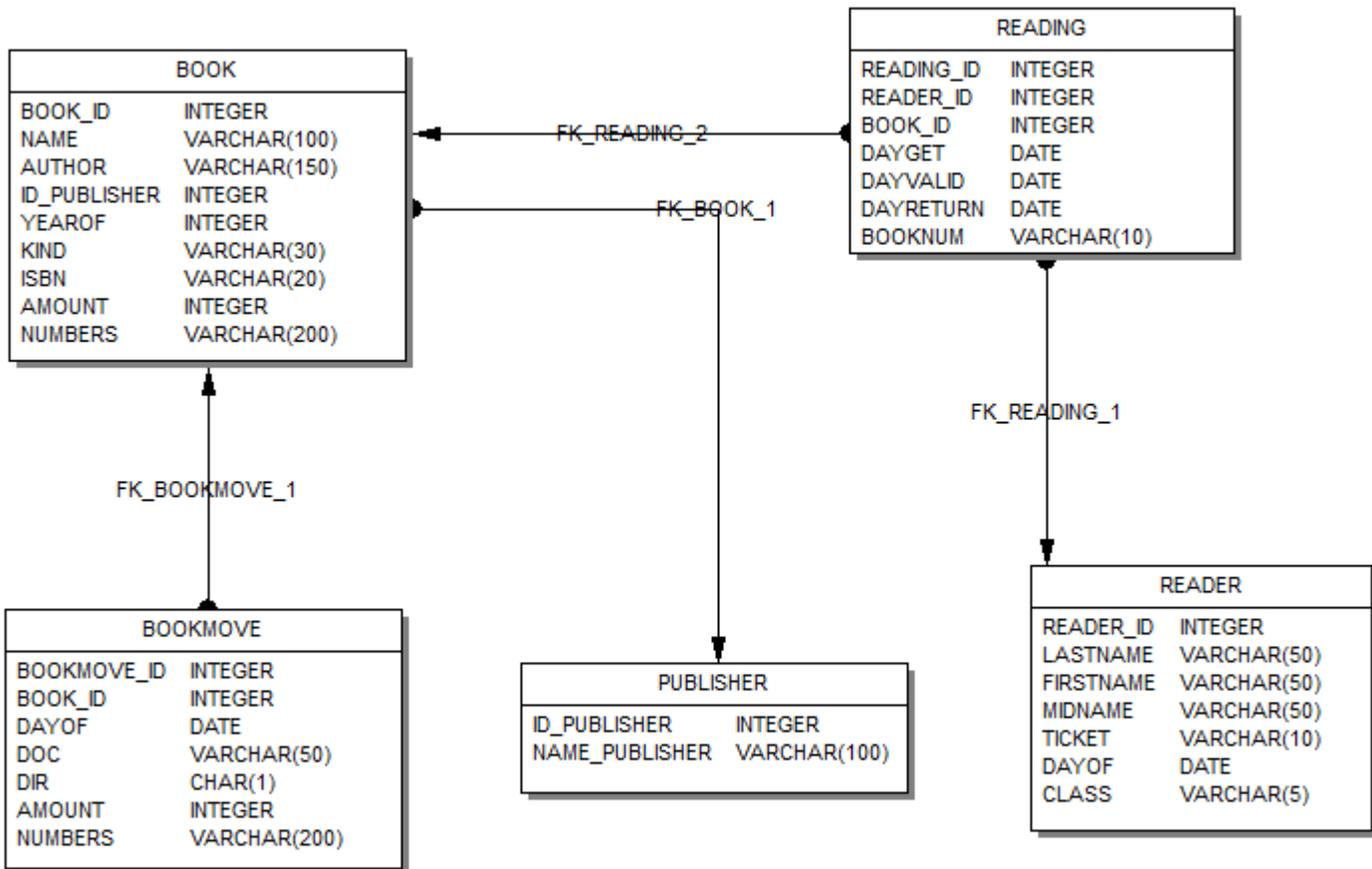


Рис 3.1. Схема базы данных

3.4 Описание клиентской части

Перечень файлов клиентской части проекта приведён в табл. 2.10. , текст программных модулей и файла проекта приведён в приложении 6.

Таблица 2.10

Файлы проекта

Название файла

Назначение

frmAboutUnit.dfm, frmAboutUnit.pas Форма и модуль «О программе»

frmLoginUnit.dfm, frmLoginUnit.pas Форма и модуль авторизации

uMain.dfm, uMain.pas	Главная форма и модуль меню
pLibrary.cfg	Файл конфигурации проекта
pLibrary.dof	Служебный файл проекта
pLibrary.dpr	Файл модулей и форм проекта
pLibrary.exe	Исполняемый файл
pLibrary.ini	Файл настроек подключения
pLibrary.res	Файл ресурсов
rep01.xlt	Шаблон отчёта «Движение книг»
rep02.xlt	Шаблон отчёта «Выданные книги»
uDm.dfm	Модуль данных
uDm.pas	
uBook.dfm	Список книг
uBook.pas	

Продолжение таблицы 2.10

uBookEd.dfm	Карточка книги
uBookEd.pas	

uPublisher.dfm	Список издательств
uPublisher.pas	
uPublisherEd.dfm	Карточка издательства
uPublisherEd.pas	
uReader.dfm	Список читателей
uReader.pas	
uReaderEd.dfm	Карточка читателя
uReaderEd.pas	
uMove.dfm	Список перемещений книги
uMove.pas	
uMoveEd.dfm	Карточка перемещения книги
uMoveEd.pas	
uReading.dfm	Выдачи книг читателю
uReading.pas	
uReadingEd.dfm	Карточка выдачи книг читателю
uReadingEd.pas	

Процедуры и функции модуля для работы со списком книг (uBook) приведены в таблице 2.11.

Таблица 2.11

Процедуры и функции модуля uBook

Процедура / функция	Назначение	Входные (in) и выходные (out) параметры
procedure TfBook.FormActivate	При активации формы	(Sender: TObject);
procedure TfBook.btEditClick	Вызов формы редактирования и корректировка строки	(Sender: TObject);
procedure TfBook.btFindClick	Поиск книги	(Sender: TObject);
procedure TfBook.btPrintClick	Вызов формы редактирования и корректировка строки	(Sender: TObject);
procedure TfBook.btDeleteClick	Удаление строки	(Sender: TObject);
procedure TfBook.btNewClick	Вызов формы редактирования и вставка строки	(Sender: TObject);
procedure TfBook.cmdMoveClick	Вызов формы перемещения книги	(Sender: TObject);
procedure TfBook.FormClose	Закрытие формы	(Sender: TObject; var Action: TCloseAction);
procedure TfBook.FormCreate	При создании формы	(Sender: TObject);

3.5 Разработка экранных форм и структуры их взаимосвязей

Структура взаимосвязи форм приведена на рисунке 2.2., некоторые формы на этапе разработки приведены на рисунках 2.3 – 2.8.

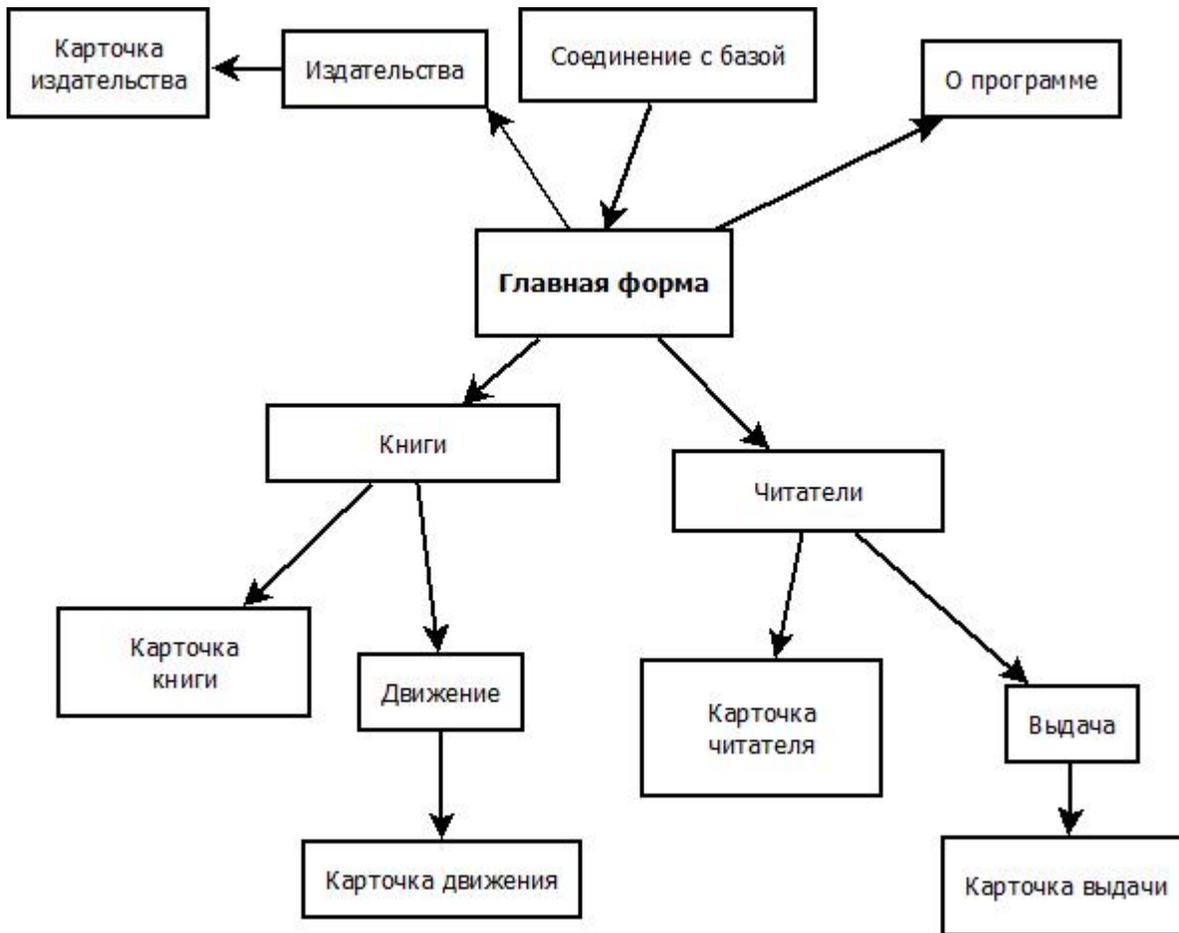


Рис. 3.2 Структура взаимосвязи экранных форм

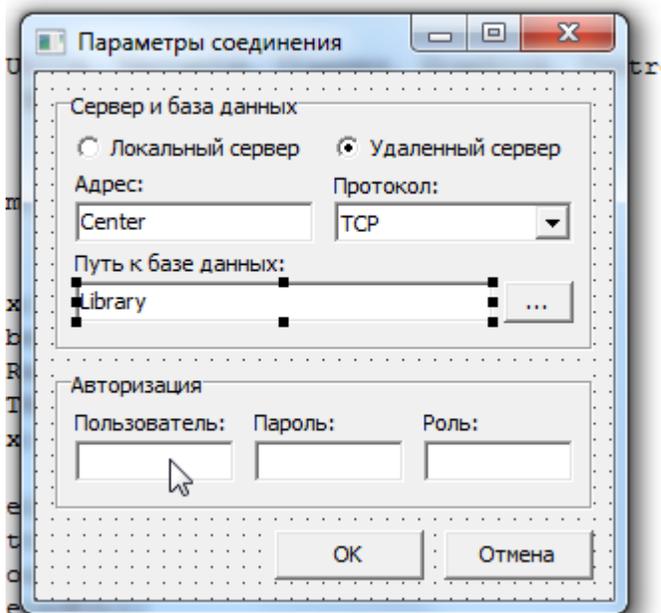


Рис. 3.3 Форма подключения к базе

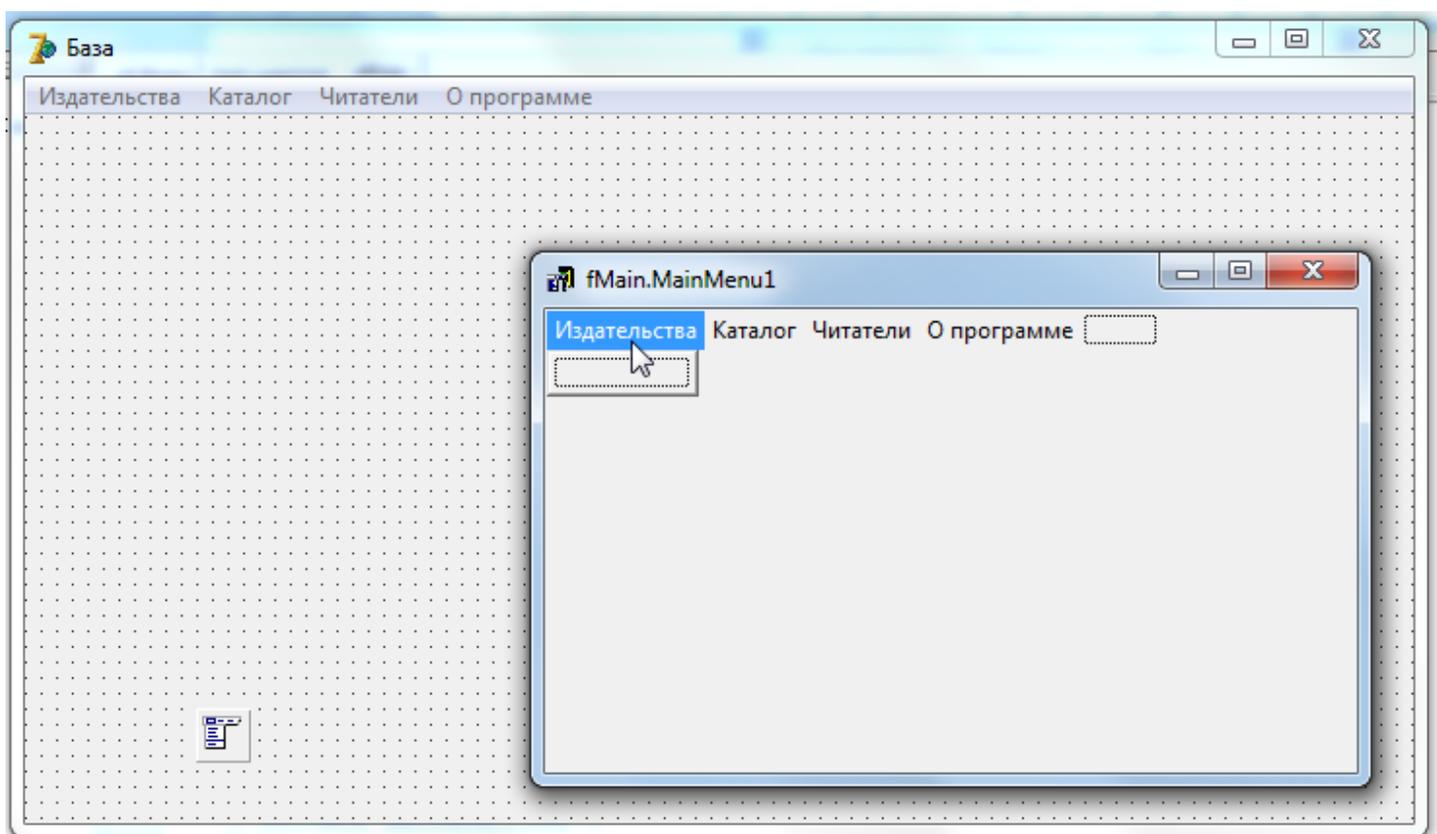


Рис. 3.4 Главная форма

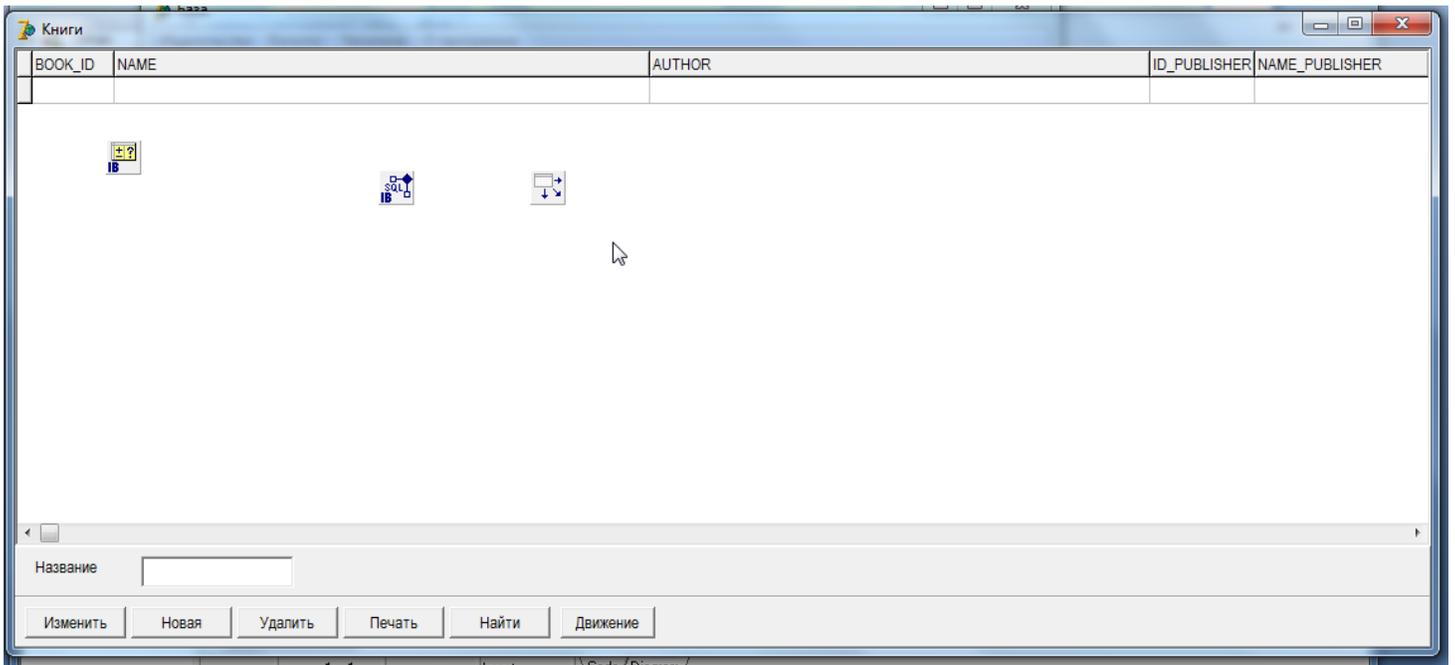


Рис. 3.5 Список книг

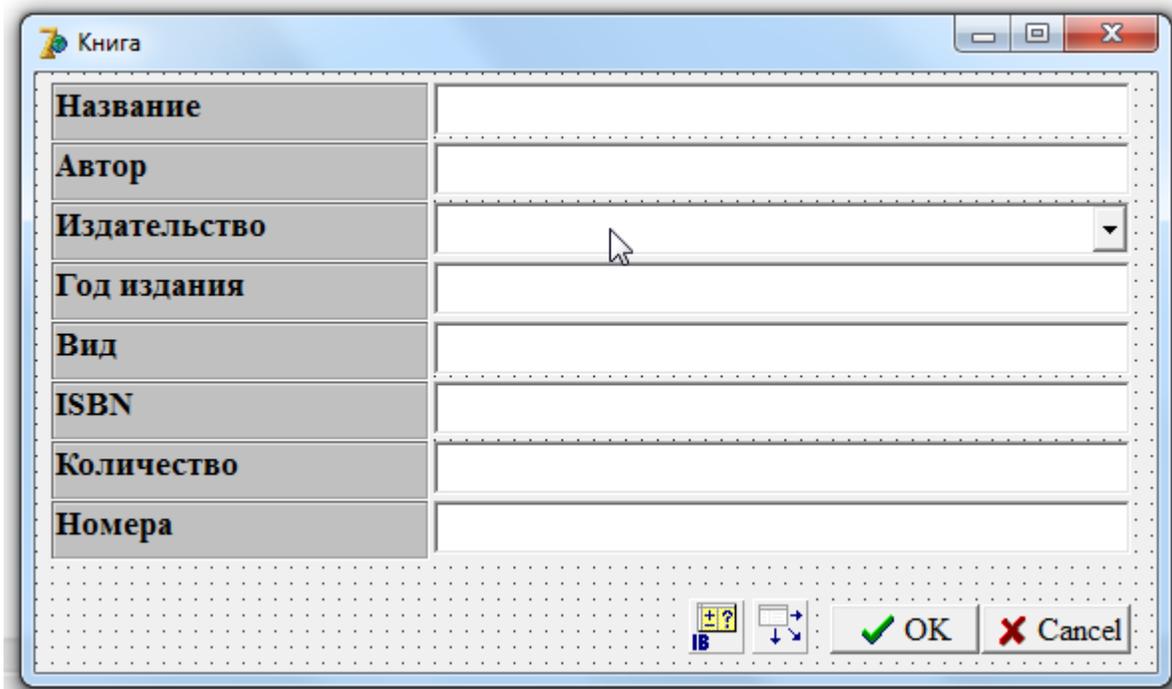


Рис. 3.6 Карточка книги

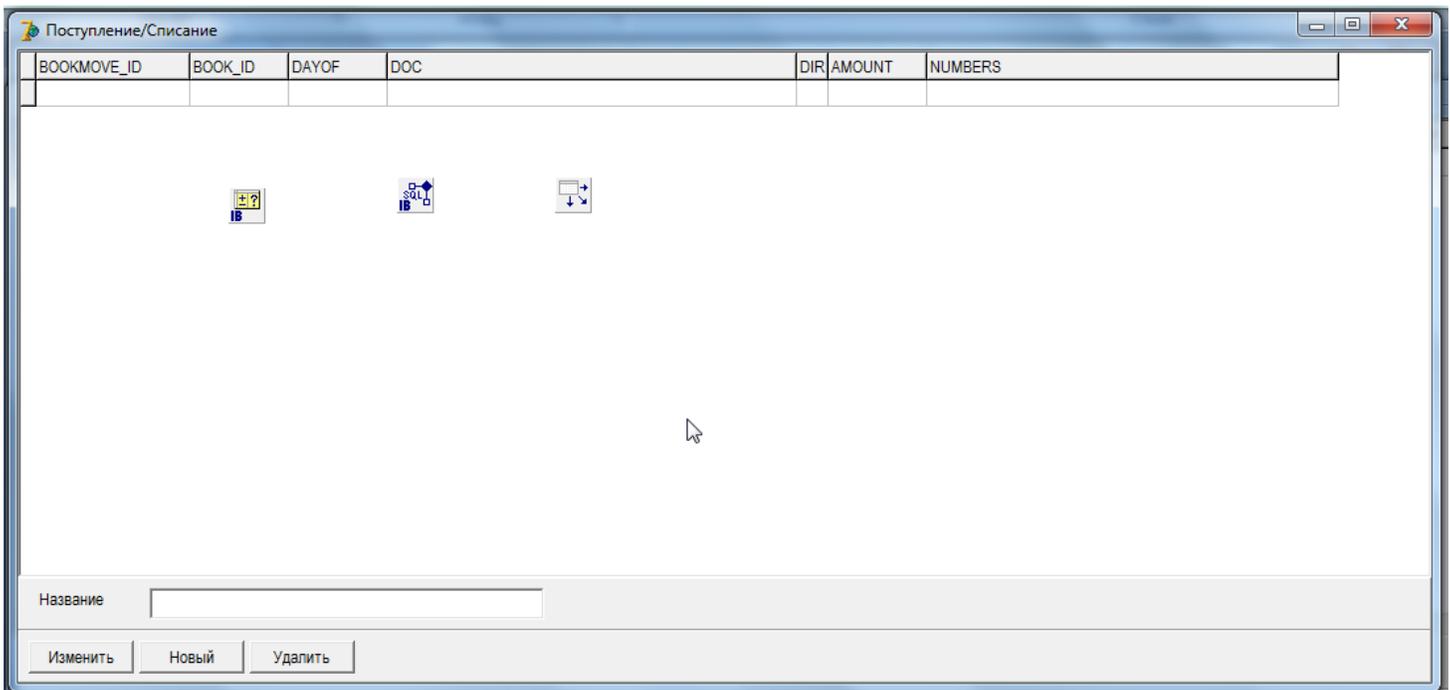


Рис. 3.7 Поступление / выбытие

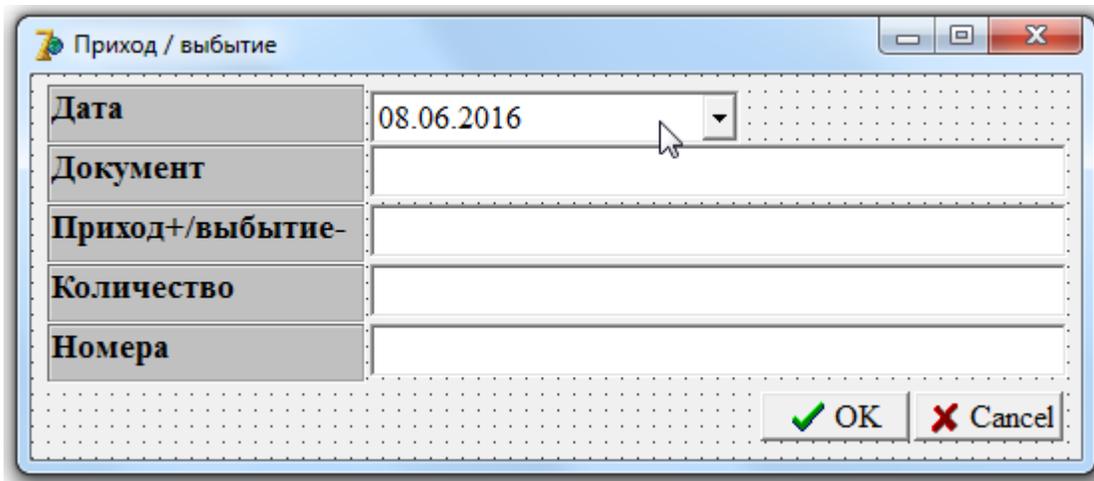


Рис. 3.8 Карточка строки поступления / выбытия

Рассмотрим более подробно «Список книг» (uBook.pas, uBook.dfm).

На форме размещены следующие компоненты:

- компонент типа TdbGrid – для отображения табличной информации,
- компонент типа TIBDataSet – для вызова запроса,
- компонент типа TIBStoredProcedure – для вызова хранимой процедуры,
- компонент типа TDataAccess – для связи dbGrid с источником данных IBDataSet,
- 3 компонента типа TButton для обработки строк
- 1 компонент типа TButton для перехода к форме перемещения книг,

- 1 компонент типа TButton для выполнения операции «Печать»,
- 1 компонент типа TButton для выполнения поиска,
- 1 компонент Panel – контейнер для размещения компонентов TButton.

Используется обработка событий нажатия клавиш. По btEditClick – подготавливаются начальные значения, вызывается форма редактирования fBookEd, коррективка строки.

На событие btNewClick – подготовка значений, вызов формы редактирования и вставка строки.

На событие btDeleteClick – создаётся диалоговое окно MessageDlg и, в случае подтверждения, – подготавливаются данные для вызова процедуры удаления строки.

На событие cmdMoveClick вызывается форма списка движения книг.

ЗАКЛЮЧЕНИЕ

В данном дипломном проекте было выполнено проектирование программы, для рабочего места сотрудника библиотеки.

Объектом исследования стало создание и ведение базы данных школьной библиотеки.

Предметом исследования стала работа библиотекаря.

Разработка представленной системы характеризуется наименьшими затратами. Это достигается следующим:

1. Используется свободное программное обеспечение:

- сервер баз данных FireBird;
- интегрированная среда разработки Code::Blocks, включая компилятор GCC;
- текстовый редактор NotePad++;

2. Возможно многократное использование клиентской программы без оплаты дополнительных лицензий, как, например, в случае с покупными системами.

На данном этапе проводится опытная эксплуатация программного продукта. Предполагается дальнейшее развитие и совершенствование разработанной информационной системы учета книжного фонда.

Выходные данные имеют возможность выгрузки в MS Excel.

Программный продукт обеспечивает работу в рамках архитектуры «Клиент-сервер».

Разработанная программа позволит сократить время выполнения функций сотрудников библиотеки, устранить случайные ошибки при расчетах, автоматизировать большую часть информационных процессов, и повысить учётно-аналитические возможности.

Были рассчитаны следующие экономические показатели: стоимость одного часа работы ЭВМ – 204,04 (руб.); затраты на составление и отладку программы – 12089,8 (руб.); денежная экономическая эффективность применения ПЭВМ – 7013,76 (руб.); рост производительности труда составляет – 45,4 %. Программа окупит себя в конце четвертого месяца использования. Соответственно решение на внедрение собственного программного продукта целесообразно.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Федеральный Закон № 149-ФЗ от 27 июля 2006 года «Об информации, информационных технологиях и о защите информации» //Консультант Плюс.
2. Федеральный закон от 29 декабря 1994 г. N 78-ФЗ «О библиотечном деле» //Консультант Плюс.
3. Межотраслевые нормы времени на работы, выполняемые в библиотеках, утвержденные постановлением Министерства труда 03.02.1997 г. //Консультант Плюс.
4. Концепция развития библиотечного дела в Российской Федерации до 2015 года: Проект Национальная программа сохранения библиотечных фондов Российской Федерации// <http://www.library.ru>
5. Адамовский Д.В. Новейшие автоматизированные библиотечные системы России. – СПб., - 2014. – 213 с.
6. Архангельский А.Я. Object Pascal в Delphi. – М.: ЗАО «Издательство БИНОМ», 2012.- 324 с.

7. Борри Х, Firebird: Руководство разработчика баз данных. Пер. с англ. – СПб.: БХВ-Петербург, 2014. – 815 с..
8. Гофман В.Э., Хомоненко А.Д. Работа с базами данных в Delphi. – 2-е изд. – СПб.: БХВ-Петербург, 2012. – 547 с.
9. Воройский Ф.С. Информатика: новый систематизированный толковый словарь справочник (Вводный курс по информатике в терминах). 3-е изд., перераб. и доп. М.: Либерейя, 2011. 535 с.
10. Воройский Ф.С. Основы проектирования автоматизированных библиотечных информационных систем. М.: Физматлит, 2013. 384 с.
11. Дунаевская С.М., Соколова Е.П. Некоторые направления работ по созданию электронного каталога и автоматизированных библиотечных технологий // НТБ.-2014.-№ 1.-С.26-29
12. Земсков А.И., Информатизация библиотек: проблемы и перспективы. Социально психологический аспект// НТБ.-2013.-№ 10.-С.34-38
13. Кулыгина И.Ю. Описание электронных ресурсов // Библиография.-2015.-№ 3.-С. 13-14
14. Лавренова О.А. Есть такая запись в электронном каталоге // Библиотека. -2016. - № 2. -С 30-33; № 3.-С. 50-53; № 4.-С. 31-36
15. Майстрович Т.В. Электронная библиотека: новые грани нашей профессии // Мир библиографии.- 2015.-№ 4.-С. 7-10
16. Меррей Р. Компоненты цифровой библиотеки и их взаимодействие // НТБ.- 2015.-№ 6.- С. 56-65
17. Петров В.Н. Информационные системы: учеб. пособие. СПб.:Питер, 2016. – 527 с.
18. Пилко И.С. Информационные и библиотечные технологии: учеб. пособие. – М.:Кнорус, 2014. – 365 с.
19. Раев А.Г., Шрайберг Я.Л. Управляемая система массового обслуживания как модель оптимального библиотечного комплектования // НТБ - 2015. - № 5. С. 15-20.
20. Рощин А.Л. Автоматизация библиотек на современном этапе. – М.: Кнорус, 2013.- 228 с.
21. Стандарты по библиотечному делу: Сборник / Сост. Захарчук Т.В., Петрова Л.И., Завадовская Т.А., Зусьман О.М. - СПб: Изд-во «Профессия», 2014. - 512 с.
22. Столяров, Ю. Н. Документный ресурс: учебное пособие для студентов высших учебных заведений. – М.: Либерейя-Бибинформ, 2014. – 224 с. – (Спец. серия «Профессиональный практикум»).

23. Сукиасян Э.Р. Компьютеризация библиотеки: с чего начиналось, тем и закончилось / Э.Р. Сукиасян // Науч. и техн. б-ки. 2015. № 9. С. 17-28.
24. Тищенко М.Н. Изменения в Федеральном законодательстве и муниципальные библиотеки // <http://www.library.ru>
25. Шафрин Ю.А. Информационные технологии: В 2ч. Ч. 2: Офисная технология и информационные системы // <http://www.library.ru>
26. Шрайберг Я.Л. Основные положения и принципы разработки автоматизированных библиотечно-информационных систем и сетей: главные тенденции окружения, основные положения и предпосылки, базовые принципы. М.: Либерия, 2014. -100 с.
27. Шрайберг Я.Л. Моделирование и библиотека // НТБ . - 2015. - № 2. -С. 13-18.
28. Чекалов А. П. «Базы данных: от проектирования до разработки приложений» Спб: БХВ - Петербург, 2015 - 384 с.

Интернет ресурсы:

1. <http://notepad-plus-plus.org/>
2. <http://firebirdsql.org/>
3. <http://www.codeblocks.org/>

ПРИЛОЖЕНИЕ 1

Книга суммарного учета библиотечного фонда

9201

КНИГА
СУММАРНОГО
УЧЕТА
БИБЛИОТЕЧНОГО
ФОНДА

Начата _____ 198 г.

Окончена _____ 198 г.

ПРИЛОЖЕНИЕ 2

Карточка учета



Пайпс, Ричард.

Русская революция: [В 3 т.: Авториз. пер. с англ.]. — М.: Росспэн, 1994 — . — 22 см.

Перевод изд.: The Russian revolution / Richard Pipes (New York, 1990).

ISBN 5-86004-018-0.

Ч. 1. — 1994. — 397, [1] с. — Библиогр. в примеч.: с. 370—391.

— Имен. указ.: с. 392—397. — ISBN 5-86004-019-9 (в пер.): Б. ц., 5000 экз.

— — 1. Революционное движение в России — История, 20 в.
2. Россия — История, 20 в.

№ 21834

УДК [947+957] «1905/1917»

3 № 554 [94-28964] п оф

ББК 63.3(2)52

© Рос.кн.палата 19.12.94 П

Изд-во «Книжная палата»

Бизнес-карта-94: [Отрасль: Справочник: В 35 кн.]. — М.: Бизнес-карта, 1994 — — 21 см.

На корешке только загл. кн.

[Кн. 21]: Химическая и нефтехимическая промышленность: Кокс, продукты нефте- и газопереработки. Химические вещества. Лакокрасочная продукция и изделия из пластмасс. Фармацевтические препараты, парфюмерия, товары бытовой химии. Резинотехнические изделия, [ч. 2]: Закавказье, Средняя Азия, Казахстан, Беларусь, Молдова, Украина, страны Балтии. — 1994. — 351, [1] с.: схем. — Алф. указ. предприятий и по видам продукции: с. 318—352. — ISBN 5-87487-435-6: Б. ц., 2000 экз.

I. Загл. тома. — — 1. Химическая промышленность, предприятия — Справочники (кн. 21). 2. Нефтехимическая промышленность, предприятия — Справочники (кн. 21).

№ 21835

УДК 334.716(470)(035.5) +

5 № 1145 [94-29319] п оф

66(474 476+477/478.9+479.2+574/575)

© Рос.кн.палата 19.12.94 Б598

ББК 35

Изд-во «Книжная палата»

ПРИЛОЖЕНИЕ 3

Журнал регистрации карточек

ЖУРНАЛ РЕГИСТРАЦИИ КАРТОЧЕК УЧЕТНОГО КАТАЛОГА

Инв. №	Фамилия и инициалы автора или первое слово заглавия книги	Акт изъятия карточки	Инв. №	Фамилия и инициалы автора или первое слово заглавия книги	Акт изъятия карточки	Инв. №	Фамилия и инициалы автора или первое слово заглавия книги	Акт изъятия карточки	Инв. №	Фамилия и инициалы автора или первое слово заглавия книги	Акт изъятия карточки
1	2	3	1	2	3	1	2	3	1	2	3

ФОРМА ЖУРНАЛА РЕГИСТРАЦИИ ПОСТУПЛЕНИЙ В СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ ЦБС «ОПИСЬ ИНВЕНТАРНЫХ НОМЕРОВ»

Год. № записи в КУБФ, дата сдачи приема	Инв. №, расписка лиц и принявших книги	Отметка о проверке фонда	Фамилия, инициалы автора или первое слово заглавия	Цена		№ акта выдать	Год. № записи в КУБФ, дата сдачи приема	Инв. № расписки лиц, сдавших книги	Отметка о проверке фонда	Фамилия, инициалы автора или первое слово заглавия	Цена		№ акта выдать
				руб.	коп.						руб.	коп.	
1	2	3	4	5	6	7	1	2	3	4	5	6	7

Примечание. При поступлении нескольких экземпляров одного издания их количество отражается в гр. 2.

наименование библиотеки

ИНВЕНТАРНАЯ КНИГА

(_____ страниц _____ инвентарных номеров)

От инвентарного № _____

До инвентарного № _____

Примечание. Реквизиты «От инвентарного № До инвентарного №» заполняет библиотека

Инвентарная книга

Приложение 5

Серверная часть комплекса

/* Частичный сценарий создания базы данных AutoStore */

SET SQL DIALECT 3;

SET NAMES WIN1251;

CREATE DATABASE 'C:\Db\Library.FDB'

USER 'SYSDBA' PASSWORD 'masterkey'

PAGE_SIZE 16384

DEFAULT CHARACTER SET WIN1251 COLLATION WIN1251;

/* Домены */

CREATE DOMAIN T_AMOUNT AS INTEGER;

CREATE DOMAIN T_AUTHOR AS VARCHAR(150);

CREATE DOMAIN T_BOOKKIND AS VARCHAR(30);

CREATE DOMAIN T_BOOKNAME AS VARCHAR(100);

CREATE DOMAIN T_CLASS AS VARCHAR(5);

CREATE DOMAIN T_DATE AS DATE;

CREATE DOMAIN T_DAY AS DATE;

CREATE DOMAIN T_DIR AS CHAR(1);

CREATE DOMAIN T_DOC AS VARCHAR(50);

CREATE DOMAIN T_FIRSTNAME AS VARCHAR(50);

CREATE DOMAIN T_ID AS INTEGER;

CREATE DOMAIN T_ISBN AS VARCHAR(20);

CREATE DOMAIN T_LASTNAME AS VARCHAR(50);

```
CREATE DOMAIN T_MIDNAME AS VARCHAR(50);  
CREATE DOMAIN T_NAME_PUBLISHER AS VARCHAR(100);  
CREATE DOMAIN T_NUMBERS AS VARCHAR(200);  
CREATE DOMAIN T_TICKET AS VARCHAR(10);  
CREATE DOMAIN T_YEAR AS INTEGER;
```

```
/* Генераторы */
```

```
CREATE GENERATOR GEN_BOOKMOVE_ID;  
CREATE GENERATOR GEN_BOOK_ID;  
CREATE GENERATOR GEN_PUBLISHER_ID;  
CREATE GENERATOR GEN_READER_ID;
```

```
/* Таблицы */
```

```
CREATE TABLE BOOK (  
BOOK_ID T_ID NOT NULL,  
NAME T_BOOKNAME,  
AUTHOR T_AUTHOR,  
ID_PUBLISHER T_ID,  
YEAROF T_YEAR,  
KIND T_BOOKKIND,  
ISBN T_ISBN,  
AMOUNT T_AMOUNT,  
NUMBERS T_NUMBERS  
);
```

```
CREATE TABLE BOOKMOVE (
```

```
BOOKMOVE_ID T_ID NOT NULL,  
  
BOOK_ID T_ID,  
  
DAYOF T_DATE,  
  
DOC T_DOC,  
  
DIR T_DIR,  
  
AMOUNT T_AMOUNT,  
  
NUMBERS T_NUMBERS  
  
);  
  
CREATE TABLE PUBLISHER (  
  
ID_PUBLISHER T_ID NOT NULL,  
  
NAME_PUBLISHER T_NAME_PUBLISHER  
  
);  
  
CREATE TABLE READER (  
  
READER_ID T_ID NOT NULL,  
  
LASTNAME T_LASTNAME,  
  
FIRSTNAME T_FIRSTNAME,  
  
MIDNAME T_MIDNAME,  
  
TICKET T_TICKET,  
  
DAYOF T_DATE,  
  
CLASS T_CLASS  
  
);  
  
/* Первичные ключи */  
  
ALTER TABLE BOOK ADD CONSTRAINT PK_BOOK
```

```
PRIMARY KEY (BOOK_ID);
```

```
ALTER TABLE BOOKMOVE ADD CONSTRAINT PK_BOOKMOVE
```

```
PRIMARY KEY (BOOKMOVE_ID);
```

```
ALTER TABLE PUBLISHER ADD CONSTRAINT PK_PUBLISHER
```

```
PRIMARY KEY (ID_PUBLISHER);
```

```
ALTER TABLE READER ADD CONSTRAINT PK_READER
```

```
PRIMARY KEY (READER_ID);
```

```
/* Внешние ключи */
```

```
ALTER TABLE BOOKMOVE ADD CONSTRAINT FK_BOOKMOVE_1 FOREIGN KEY (BOOK_ID)  
REFERENCES BOOK (BOOK_ID)
```

```
ON UPDATE CASCADE;
```

```
ALTER TABLE READING ADD CONSTRAINT FK_READING_1 FOREIGN KEY (READER_ID)  
REFERENCES READER (READER_ID)
```

```
ON UPDATE CASCADE;
```

```
ALTER TABLE READING ADD CONSTRAINT FK_READING_2 FOREIGN KEY (BOOK_ID)  
REFERENCES BOOK (BOOK_ID)
```

```
ON UPDATE CASCADE;
```

```
/* Триггеры */
```

```
/* Trigger: BOOKMOVE_BI */
```

```
CREATE TRIGGER BOOKMOVE_BI FOR BOOKMOVE
```

```
ACTIVE BEFORE INSERT POSITION 0
```

```
as
```

```
begin
```

```
if (new.bookmove_id is null) then
```

```
new.bookmove_id = gen_id(gen_bookmove_id,1);
```

```
end;
```

```
/* Trigger: BOOK_BI */
```

```
CREATE TRIGGER BOOK_BI FOR BOOK
```

```
ACTIVE BEFORE INSERT POSITION 0
```

```
as
```

```
begin
```

```
if (new.book_id is null) then
```

```
new.book_id = gen_id(gen_book_id,1);
```

```
end;
```

```
/* Trigger: PUBLISHER_BI */
```

```
CREATE TRIGGER PUBLISHER_BI FOR PUBLISHER
```

```
ACTIVE BEFORE INSERT POSITION 0
```

```
As begin
```

```
if (new.id_publisher is null) then
```

```
new.id_publisher = gen_id(gen_publisher_id,1);
```

```
end;
```

```
/* Trigger: READER_BI */
```

```
CREATE TRIGGER READER_BI FOR READER
```

```
ACTIVE BEFORE INSERT POSITION 0
```

```
as
```

```
begin
```

```
if (new.reader_id is null) then
```

```
new.reader_id = gen_id(gen_reader_id,1);
```

```
end;
```

```
/* Хранимые процедуры */
```

```
CREATE OR ALTER PROCEDURE BOOK_DEL (
```

```
BOOK_ID TYPE OF COLUMN BOOK.BOOK_ID)
```

```
AS
```

```
begin
```

```
delete from book
```

```
where (book_id = :book_id);
```

```
end;
```

```
CREATE OR ALTER PROCEDURE BOOK_INS (
```

```
NAME TYPE OF COLUMN BOOK.NAME,
```

```
AUTHOR TYPE OF COLUMN BOOK.AUTHOR,
```

```
ID_PUBLISHER TYPE OF COLUMN BOOK.ID_PUBLISHER,
```

```
YEAROF TYPE OF COLUMN BOOK.YEAROF,
```

```
KIND TYPE OF COLUMN BOOK.KIND,
```

```
ISBN TYPE OF COLUMN BOOK.ISBN,
```

```
AMOUNT TYPE OF COLUMN BOOK.AMOUNT,
```

```
NUMBERS TYPE OF COLUMN BOOK.NUMBERS)
```

```
RETURNS (
```

```
ID TYPE OF COLUMN BOOK.BOOK_ID)
```

```
AS
```

```
begin
```

```
insert into book (  
name,  
author,  
id_publisher,  
yearof,  
kind,  
isbn,  
amount,  
numbers)  
values (  
:name,  
:author,  
:id_publisher,  
:yearof,  
:kind,  
:isbn,  
:amount,  
:numbers)  
returning book.book_id into :id;  
suspend;  
end;  
CREATE OR ALTER PROCEDURE BOOK_SEL  
RETURNS (
```

BOOK_ID TYPE OF COLUMN BOOK.BOOK_ID,
NAME TYPE OF COLUMN BOOK.NAME,
AUTHOR TYPE OF COLUMN BOOK.AUTHOR,
ID_PUBLISHER TYPE OF COLUMN BOOK.ID_PUBLISHER,
YEAROF TYPE OF COLUMN BOOK.YEAROF,
KIND TYPE OF COLUMN BOOK.KIND,
ISBN TYPE OF COLUMN BOOK.ISBN,
AMOUNT TYPE OF COLUMN BOOK.AMOUNT,
NUMBERS TYPE OF COLUMN BOOK.NUMBERS)

AS

begin

for select book_id,

name,

author,

id_publisher,

yearof,

kind,

isbn,

amount,

numbers

from book

into :book_id,

:name,

:author,

:id_publisher,

:yearof,

:kind,

:isbn,

:amount,

:numbers

do

begin

suspend;

end

end;

CREATE OR ALTER PROCEDURE BOOK_UPD (

BOOK_ID TYPE OF COLUMN BOOK.BOOK_ID,

NAME TYPE OF COLUMN BOOK.NAME,

AUTHOR TYPE OF COLUMN BOOK.AUTHOR,

ID_PUBLISHER TYPE OF COLUMN BOOK.ID_PUBLISHER,

YEAROF TYPE OF COLUMN BOOK.YEAROF,

KIND TYPE OF COLUMN BOOK.KIND,

ISBN TYPE OF COLUMN BOOK.ISBN,

AMOUNT TYPE OF COLUMN BOOK.AMOUNT,

NUMBERS TYPE OF COLUMN BOOK.NUMBERS)

AS

```
begin

update book

set name = :name,

author = :author,

id_publisher = :id_publisher,

yearof = :yearof,

kind = :kind,

isbn = :isbn,

amount = :amount,

numbers = :numbers

where (book_id = :book_id);

end;

CREATE OR ALTER PROCEDURE BOOKMOVE_DEL (

BOOKMOVE_ID TYPE OF COLUMN BOOKMOVE.BOOKMOVE_ID)

AS

begin

delete from bookmove

where (bookmove_id = :bookmove_id);

end;

ALTER PROCEDURE BOOKMOVE_INS (

BOOK_ID TYPE OF COLUMN BOOKMOVE.BOOK_ID,

DAYOF TYPE OF COLUMN BOOKMOVE.DAYOF,

DOC TYPE OF COLUMN BOOKMOVE.DOC,
```

```
DIR TYPE OF COLUMN BOOKMOVE.DIR,  
AMOUNT TYPE OF COLUMN BOOKMOVE.AMOUNT,  
NUMBERS TYPE OF COLUMN BOOKMOVE.NUMBERS)  
RETURNS (  
BOOKMOVE_ID TYPE OF COLUMN BOOKMOVE.BOOKMOVE_ID)  
AS  
begin  
insert into bookmove (  
book_id,  
dayof,  
doc,  
dir,  
amount,  
numbers)  
values (  
:book_id,  
:dayof,  
:doc,  
:dir,  
:amount,  
:numbers)  
returning bookmove.bookmove_id into :bookmove_id;  
suspend;
```

```
end;

CREATE OR ALTER PROCEDURE BOOKMOVE_SEL

RETURNS (

BOOKMOVE_ID TYPE OF COLUMN BOOKMOVE.BOOKMOVE_ID,

BOOK_ID TYPE OF COLUMN BOOKMOVE.BOOK_ID,

DAYOF TYPE OF COLUMN BOOKMOVE.DAYOF,

DOC TYPE OF COLUMN BOOKMOVE.DOC,

DIR TYPE OF COLUMN BOOKMOVE.DIR,

AMOUNT TYPE OF COLUMN BOOKMOVE.AMOUNT,

NUMBERS TYPE OF COLUMN BOOKMOVE.NUMBERS)

AS

begin

for select bookmove_id,

book_id,

dayof,

doc,

dir,

amount,

numbers

from bookmove

into :bookmove_id,

:book_id,

:dayof,
```

:doc,

:dir,

:amount,

:numbers

do

begin

suspend;

end

end;

```
CREATE OR ALTER PROCEDURE BOOKMOVE_UPD (  
    BOOKMOVE_ID TYPE OF COLUMN BOOKMOVE.BOOKMOVE_ID,  
    BOOK_ID TYPE OF COLUMN BOOKMOVE.BOOK_ID,  
    DAYOF TYPE OF COLUMN BOOKMOVE.DAYOF,  
    DOC TYPE OF COLUMN BOOKMOVE.DOC,  
    DIR TYPE OF COLUMN BOOKMOVE.DIR,  
    AMOUNT TYPE OF COLUMN BOOKMOVE.AMOUNT,  
    NUMBERS TYPE OF COLUMN BOOKMOVE.NUMBERS)
```

AS

begin

update bookmove

set book_id = :book_id,

dayof = :dayof,

doc = :doc,

dir = :dir,

amount = :amount,

numbers = :numbers

where (bookmove_id = :bookmove_id);

end;

CREATE OR ALTER PROCEDURE PUBLISHER_DEL (

ID_PUBLISHER TYPE OF COLUMN PUBLISHER.ID_PUBLISHER)

AS

begin

delete from publisher

where (id_publisher = :id_publisher);

end;

CREATE OR ALTER PROCEDURE PUBLISHER_INS (

NAME_PUBLISHER TYPE OF COLUMN PUBLISHER.NAME_PUBLISHER)

RETURNS (

ID TYPE OF COLUMN PUBLISHER.ID_PUBLISHER)

AS

begin

insert into publisher (

name_publisher)

values (

:name_publisher)

returning id_publisher into :id;

suspend;

end;

CREATE OR ALTER PROCEDURE PUBLISHER_SEL

RETURNS (

ID_PUBLISHER TYPE OF COLUMN PUBLISHER.ID_PUBLISHER,

NAME_PUBLISHER TYPE OF COLUMN PUBLISHER.NAME_PUBLISHER)

AS

begin

for select id_publisher,

name_publisher

from publisher

into :id_publisher,

:name_publisher

do

begin

suspend;

end

end;

CREATE OR ALTER PROCEDURE PUBLISHER_UPD (

ID_PUBLISHER TYPE OF COLUMN PUBLISHER.ID_PUBLISHER,

NAME_PUBLISHER TYPE OF COLUMN PUBLISHER.NAME_PUBLISHER)

AS

begin

```
update publisher

set name_publisher = :name_publisher

where (id_publisher = :id_publisher);

end;

CREATE OR ALTER PROCEDURE READER_DEL (

READER_ID TYPE OF COLUMN READER.READER_ID)

AS

begin

delete from reader

where (reader_id = :reader_id);

end;

CREATE OR ALTER PROCEDURE READER_INS (

LASTNAME TYPE OF COLUMN READER.LASTNAME,

FIRSTNAME TYPE OF COLUMN READER.FIRSTNAME,

MIDNAME TYPE OF COLUMN READER.MIDNAME,

TICKET TYPE OF COLUMN READER.TICKET,

DAYOF TYPE OF COLUMN READER.DAYOF,

CLASS TYPE OF COLUMN READER.CLASS)

RETURNS (

ID TYPE OF COLUMN READER.READER_ID)

AS

begin

insert into reader (
```

lastname,

firstname,

midname,

ticket,

dayof,

class)

values (

:lastname,

:firstname,

:midname,

:ticket,

:dayof,

:class)

returning reader.reader_id into :id;

suspend;

end;

CREATE OR ALTER PROCEDURE READER_SEL

RETURNS (

READER_ID TYPE OF COLUMN READER.READER_ID,

LASTNAME TYPE OF COLUMN READER.LASTNAME,

FIRSTNAME TYPE OF COLUMN READER.FIRSTNAME,

MIDNAME TYPE OF COLUMN READER.MIDNAME,

TICKET TYPE OF COLUMN READER.TICKET,

```
DAYOF TYPE OF COLUMN READER.DAYOF,  
CLASS TYPE OF COLUMN READER.CLASS)  
AS  
begin  
for select reader_id,  
lastname,  
firstname,  
midname,  
ticket,  
dayof,  
class  
from reader  
into :reader_id,  
:lastname,  
:firstname,  
:midname,  
:ticket,  
:dayof,  
:class  
do  
begin  
suspend;  
end
```

```
end;

CREATE OR ALTER PROCEDURE READER_UPD (
  READER_ID TYPE OF COLUMN READER.READER_ID,
  LASTNAME TYPE OF COLUMN READER.LASTNAME,
  FIRSTNAME TYPE OF COLUMN READER.FIRSTNAME,
  MIDNAME TYPE OF COLUMN READER.MIDNAME,
  TICKET TYPE OF COLUMN READER.TICKET,
  DAYOF TYPE OF COLUMN READER.DAYOF,
  CLASS TYPE OF COLUMN READER.CLASS)
AS
begin
  update reader
  set lastname = :lastname,
  firstname = :firstname,
  midname = :midname,
  ticket = :ticket,
  dayof = :dayof,
  class = :class
  where (reader_id = :reader_id);
end;

CREATE OR ALTER PROCEDURE SP_GEN_BOOKMOVE_ID
  RETURNS (
  ID INTEGER)
```

```
AS  
  
begin  
  
id = gen_id(gen_bookmove_id, 1);  
  
suspend;  
  
end;  
  
/* Описания */  
  
DESCRIBE TABLE BOOKMOVE 'Поступление/выбытие';  
  
DESCRIBE FIELD ID_PUBLISHER TABLE PUBLISHER 'ID Издательства';  
  
DESCRIBE FIELD NAME_PUBLISHER TABLE PUBLISHER 'Название издательства';
```

Приложение 6
Клиентская часть комплекса

Файл pLibrary.dpr

```
program pLibrary;  
  
uses  
  
Forms,  
  
frmLoginUnit in 'frmLoginUnit.pas' {frmLogin},  
  
uMain in 'uMain.pas' {fMain},  
  
uDm in 'uDm.pas' {dm: TDataModule},  
  
frmAboutUnit in 'frmAboutUnit.pas' {frmAbout},  
  
uCat in 'uCat.pas' {fCat},  
  
uReader in 'uReader.pas' {fReader},  
  
uReaderEd in 'uReaderEd.pas' {fReaderEd},  
  
uPublisher in 'uPublisher.pas' {fPublisher},
```

```
uPublisherEd in 'uPublisherEd.pas' {fPublisherEd},
uBook in 'uBook.pas' {fBook},
uBookEd in 'uBookEd.pas' {fBookEd};

{$R *.res}

begin

Application.Initialize;

Application.CreateForm(Tdm, dm);

if not dm.ReConnectToDB then

Application.Terminate

else begin

Application.CreateForm(TfMain, fMain);

Application.CreateForm(TfPublisher, fPublisher);

Application.CreateForm(TfReader, fReader);

Application.CreateForm(TfCat, fCat);

Application.CreateForm(TfPublisherEd, fPublisherEd);

Application.CreateForm(TfReaderEd, fReaderEd);

Application.CreateForm(TfBookEd, fBookEd);

Application.Run;

end;

end.

unit uReader;

interface

uses
```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, Buttons, StdCtrls, ExtCtrls, DB, DBTables, Menus,
IBCustomDataSet, IBStoredProc, Mask;

type

TfReader = class(TForm)

Panel1: TPanel;

btEdit: TButton;

btNew: TButton;

btDelete: TButton;

DBGrid1: TDBGrid;

ds: TDataSource;

sp: TIBStoredProc;

btFind: TButton;

Panel2: TPanel;

Label1: TLabel;

MaskEdit1: TMaskEdit;

q: TIBDataSet;

qREADER_ID: TIntegerField;

qLASTNAME: TIBStringField;

qFIRSTNAME: TIBStringField;

qMIDNAME: TIBStringField;

qFIO: TIBStringField;

qTICKET: TIBStringField;

```
qDAYOF: TDateField;

qCLASS: TIBStringField;

procedure FormActivate(Sender: TObject);

procedure btEditClick(Sender: TObject);

procedure btDeleteClick(Sender: TObject);

procedure btNewClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure btFindClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

fReader: TfReader;

implementation

uses uDm, uReaderEd;

{$R *.dfm}

procedure TfReader.FormActivate(Sender: TObject);

// При активации формы

begin

dbGrid1.ReadOnly := true;
```

```
if ds.DataSet.Active then ds.DataSet.Close;

ds.DataSet.Open;

end;

procedure TfReader.btEditClick(Sender: TObject);

// Вызов формы редактирования и корректировка строки

var

id: integer;

begin

id := ds.DataSet.FieldByName( 'Reader_ID' ).AsInteger;

fReaderEd.LastName := ds.DataSet.FieldByName( 'LastName' ).AsString;

fReaderEd.FirstName := ds.DataSet.FieldByName( 'FirstName' ).AsString;

fReaderEd.MidName := ds.DataSet.FieldByName( 'MidName' ).AsString;

fReaderEd.ticket := ds.DataSet.FieldByName( 'ticket' ).AsString;

fReaderEd.dayOf := ds.DataSet.FieldByName( 'dayOf' ).AsDateTime;

fReaderEd.xClass := ds.DataSet.FieldByName( 'Class' ).AsString;

if fReaderEd.ShowModal = mrOK then begin

try

with sp do begin

if Active then Close;

StoredProcName := UpperCase('Reader_UPD');

with Params do begin

Clear;

CreateParam( ftInteger,'Reader_ID',ptInput ).AsInteger := ds.DataSet['Reader_ID'];
```

```
CreateParam( ftString, 'LastName', ptInput ).AsString := fReaderEd.LastName;
CreateParam( ftString, 'FirstName', ptInput ).AsString := fReaderEd.FirstName;
CreateParam( ftString, 'MidName', ptInput ).AsString := fReaderEd.MidName;
CreateParam( ftString, 'ticket', ptInput ).AsString := fReaderEd.ticket;
CreateParam( ftString, 'dayof', ptInput ).AsDate := fReaderEd.dayOf;
CreateParam( ftString, 'class', ptInput ).AsString := fReaderEd.xClass;

end; // with Params

Prepare;

ExecProc;

dm.trans.Commit;

end; // with sp

if not ds.DataSet.Active then

ds.DataSet.Open;

ds.DataSet.Locate('Reader_ID', id, [] );

except

ShowMessage('Ошибка редактирования!');

end;

end;

end;

procedure TfReader.btDeleteClick(Sender: TObject);

// Удаление строки

begin

if MessageDlg('Удалить', mtConfirmation, mbYesNoCancel, 0) <> mrYes then
```

```
Abort
else
with sp do begin
if Active then Close;
StoredProcName:= UpperCase('Reader_DEL');
with Params do begin
Clear;
CreateParam(ftInteger, 'Reader_ID', ptInput).AsInteger := ds.DataSet['Reader_ID'];
end; // with Params
try
Prepare;
ExecProc;
ds.DataSet.Close;
ds.DataSet.Open;
except
ShowMessage('Удаление запрещено!');
end; // with sp
end; // if
end;
procedure TfReader.btNewClick(Sender: TObject);
// Вызов формы редактирования и вставка строки
var
id: integer;
```

```
begin
fReaderEd.name := '';
if fReaderEd.ShowModal = mrOK then begin
try
with sp do begin
if Active then Close;
StoredProcName := UpperCase('Reader_INS');
with Params do begin
Clear;
CreateParam( ftString, 'LastName', ptInput ).AsString := fReaderEd.LastName;
CreateParam( ftString, 'FirstName', ptInput ).AsString := fReaderEd.FirstName;
CreateParam( ftString, 'MidName', ptInput ).AsString := fReaderEd.MidName;
CreateParam( ftString, 'ticket', ptInput ).AsString := fReaderEd.ticket;
CreateParam( ftString, 'dayof', ptInput ).AsDate := fReaderEd.dayOf;
CreateParam( ftString, 'class', ptInput ).AsString := fReaderEd.xClass;
CreateParam( ftInteger, 'Reader_ID', ptOutput ).AsInteger;
end; // with Params
Prepare;
ExecProc;
dm.trans.Commit;
id := sp.ParamByName( 'Reader_ID' ).AsInteger;
end; // with sp
ds.DataSet.Close;
```

```
ds.DataSet.Open;

ds.DataSet.Locate('Reader_ID', id, [] );

except

ShowMessage('Ошибка добавления!');

end;

end;

end;

procedure TfReader.FormClose(Sender: TObject; var Action: TCloseAction);

// При закрытии формы

begin

ds.DataSet.Close;

end;

procedure TfReader.btFindClick(Sender: TObject);

// Поиск по названию

begin

if ( MaskEdit1.Text <> '' ) then

begin

if not ds.DataSet.Locate('Lastname', MaskEdit1.Text, [loPartialKey]) then

ShowMessage('Не найдено');

end;

end;

end;

procedure TfReader.FormCreate(Sender: TObject);

// При создании формы
```

```
begin
with q do begin
FieldByName( 'FIO').DisplayLabel := 'ФИО';
FieldByName( 'ticket').DisplayLabel := 'Билет';
FieldByName( 'dayOf').DisplayLabel := 'Дата';
FieldByName( 'Class').DisplayLabel := 'Класс';
end;
end;
end.
```

unit uReaderEd;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
Dialogs, StdCtrls, Mask, Buttons, ExtCtrls, DBCtrls, DB, DBTables,
```

```
ComCtrls;
```

```
type
```

```
TfReaderEd = class(TForm)
```

```
bbOk: TBitBtn;
```

```
MaskEdit1: TMaskEdit;
```

```
BitBtn1: TBitBtn;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

MaskEdit2: TMaskEdit;

MaskEdit3: TMaskEdit;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

MaskEdit4: TMaskEdit;

DateTimePicker1: TDateTimePicker;

MaskEdit5: TMaskEdit;

procedure FormActivate(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

LastName: string;

FirstName: string;

MidName: string;

ticket: string;

dayOf: TDateTime;

xClass: string;

end;

var

fReaderEd: TfReaderEd;

implementation

uses uDm;

{ \$R *.dfm }

procedure TfReaderEd.FormActivate(Sender: TObject);

// При активации формы

begin

MaskEdit1.Text := LastName;

MaskEdit2.Text := FirstName;

MaskEdit3.Text := MidName;

MaskEdit4.Text := ticket;

dateTimePicker1.DateTime := DayOf;

MaskEdit5.Text := xClass;

MaskEdit1.SetFocus;

end;

procedure TfReaderEd.FormClose(Sender: TObject;

var Action: TCloseAction);

// При закрытии формы

begin

LastName := MaskEdit1.Text;

FirstName := MaskEdit2.Text;

MidName := MaskEdit3.Text;

ticket := MaskEdit4.Text;

DayOf := dateTimePicker1.DateTime;

```
xClass := MaskEdit5.Text;
```

```
end;
```

```
end.
```