# МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное бюджетное учреждение

"Слободской колледж педагогики и социальных отношений"

## ДИПЛОМНЫЙ ПРОЕКТ

# Разработка программного обеспечения для комплексной автоматизации библиотеки

И

		Збруев Антон Владимирович	
		Специальность 09.02.07	
		Информационные программирование	системы
		Группа 21П-1	
		Форма обучения: очная	
		Руководитель:	
		Пентин Николай Се	ергеевич
Дипломный проект	защищен		
""	2025 г.		
Оценка			
Секретарь ГЭК			

Слободской

## ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ	5
1.1 Анализ предметной области	5
1.2. Техническое задание	12
ВЫВОД ПО ГЛАВЕ 1	16
ГЛАВА 2. КОНСТРУКТОРСКАЯ ЧАСТЬ	17
2.1. Архитектура программы	17
2.2. Описание алгоритмов и функционирования программы	20
ВЫВОДЫ ПО ГЛАВЕ 2	24
ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНО-ПРИКЛАДНАЯ ЧАСТЬ	25
3.1. Тестирование и опытная эксплуатация программы	25
3.2. Руководство пользователя	34
ВЫВОДЫ ПО ГЛАВЕ 3	41
ЗАКЛЮЧЕНИЕ	42
СПИСОК ЛИТЕРАТУРЫ	44
ПРИЛОЖЕНИЯ	47
Приложение 1	48
Приложение ?	10

#### **ВВЕДЕНИЕ**

В условиях современного информационного общества библиотеки играют важную роль, предоставляя доступ к знаниям и культурному наследию. Однако эффективное выполнение этой роли требует постоянной модернизации и оптимизации библиотечных процессов. Традиционные методы учета книжного фонда, выдачи и возврата литературы, регистрации читателей и других операций становятся все менее эффективными в условиях растущих информационных потоков и изменяющихся потребностей пользователей. Внедрение современных информационных систем позволяет библиотекам решать эти задачи, обеспечивая более высокий уровень обслуживания, сокращение трудозатрат и эффективное управление ресурсами.

Актуальность разработки информационных систем для библиотек подтверждается стремлением превратить их в интерактивные информационно-культурные центры. Автоматизация позволяет библиотекам расширить спектр предоставляемых услуг, упростить доступ к информации, интегрироваться с электронными ресурсами и повысить общую эффективность своей деятельности.

Платформа «1С:Предприятие 8.5» является наиболее одним ИЗ подходящих инструментов для разработки библиотечных ИС. Ее гибкость, масштабируемость, широкие интеграционные возможности и наличие развитой инфраструктуры поддержки позволяют создавать системы, максимально удовлетворяющие специфическим требованиям библиотек различного профиля масштаба. Доступность квалифицированных специалистов И консультационных услуг в области «1С:Предприятие» минимизирует риски и затраты на внедрение и сопровождение систем.

Цель данной дипломной работы – разработка информационной системы для автоматизации работы библиотеки на платформе «1С:Предприятие 8.5». Для достижения поставленной цели необходимо решить следующие задачи: провести анализ предметной области, разработать структуру базы данных и пользовательский интерфейс, реализовать функционал учета книжного фонда и

работы с читателями, провести тестирование и отладку системы, подготовить пользовательскую документацию.

Объектом исследования являются библиотечные процессы, а предметом — информационная система для их автоматизации. Методы исследования включают анализ предметной области, моделирование бизнес-процессов, объектно-ориентированное программирование и тестирование.

Практическая значимость дипломного проекта заключается в автоматизации ключевых библиотечных процессов, что позволит повысить эффективность работы библиотеки, улучшить качество обслуживания читателей и оптимизировать использование ресурсов.

Методологический аппарат исследования базируется на современных методологиях разработки программного обеспечения на платформе «1С:Предприятие», а также на методах анализа и проектирования информационных систем.

Дипломная работа состоит из введения, трех глав, заключения, списка литературы и приложений. Первая глава посвящена анализу предметной области и формулированию требований к системе. Во второй главе описана разработка информационной системы, ее структура и функциональные возможности. Третья глава содержит результаты тестирования и руководство пользователя.

В данном введении обоснована актуальность разработки ИС для библиотеки, определены цель, задачи, объект и предмет исследования, методы и методологический аппарат, а также отмечена практическая значимость проекта. Описание структуры работы завершает введение.

#### ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1 Анализ предметной области

Современная библиотека — это сложный организм, эффективное функционирование которого зависит от множества взаимосвязанных процессов. Для успешной работы библиотеки необходимо обеспечить точный учет книжного фонда, оперативный контроль за выдачей и возвратом литературы, а также вести базу данных читателей. Кроме того, важными аспектами деятельности являются закупка новых изданий, списание устаревших или поврежденных книг и прием пожертвований.

В данном разделе будет проведен детальный анализ предметной области, охватывающий ключевые библиотечные процессы и участников этих процессов. Это позволит выявить требования к разрабатываемой информационной системе и обеспечить ее максимальное соответствие потребностям библиотеки.

Основой любой библиотеки является ее книжный фонд. Эффективное управление этим фондом, как ключевым ресурсом библиотеки, невозможно без точного и подробного учета каждого экземпляра [12]. Разрабатываемая информационная система должна обеспечивать хранение и обработку всей необходимой информации о книгах, включая такие данные, как название, автор, жанр, издательство, год издания, ISBN, количество страниц и краткое описание содержания. Кроме того, система должна учитывать возрастной рейтинг книг, чтобы контролировать доступ к литературе для разных возрастных групп читателей. Информация о месте хранения каждой книги (зал, стеллаж, полка) позволит быстро находить нужные экземпляры. Важным аспектом учета является также статус книги, который может принимать значения "Есть на складе", "Отсутствует на складе", отражающие текущее состояние книги. Для удобства поиска и идентификации книг система должна предоставлять возможность добавления изображения обложки.

ISBN (International Standard Book Number) - это международный стандартный книжный номер, уникальный идентификатор для каждой книги. Он используется для упрощения учета и поиска книг в библиотеках, книжных магазинах и других организациях. Наличие ISBN в системе позволит быстро и точно идентифицировать книги, а также интегрировать систему с внешними базами данных и каталогами. Принципы использования стандартных идентификаторов, таких как ISBN, являются важной частью проектирования современных информационных систем [13].

Работа с читателями: для эффективного обслуживания читателей, система должна обеспечивать быструю и удобную регистрацию новых пользователей. При регистрации в системе фиксируются стандартные данные: фамилия, имя, отчество, дата рождения, полный почтовый адрес и контактная информация (телефон, электронная почта). Хранение этой информации позволяет библиотекарям легко идентифицировать читателей, связываться с ними по возникшим вопросам, а также вести статистику пользователей библиотеки. Кроме того, система должна предоставлять возможность просмотра истории взаимодействия с каждым читателем, включая информацию о выданных и возвращенных книгах, а также наличие задолженностей по штрафам. Разработка требований к хранению и обработке персональных данных пользователей является важным этапом создания любой информационной системы [14].

Выдача и возврат книг: центральным процессом в работе библиотеки является выдача и возврат книг. Система должна автоматизировать и упростить эти операции. При выдаче книги библиотекарь регистрирует в системе читателя, выбранную книгу, количество выдаваемых экземпляров, дату выдачи и расчетную дату возврата. При возврате книги фиксируется фактическая дата возврата и оценивается состояние возвращенной книги (на предмет повреждений). Автоматизация этого процесса позволяет контролировать соблюдение сроков возврата, вести точный учет книг, находящихся в обращении, и своевременно выявлять просроченную задолженность. При реализации таких операций в системах на платформе 1С:Предприятие

используются механизмы проведения документов и движения по регистрам накопления [5].

Продление книг: система должна предусматривать возможность продления срока выдачи книг по запросу читателя. При продлении в системе регистрируется новая дата планируемого возврата, что позволяет избежать начисления необоснованных штрафов.

Учет читательских билетов: читательские билеты являются необходимым условием для пользования библиотекой. Система должна обеспечивать регистрацию и учет читательских билетов, храня информацию о читателе, дате выдачи билета, выбранном абонементе, сроке действия абонемента, а также возрастном рейтинге, установленном для данного билета. Система должна поддерживать различные типы абонементов, отличающиеся стоимостью и сроком действия, что позволяет библиотеке предоставлять гибкие условия обслуживания для разных категорий читателей. Продление читательских билетов также должно отражаться в системе с фиксацией новой даты окончания срока действия.

Учет штрафов: система должна автоматически рассчитывать штрафы за просроченные книги на основании стоимости штрафа за день просрочки и количества просроченных дней. Должна быть предусмотрена возможность регистрации оплаты штрафов.

Закупка книг: процесс закупки новых книг должен быть автоматизирован. Система должна позволять создавать заказы на закупку, указывая поставщика (или издательство), список заказываемых книг и их количество. Необходимо отслеживать статус заказа (сформирован, оплачен, в пути, получен).

Списание книг: система должна позволять списывать книги с учетом причины списания (износ, повреждение, утрата).

Учет пожертвований: библиотеки часто получают книги в качестве пожертвований. Система должна предусматривать функционал для регистрации пожертвованных книг с указанием жертвователя и списка книг.

Отчетность: для анализа работы библиотеки и принятия управленческих решений система должна формировать различные отчеты. Например, отчеты по остаткам книг, популярности книг среди читателей, читателям с просроченными книгами, закупкам и списаниям.

Сотрудники, отделы и должности: для обеспечения бесперебойной работы библиотеки и корректного функционирования информационной системы, в библиотеке выделены следующие отделы и должности:

• Отдел обслуживания читателей: библиотекари этого отдела взаимодействуют с читателями, оформляют выдачу и возврат книг, проводят регистрацию новых читателей, оформляют и продлевают читательские билеты, а также взимают штрафы (Рисунок 1). В системе они работают с документами "ВыдачаКниг", "ВозвратКниг", "ЧитательскийБилет", "ПродлениеЧитательскогоБилета", "ОплатаШтрафов".

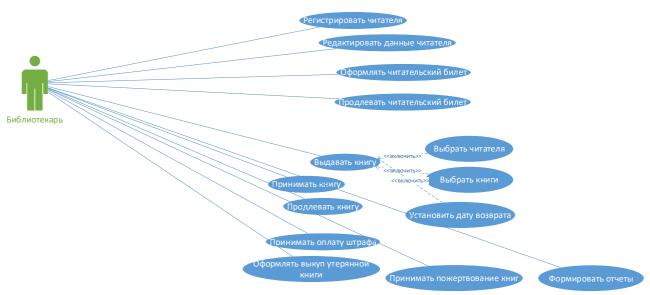


Рисунок 1 - Сценарии работы роли "библиотекарь" в ИС

• Отдел комплектования и обработки литературы: сотрудники этого отдела (библиографы, заведующий отделом) отвечают за формирование и обновление книжного фонда (Рисунок 2). Они проводят закупку новых книг, регистрируют их поступление в библиотеку, списывают устаревшую или поврежденную литературу, а также оформляют пожертвования. В системе они

используют документы "ЗакупкаКниг", "ПриходКниг", "СписаниеКниг", "Пожертвование".

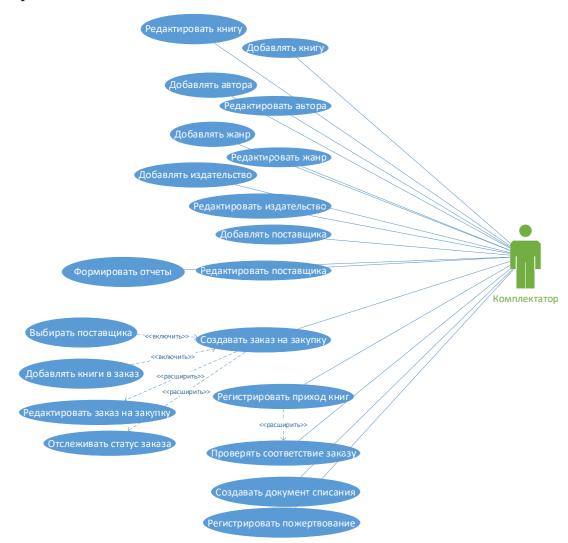


Рисунок 2 - Сценарии работы роли "комплектатор" в ИС

• Административно-управленческий отдел: руководители (директор, заместитель директора) и администраторы системы имеют полный доступ к функционалу ИС (Рисунок 3). Они отвечают за общее управление библиотекой, настройку системы, управление пользователями и их правами доступа, а также формирование отчетности. Руководители, в отличие от администраторов, могут быть ограничены в правах на удаление информации. Реализация ролевой модели и настройка прав доступа являются стандартными задачами при разработке конфигураций на платформе 1С:Предприятие [6].

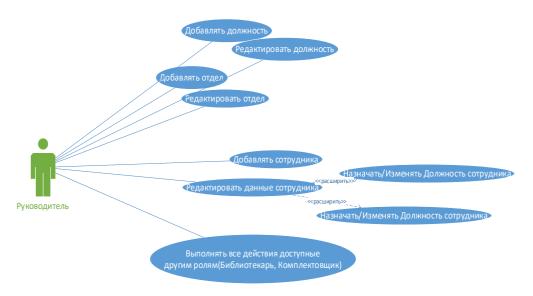


Рисунок 3 - Сценарии работы роли "Руководитель" в ИС

**Программы-аналоги:** перед разработкой собственной информационной системы важно проанализировать существующие на рынке решения, чтобы учесть их достоинства и недостатки. Анализ аналогов позволяет уточнить требования к собственной разработке и избежать повторения известных проблем [14]. Рассмотрим две программы-аналога от разных компаний:

## 1. АИБС "МАРК-SQL" (разработчик – компания "РУСЛАН"):

• Описание: Комплексная автоматизированная информационнобиблиотечная система, предназначенная для автоматизации всех основных библиотечных процессов: комплектование, каталогизация, читательское обслуживание, учет фондов и т.д.

#### • Плюсы:

- о Широкий функционал, покрывающий большинство библиотечных процессов.
  - о Возможность работы с большими объемами данных.
  - о Интеграция с различными внешними системами.

#### • Минусы:

- Высокая стоимость лицензий.
- о Сложность внедрения и настройки.
- о Требует высокой квалификации персонала.

## 2. АИБС "Ирбис64" (разработчик – ГПНТБ России):

• Описание: Автоматизированная информационно-библиотечная система, предоставляющая инструменты для автоматизации библиотечных процессов.

#### Плюсы:

- о Бесплатное распространение.
- о Открытый исходный код.
- о Большое сообщество пользователей.

## • Минусы:

- о Ограниченный функционал по сравнению с коммерческими АИБС.
- о Может требовать дополнительной доработки под специфические потребности библиотеки.
  - о Менее удобный интерфейс.

Вывод по анализу программ-аналогов: анализ программ-аналогов показал, что существующие решения либо обладают широким функционалом, но имеют высокую стоимость и сложность внедрения ("MAPK-SQL"), либо бесплатны и доступны, но имеют ограниченный функционал и требуют доработки ("Ирбис64"). Разрабатываемая в рамках данного дипломного проекта система на платформе "1С:Предприятие 8.5" направлена на то, чтобы сочетать в себе достоинства обоих подходов: предоставить необходимый функционал для автоматизации библиотечных процессов, обладая при этом гибкостью, масштабируемостью и относительно невысокой стоимостью внедрения и сопровождения благодаря использованию популярной платформы.

Общий вывод по анализу предметной области: проведенный анализ предметной области, включающий рассмотрение основных библиотечных процессов, ролей пользователей и доступных программных аналогов, позволил сформулировать четкие требования к разрабатываемой информационной системе. Система должна обеспечить автоматизацию учета книжного фонда, работы с читателями, закупок, списаний и других важных аспектов библиотечной деятельности. Полученные результаты будут использованы для проектирования базы данных, разработки пользовательского интерфейса и

реализации функциональных возможностей системы. Таким образом, этап анализа предметной области закладывает фундамент для последующих стадий разработки программного обеспечения в соответствии с процессами жизненного цикла [4]. Выбранная платформа "1С:Предприятие 8.5" позволит создать гибкое и масштабируемое решение, адаптированное к специфическим потребностям библиотеки, а так же имеет современный стиль, который является более удобным решением в сравнений с более старыми версиями.

## 1.2. Техническое задание

**Наименование программы:** "Автоматизированная библиотечная информационная система" (АБИС).

Заказчик: МастерСофт-Сервис.

**Назначение:** программа предназначена для автоматизации работы библиотеки, включая учет библиотечного фонда, обслуживание читателей, закупку и списание книг, учет пожертвований и формирование отчетов.

**Область применения:** программа может эксплуатироваться на ПК на любом предприятии, связанном с библиотечной деятельностью.

#### Функциональные требования.

#### 1. Учет библиотечного фонда.

- Добавление, редактирование и удаление записей о книгах, журналах, газетах и других изданиях с указанием названия, автора, жанра, издательства, года издания, ISBN, количества страниц, описания, возрастного рейтинга, статуса, места хранения и изображения обложки.
  - Каталогизация и классификация изданий.
- Учет поступлений (закупка, пожертвование) и выбытия (списание, утеря).
  - Инвентаризация фонда.
  - Поиск изданий по различным критериям.

## 2. Обслуживание читателей.

- Регистрация читателей с возможностью хранения полной информации о каждом читателе (ФИО, дата рождения, адрес, контакты, выданные книги и т.д.).
- Выдача и возврат книг с автоматическим учетом сроков и начислением штрафов за просрочку.
  - Продление сроков пользования.
  - Резервирование книг.
- Формирование читательских билетов с различными типами абонементов и возрастными рейтингами.
  - Учет и оплата штрафов.

## 3. Закупка книг.

- Формирование заказов на закупку книг у поставщиков и издательств.
- Учет статусов заказов (сформирован, оплачен, в пути, получен).
- Регистрация поступления книг по заказам.

#### 4. Списание книг.

• Регистрация списания книг с указанием причины.

## 5. Учет пожертвований.

• Регистрация пожертвованных книг с указанием жертвователя.

## 6. Формирование отчетов.

- Отчеты по библиотечному фонду (остатки, движение, популярность, анализ операций с книгами).
  - Отчеты по читателям (задолженности).
  - Отчеты по закупкам и списаниям.

## 7. Система разграничения прав доступа.

• Различные роли пользователей (администратор, комплектатор фондов, библиотекарь) с разными уровнями доступа к функциям системы.

## Требования к пользовательскому интерфейсу.

- Интуитивно понятный и удобный интерфейс.
- Наличие системы подсказок и помощи.
- Соответствие принципам эргономики.

## Требования к техническому обеспечению.

- Операционная система: Windows 10/11.
- Платформа разработки: 1С:Предприятие 8.5.
- СУБД: Использование внутренней СУБД у 1С:Предприятие 8.5.

## Требования к надежности и безопасности.

- Устойчивость к ошибкам пользователя.
- Резервное копирование данных.
- Авторизация и разграничение прав доступа.
- Защита от несанкционированного доступа.
- Журналирование действий пользователей.

## Стадии и этапы разработки.

Разработка АБИС будет осуществляться в несколько стадий, каждая из которых включает определенные этапы:

## 1. Анализ и проектирование.

- Анализ требований: сбор и анализ требований к системе, детальное изучение бизнес-процессов библиотеки, определение функциональных возможностей будущей системы. Результатом этого этапа является техническое задание.
- Проектирование базы данных: разработка структуры базы данных, определение сущностей, атрибутов и связей между ними. Создание логической и физической моделей базы данных.
- Проектирование пользовательского интерфейса: разработка макетов форм и отчетов, определение навигации и взаимодействия пользователя с системой.

## 2. Разработка:

- Разработка модулей системы: написание программного кода для реализации функциональных модулей системы (учет книг, работа с читателями, закупки, списания и т.д.).
- Создание форм и отчетов: реализация пользовательского интерфейса на основе разработанных макетов.

• **Настройка прав доступа:** настройка ролей пользователей и их прав доступа к различным функциям системы.

## 3. Тестирование:

- Модульное тестирование: проверка корректности работы отдельных модулей системы.
- Интеграционное тестирование: тестирование взаимодействия между модулями системы.
- Системное тестирование: проверка работы системы в целом на соответствие требованиям технического задания.
- Нагрузочное тестирование: оценка производительности системы при большой нагрузке.

#### 4. Внедрение:

- Установка и настройка системы: установка системы на сервере и рабочих станциях пользователей, настройка параметров подключения к базе данных.
- Обучение пользователей: проведение обучения пользователей работе с системой.
- Миграция данных: перенос данных из существующих систем в новую АБИС (при необходимости).

#### 5. Сопровождение:

- Исправление ошибок: оперативное исправление выявленных ошибок и недочетов.
- Доработка функционала: добавление нового функционала по запросу заказчика.
- Консультационная поддержка: предоставление консультаций пользователям по работе с системой.

#### Документация:

- Техническое задание.
- Руководство пользователя.

## ВЫВОД ПО ГЛАВЕ 1

В первой главе дипломного проекта проведен анализ предметной области, разработки, определяющей контекст автоматизированной библиотечной информационной системы (АБИС). Детально рассмотрены основные бизнеспроцессы библиотеки, включая учет книжного фонда, обслуживание читателей, процессы закупки и списания литературы, а также учет пожертвований. Особое библиотекарей, внимание уделено ролям пользователей системы комплектаторов и администраторов, – и их взаимодействию с АБИС в рамках выполняемых ими функций.

Анализ существующих программных аналогов, таких как АИБС "МАРК-SQL" и "Ирбис64", позволил выявить их сильные и слабые стороны, а также обосновать выбор платформы "1С:Предприятие 8.5" для разработки АБИС. Преимущества данной платформы, такие как гибкость, масштабируемость и наличие широкой сети специалистов, делают ее оптимальным решением для создания системы, удовлетворяющей специфическим потребностям библиотеки.

На основе проведенного анализа сформулированы функциональные, технические и пользовательские требования к разрабатываемой АБИС, которые зафиксированы в техническом задании. Это задание служит основой для дальнейших стадий проектирования и разработки системы, обеспечивая ее соответствие потребностям библиотеки и эффективность ее использования. Результаты анализа предметной области будут использованы на последующих этапах разработки для создания оптимальной архитектуры системы, проектирования базы данных и разработки удобного и функционального пользовательского интерфейса.

#### ГЛАВА 2. КОНСТРУКТОРСКАЯ ЧАСТЬ

## 2.1. Архитектура программы

Разработанная автоматизированная библиотечная информационная система (АБИС) представляет собой прикладное решение, созданное на технологической платформе "1С:Предприятие 8.5". В состав программного обеспечения входит основной файл конфигурации Библиотека.cf, который содержит все объекты метаданных, определяющие структуру данных и логику работы системы.

## Платформа и среда разработки:

Выбор платформы "1С:Предприятие 8.5" обусловлен ее широкими возможностями для быстрой разработки учетных систем, гибкостью настройки, масштабируемостью и наличием развитых механизмов интеграции. Платформа обеспечивает высокую производительность, надежность хранения данных и современные средства безопасности. Весь процесс разработки, включая создание объектов метаданных, написание программного кода и проектирование пользовательского интерфейса, осуществлялся в интегрированной среде разработки – Конфигураторе 1С.

## Архитектура:

АБИС реализована с использованием **клиент-серверной архитектуры**, которая является рекомендуемой для многопользовательских систем и обеспечивает оптимальное распределение нагрузки.

- Серверная часть: отвечает за основную бизнес-логику, обработку данных и взаимодействие с системой управления базами данных (СУБД). Сервер "1С:Предприятие" управляет подключениями клиентских приложений, выполняет серверные процедуры и функции, обеспечивает целостность данных при одновременной работе нескольких пользователей.
- Клиентская часть: представляет собой пользовательский интерфейс, с которым взаимодействуют сотрудники библиотеки. В данной

разработке используется тонкий клиент и веб-клиент, что позволяет пользователям работать с системой как через установленное приложение, так и через веб-браузер, обеспечивая гибкость доступа. Взаимодействие между клиентом и сервером происходит по протоколу TCP/IP.

- База данных: для хранения всей информации системы (данные справочников, документы, записи регистров) используется встроенная файловая СУБД платформы "1С:Предприятие". Это упрощает развертывание и администрирование системы для небольших и средних библиотек.
  - Основные компоненты системы (объекты метаданных):

структура данных и функциональность АБИС определяются следующими основными объектами метаданных:

- Справочники: используются для хранения условно-постоянной, нормативно-справочной информации. Ключевыми справочниками являются: "Книги" (основной каталог фонда), "Читатели", "Авторы", "Жанры", "Издательства", "Поставщики", "Сотрудники" (для ведения списка персонала), "Должности", "Отделы", "Статусы" (для книг), "Руководители". Иерархическая структура справочника "Книги" позволяет группировать издания по типам (Книги, Журналы, Периодические издания).
- Документы: предназначены для регистрации хозяйственных операций и событий в хронологическом порядке. Основные документы системы: "ПриходКниг", "СписаниеКниг", "ЗакупкаКниг", "Пожертвование" (учет движения фонда); "ВыдачаКниг", "ВозвратКниг", "ПродлениеКниги" (обслуживание читателей); "ЧитательскийБилет", "ПродлениеЧитательскогоБилета" (учет билетов); "ВыкупУтеряннойКниги", "ОплатаШтрафов" (финансовые операции с читателями).
- **Регистры сведений:** служат для хранения информации, характеризующей состояние объектов или взаимосвязи между ними на определенный момент времени или периодически. Используются регистры: "Цены" (возможно, для оценки книг), "ЦеныЧитательскихБилетов",

"ЦеныШтрафов", "СрокиДействияЧитательскихБилетов", "СрокВозвратаКниг", "ШтрафыЗаПросрочку".

- Регистры накопления: предназначены для хранения итоговой (агрегированной) информации о количественных или суммовых показателях в различных разрезах (измерениях). Ключевые регистры накопления: "ОстаткиКниг" (учет количества книг на складе), "ОборотЗаказовКниг" (отслеживание выполнения заказов), "ДоступныеКниги" (оперативный учет книг, доступных для выдачи), "ИсторияВыдачКниг" (отслеживание книг на руках у читателей), "ПопулярностьКниги" (сбор статистики для анализа спроса), "ОборотыПожертвований", "Взаиморасчеты" (если используются). Регистр "РегистрБухУчет" был предусмотрен, но его использование в рамках данной работы ограничено.
- Перечисления: используются для хранения фиксированных наборов значений, определяющих различные классификационные признаки (например, ИсточникПолучения, СтатусЗакупки, СостояниеКниги, ПолМЖ и т.д.).
- Обработки: предназначены для выполнения сервисных функций и пакетной обработки данных. В системе реализованы обработки: "Загрузить Авторов" (для импорта данных), "Установка Цен Билетов", "Виджеты" (для формирования данных начальной страницы), "Первоначальное Заполнение Фонда".
- Отчеты: служат для анализа накопленной информации и представления ее в удобном для пользователя виде. Основные отчеты: ОтчетПоОстаткамКниг, СоотношениеКниг, ЗаказКниг, КнигиНаРуках, ДанныеОВозвратахКниг, ДоступныеКнигиДляВыдачи, ПопулярностьКнигСредиЧитателей, ОтчетПоВыкупуУтерянныхКниг, ПродажиДиаграмма.

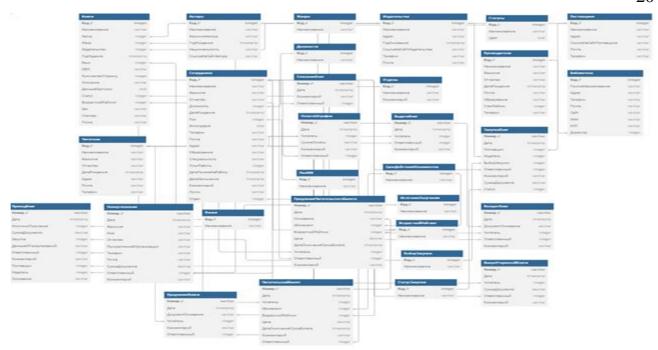


Рисунок 4 – Логическая структура базы данных (ЕR-диаграмма)

На рисунке 4 представлена ER-диаграмма, иллюстрирующая основные сущности базы данных и связи между ними, что дает наглядное представление о логической структуре хранения информации в АБИС.

Взаимодействие между компонентами системы осуществляется через вызовы процедур и функций общих модулей, а также через механизмы проведения документов, которые формируют записи в соответствующих регистрах. Пользовательский интерфейс построен на основе управляемых форм, обеспечивающих современный внешний вид и удобство работы.

Этот раздел описывает основные компоненты разработанной АБИС и используемые технологии. Более подробная информация о реализации отдельных объектов метаданных и алгоритмах их работы будет представлена в следующих разделах.

## 2.2. Описание алгоритмов и функционирования программы

## • Функционирование программы

При запуске разработанной "Автоматизированной библиотечной информационной системы" (АБИС) происходит подключение к созданной базе

данных на платформе "1С:Предприятие 8.5". Затем система запрашивает у пользователя данные для авторизации. Механизмы подключения к базе данных и аутентификации пользователей являются стандартными возможностями платформы 1С:Предприятие [16].

Рассмотрим основные алгоритмы, используемые в программе. Ниже приведено их краткое описание.

## • Запуск программы

При запуске АБИС происходит инициализация соединения с базой данных и отображение окна аутентификации.

## • Вход в программу

Вход осуществляется после ввода логина и пароля. Если введенные данные соответствуют зарегистрированному пользователю (сотруднику администратору), программа предоставляет доступ и открывает основной интерфейс (рабочий стол), соответствующий роли пользователя. Реализация разграничения доступа на основе ролей и управление пользовательскими интерфейсами выполняются  $\mathbf{c}$ использованием встроенных средств конфигуратора [8]. Если логин или пароль неверны, доступ запрещается до ввода корректных данных.

#### • Регистрация читателя и выдача билета

Библиотекарь переходит в подсистему управления читателями, создает новую запись в справочнике "Читатели", заполняя необходимые данные. Затем создается документ "Читательский Билет", где указывается читатель, выбирается абонемент и возрастной рейтинг. Система автоматически рассчитывает цену и дату окончания действия. Документ проводится, сохраняя информацию в регистре сведений. Для хранения атрибутов объектов, таких как данные читателей или параметры билетов, используются справочники и регистры сведений [5].

#### • Выдача книги

В подсистеме обслуживания читателей библиотекарь создает документ "ВыдачаКниг". Указывается читатель, в табличную часть добавляются книги,

устанавливается качество при выдаче и дата возврата. При проведении система проверяет наличие книги и действительность читательского билета. Данные о выдаче фиксируются в регистрах "ДоступныеКниги", "ИсторияВыдачКниг", "ПопулярностьКниги" и "СрокВозвратаКниг". Проведение документа "ВыдачаКниг" инициирует запись движений в соответствующие регистры накопления и сведений, отражая изменения в состоянии библиотечного фонда и истории взаимодействия с читателем [21].

## • Возврат книги

Библиотекарь создает документ "ВозвратКниг", часто на основании выдачи. Указывается качество книги при возврате. Система сравнивает состояния и информирует о расхождениях. При проведении обновляются регистры "ИсторияВыдачКниг", "ДоступныеКниги", "СрокВозвратаКниг". Если

возврат просрочен, рассчитывается и фиксируется штраф в регистре "ШтрафыЗаПросрочку". Алгоритмы проведения документа возврата включают логику контроля остатков по регистрам накопления и запись информации в периодические регистры сведений для отслеживания истории и начисления штрафов [22].

#### • Закупка и приход книг

Комплектатор создает документ "ЗакупкаКниг", указывая поставщика и заказываемые книги. Документ записывается, а при необходимости может формировать движения, например, для резервирования или планирования поступлений в регистре "ОборотЗаказовКниг" [9]. При фактическом поступлении книг создается документ "ПриходКниг" (возможно, на основании закупки). При проведении "ПриходКниг" книги поступают на учет в регистры "ОстаткиКниг" и "ДоступныеКниги", а также списывается информация из "ОборотЗаказовКниг".

## • Просмотр отчетов

Пользователи с соответствующими правами могут перейти в раздел отчетов и сформировать необходимый отчет (например, "ОтчетПоОстаткамКниг", "ПопулярностьКнигСредиЧитателей",

"ДанныеОВозвратахКниг"). Отчеты формируются на основе актуальных данных из регистров системы. Для формирования сложных аналитических отчетов в системе используется механизм Системы Компоновки Данных (СКД) [7].

## • Работа с виджетами (на основной форме)

При открытии основной формы автоматически загружаются данные для виджетов: информация о текущем пользователе, данные о библиотеке, список ожидаемых заказов и диаграмма популярности книг за выбранный период. Пользователь может изменять период для диаграммы популярности. Настройка начальной страницы и использование управляемых форм позволяют создавать удобные и информативные рабочие столы для пользователей [20]

#### • Входные данные

Входными данными программы является информация, вносимая пользователем с клавиатуры и путем выбора из справочников в поля документов и справочников. Эта информация хранится в базе данных в соответствующих таблицах (объектах метаданных). Структура хранения данных определяется объектами метаданных конфигурации, такими как справочники, документы и регистры [6].

#### • Выходные данные

Выходные данные выводятся в виде сформированных отчетов, информации в списках и формах документов и справочников, а также в виде печатных форм документов (например, акт пожертвования, регистрация книг, чеки об оплате). Формирование печатных форм осуществляется с использованием механизма макетов и языка запросов для выборки необходимых данных [24]. Отчеты и списки формируются на основе данных, хранящихся в регистрах и справочниках системы.

#### ВЫВОДЫ ПО ГЛАВЕ 2

Во второй главе дипломного проекта была рассмотрена конструкторская часть разработанной автоматизированной библиотечной информационной системы (АБИС).

Был проведен анализ и описание архитектуры программы. Система построена на платформе "1С:Предприятие 8.5" с использованием клиент-серверной архитектуры, что обеспечивает надежность, производительность и возможность масштабирования. Определен состав программного обеспечения, включая основную конфигурацию и потенциальные расширения. Перечислены ключевые компоненты системы: справочники, документы, регистры сведений и накопления, перечисления, обработки и отчеты, составляющие основу структуры данных и функциональности АБИС.

Далее было представлено описание основных алгоритмов и функционирования программы. Рассмотрены ключевые сценарии работы пользователей с системой, начиная от запуска и авторизации, до выполнения основных операций: регистрации читателей и выдачи билетов, выдачи и возврата книг, закупки и прихода литературы, просмотра отчетов и взаимодействия с информационными виджетами на основной форме. Описана логика проверок, выполняемых системой при проведении документов, и движений, формируемых в регистрах учета. Также определены основные входные и выходные данные программы.

Разработанная архитектура и реализованные алгоритмы обеспечивают выполнение всех функциональных требований, сформулированных в техническом задании, и создают основу для эффективной автоматизации библиотечных процессов. Структура системы является логичной и позволяет в дальнейшем развивать и модифицировать функционал АБИС.

## ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНО-ПРИКЛАДНАЯ ЧАСТЬ

## 3.1. Тестирование и опытная эксплуатация программы

Тестирование программного обеспечения является неотъемлемым этапом разработки, позволяющим выявить и устранить ошибки, а также убедиться в соответствии системы заявленным требованиям. В рамках дипломного проекта было проведено функциональное тестирование основных модулей АБИС.

## Подготовка к тестированию:

Для проведения тестирования была создана тестовая база данных, заполненная необходимыми начальными данными с помощью специально разработанных обработок. Были созданы записи в справочниках "Языки", "Жанры", "Авторы", "Издательства", "Читатели", "Сотрудники" и др. Также была имитирована начальная загрузка книжного фонда, выдача читательских билетов и выдача книг за предыдущий период.

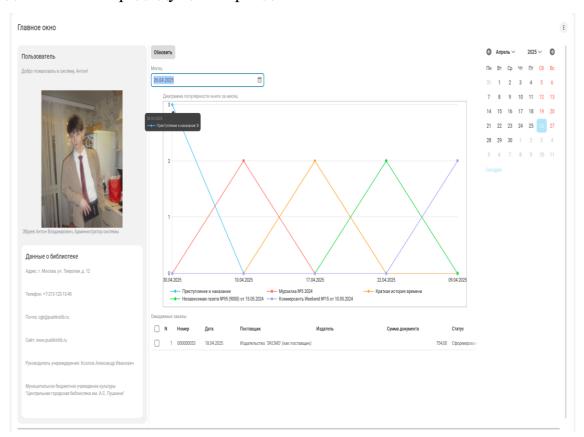


Рисунок 3.1 – Основной интерфейс АБИС

#### Тестирование основных сценариев:

Рассмотрим тестирование нескольких ключевых сценариев работы системы:

- Сценарий 1: Регистрация читателя и выдача читательского билета.
- о **Действия:** Библиотекарь входит в систему под своей учетной записью. Переходит в справочник "Читатели". Создает новую запись, заполняя ФИО, дату рождения и контактные данные.

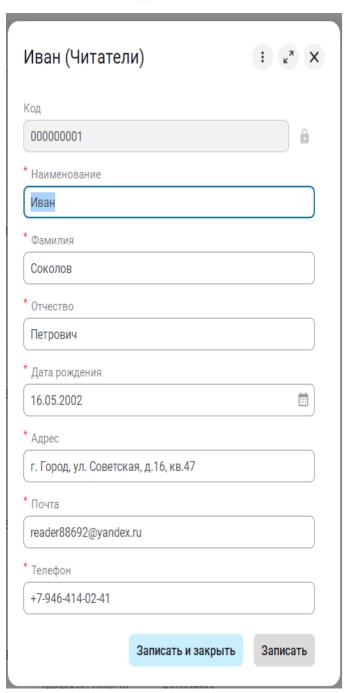
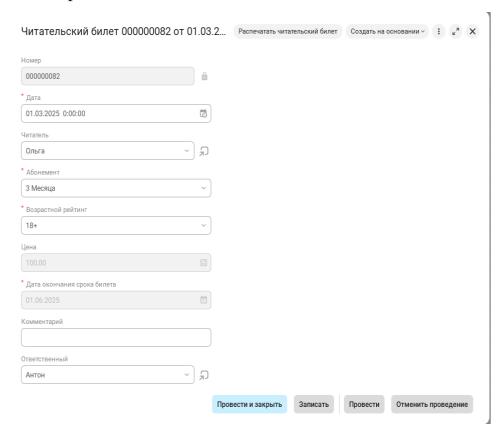


Рисунок 3.2 – Форма регистрации нового читателя

о Далее библиотекарь создает документ "Читательский Билет". Выбирает созданного читателя, указывает тип абонемента (например, "Месяц\_3") и возрастной рейтинг. Система автоматически рассчитывает цену и дату окончания срока действия билета.



3.3 – Оформление читательского билета

- о Документ проводится.
- о **Ожидаемый результат:** Запись о читателе успешно создана. Документ "Читательский Билет" проведен, информация о сроке действия билета записана в регистр сведений "Сроки Действия Читательских Билетов".
- **Результат тестирования:** Сценарий выполнен успешно, ошибок не выявлено.
  - Сценарий 2: Выдача книги читателю.
- о **Действия:** Библиотекарь создает документ "ВыдачаКниг". Выбирает читателя (у которого есть действующий билет). В табличную часть добавляет одну или несколько книг из доступных. Устанавливает качество книги при выдаче и дату возврата (не более месяца).

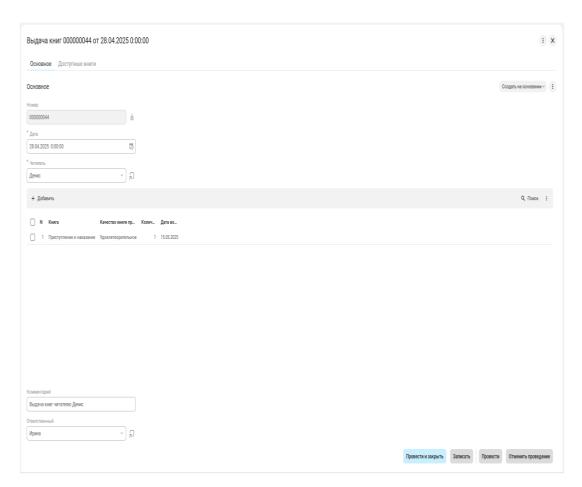


Рисунок 3.4 – Оформление выдачи книг

- о Документ проводится.
- о **Ожидаемый результат:** Система проверяет наличие книг и действительность билета. Документ успешно проводится. Формируются движения в регистрах "ДоступныеКниги" (расход), "ИсторияВыдачКниг" (приход), "ПопулярностьКниги" (приход), "СрокВозвратаКниг".
- **Результат тестирования:** Сценарий выполнен успешно. Проверка доступности книг и срока действия билета работает корректно. Движения по регистрам формируются правильно.
- Сценарий 3: Возврат книги с просрочкой и начислением штрафа.
- о Действия: Библиотекарь создает документ "ВозвратКниг" на основании ранее созданного документа "ВыдачаКниг". Дата документа устанавливается позже планируемой даты возврата. Указывается качество книги при возврате (отличное от качества при выдаче для проверки).

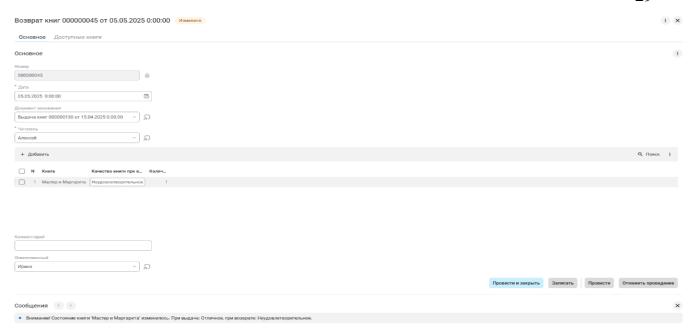


Рисунок 3.5 – Оформление возврата книг

- о При изменении качества система выдает предупреждающее сообщение.
  - о Документ проводится.
- о **Ожидаемый результат:** Документ проводится. Формируются движения в регистрах "ИсторияВыдачКниг" (расход), "ДоступныеКниги" (приход), "СрокВозвратаКниг" (запись фактической даты возврата). Автоматически рассчитывается и записывается штраф в регистр сведений "ШтрафыЗаПросрочку".
- **Результат тестирования:** Сценарий выполнен успешно. Сравнение качества книги при изменении в форме возврата работает. Штраф рассчитывается и регистрируется корректно при проведении.
  - Сценарий 4: Оплата штрафа.
- о Действия: Библиотекарь создает документ "ОплатаШтрафов". Выбирает читателя, имеющего неоплаченный штраф. Табличная часть автоматически заполняется неоплаченными штрафами этого читателя. Библиотекарь отмечает флажком штраф(ы) для оплаты. Сумма оплаты рассчитывается автоматически.

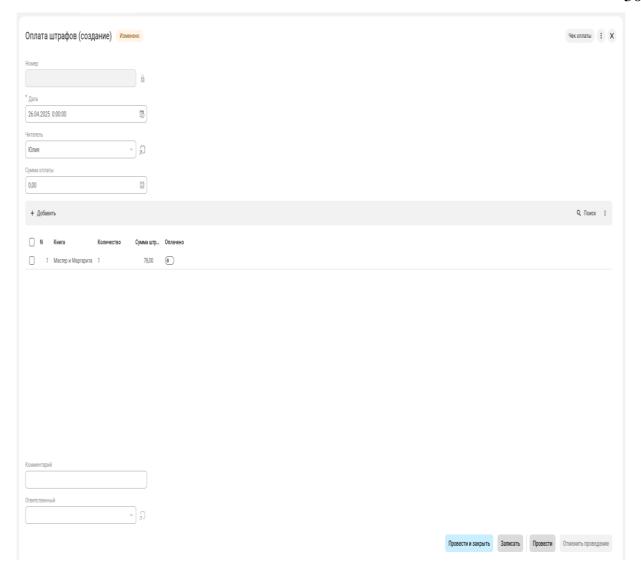


Рисунок 3.6 – Оформление оплаты штрафа

- о Документ проводится.
- о **Ожидаемый результат:** Документ проводится. В регистре сведений "ШтрафыЗаПросрочку" для оплаченного штрафа устанавливается флаг "Оплачено".
- **Результат тестирования:** Сценарий выполнен успешно. Статус штрафа обновляется корректно.
  - Сценарий 5: Закупка и приход книг.
- о **Действия:** Комплектатор создает документ "ЗакупкаКниг". Выбирает поставщика, заполняет табличную часть заказываемыми книгами, указывает количество и цену. Устанавливает статус "Согласовано". Документ записывается.

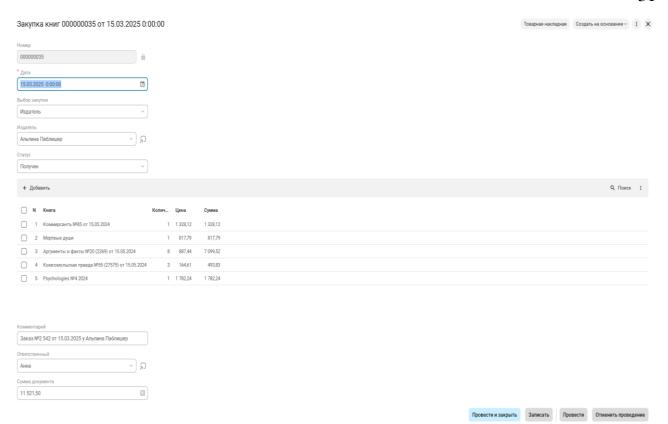


Рисунок 3.7 – Создание заказа на закупку

о После фактического получения книг комплектатор на основании документа "ЗакупкаКниг" создает документ "ПриходКниг". Данные автоматически переносятся. Пользователь проверяет и при необходимости корректирует количество.

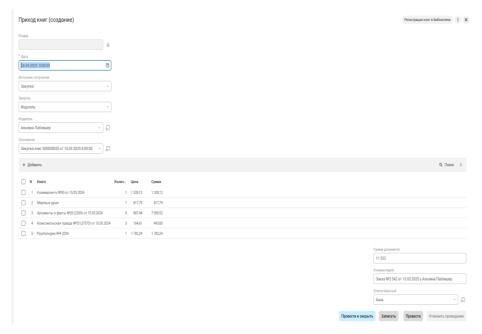


Рисунок 3.8 – Оформление прихода книг

- о Документ "ПриходКниг" проводится.
- о **Ожидаемый результат:** Документ "ЗакупкаКниг" записан. Документ "ПриходКниг" проведен. Книги оприходованы в регистрах "ОстаткиКниг" и "ДоступныеКниги". Информация о выполненном заказе отражена в регистре "ОборотЗаказовКниг" (расход по заказу).
- **Результат тестирования:** Сценарий выполнен успешно. Создание на основании работает, данные переносятся корректно. Движения по регистрам формируются правильно.
  - Сценарий 6: Списание книги.
- о **Действия:** Комплектатор создает документ "СписаниеКниг". В табличную часть добавляет книгу, подлежащую списанию, указывает количество и причину списания (например, "Износ").

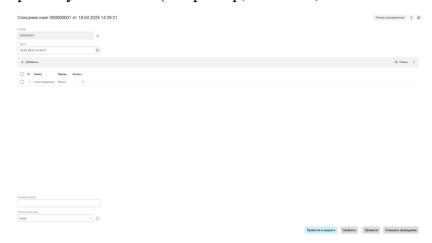


Рисунок 3.9 – Оформление списания книги

- о Документ проводится.
- о **Ожидаемый результат:** Документ проведен. Списанное количество отражено в регистрах "ОстаткиКниг" (расход) и "ДоступныеКниги" (расход).
- о **Результат тестирования:** Сценарий выполнен успешно. Движения по регистрам формируются корректно.
  - Сценарий 7: Продление книги.
- о **Действия:** Библиотекарь создает документ "ПродлениеКниги" на основании документа "ВыдачаКниг". В табличной части для нужной книги указывает новую дату сдачи (не превышающую допустимый срок продления).

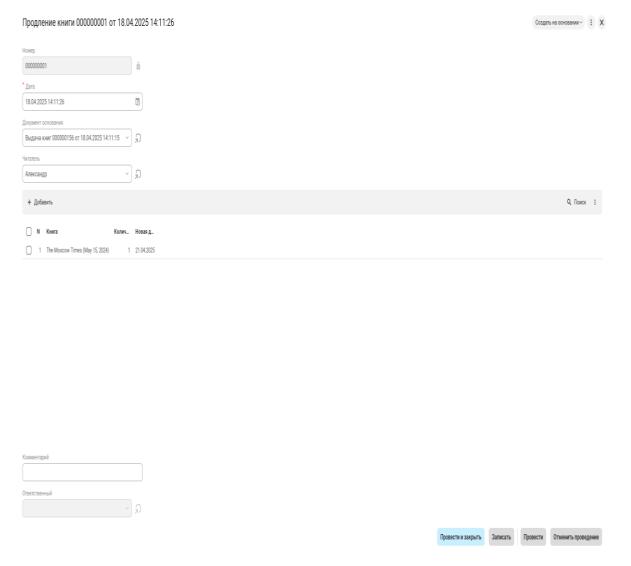


Рисунок 3.10 – Оформление продления книги

- о Документ проводится.
- о **Ожидаемый результат:** Документ проведен. В регистре сведений "СрокВозвратаКниг" создана новая запись с обновленной планируемой датой возврата.
- о **Результат тестирования:** Сценарий выполнен успешно. Информация в регистре обновляется правильно.
  - Сценарий 8: Формирование отчета "Популярность книг".
- Действия: Пользователь (например, руководитель ИЛИ Выбирает комплектатор) переходит В раздел отчетов. отчет "ПопулярностьКнигСредиЧитателей". Устанавливает период (например, прошлый месяц). Нажимает "Сформировать".

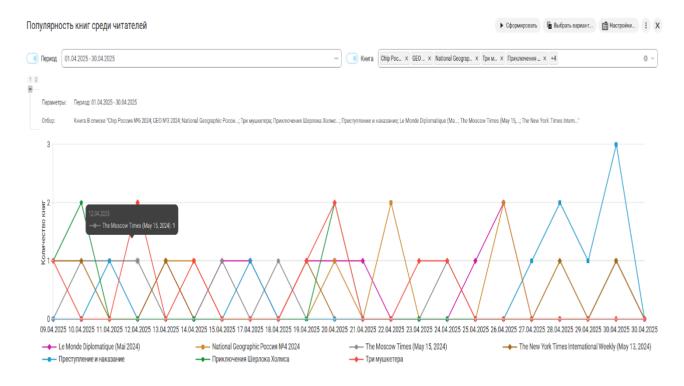


Рисунок 3.11 – Отчет по популярности книг

- о **Ожидаемый результат:** Отчет сформирован, отображает список книг, отсортированный по количеству выдач за указанный период. Данные соответствуют движениям в регистре "ПопулярностьКниги".
- о **Результат тестирования:** Отчет формируется корректно, данные соответствуют учетным данным.

#### Выводы по тестированию:

Проведенное функциональное тестирование показало, что основные модули и сценарии работы АБИС функционируют корректно в соответствии с требованиями технического задания. Выявленные в процессе тестирования незначительные ошибки были оперативно устранены. Система готова к опытной эксплуатации и дальнейшему развитию.

## 3.2. Руководство пользователя

Данное руководство предназначено для пользователей автоматизированной библиотечной информационной системы (АБИС) и описывает порядок выполнения основных операций.

## 1. Начало работы

- Запустите ярлык "1С:Предприятие" на рабочем столе или в меню "Пуск".
  - В окне запуска выберите информационную базу вашей библиотеки.
- Введите ваш Логин и Пароль, предоставленные администратором системы.
  - Нажмите кнопку "ОК".

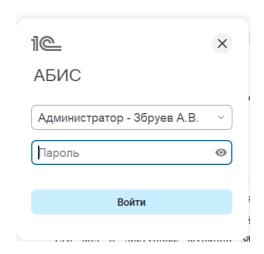


Рисунок 3.12 – Окно входа в систему

## 2. Основной интерфейс (Рабочий стол)

После успешного входа в систему откроется основной рабочий стол АБИС. Его вид и доступные функции зависят от вашей роли (Библиотекарь, Комплектатор фондов, Администратор).

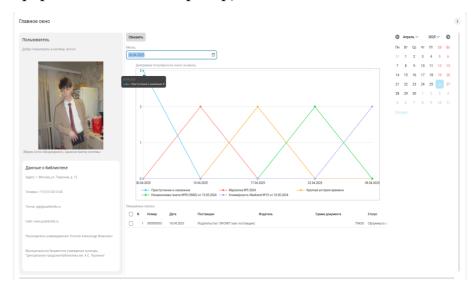


Рисунок 3.13 – Основной рабочий стол АБИС

Основные элементы интерфейса:

- Панель разделов: Обычно расположена слева или сверху и содержит основные функциональные блоки системы (например, "Главное", "Книжный фонд", "Читатели", "Обслуживание", "Закупки", "Отчеты", "Администрирование").
- Панель навигации: При выборе раздела в этой панели отображаются команды для перехода к справочникам, документам и отчетам, относящимся к данному разделу.
- Рабочая область: Основное пространство окна, где открываются списки, формы объектов (справочников, документов), отчеты и обработки.
- **Начальная страница (виджеты):** На начальной странице могут быть размещены информационные блоки (виджеты) для быстрого доступа к важной информации (например, данные о пользователе, ожидаемые заказы, диаграмма популярности книг).

## 3. Общие принципы работы

- Списки: Для просмотра и выбора данных используются формы списков (справочников, документов). В списках доступны функции сортировки (щелчком по заголовку колонки), поиска (в строке поиска над списком), отбора (через кнопку "Еще" -> "Настроить список").
- Создание объектов: Для создания нового элемента справочника или документа используется кнопка "Создать" в форме списка или соответствующая команда в панели навигации.
- **Редактирование объектов:** Для изменения существующего объекта выберите его в списке и нажмите кнопку "Изменить" (или иконку карандаша), либо дважды щелкните по строке.
- Запись и проведение: После внесения данных в форму документа необходимо его сохранить. Используйте кнопки:
- о "Записать": Сохраняет документ без формирования движений в регистрах.

- о "Провести": Сохраняет документ и формирует движения в учетных регистрах.
  - о "Провести и закрыть": Проводит документ и закрывает его форму.
- Печать: Большинство документов и отчетов имеют кнопку "Печать" для вывода соответствующих печатных форм.

## 4. Основные операции (для роли Библиотекарь)

- Регистрация нового читателя:
- 1. Перейдите в раздел "Читатели".
- 2. В панели навигации выберите "Читатели".
- 3. Нажмите "Создать".
- 4. Заполните поля: ФИО, Дата рождения, Контакты, Адрес. (Рисунок 3.2)
  - 5. Нажмите "Записать и закрыть".
  - Оформление читательского билета:
  - 1. Перейдите в раздел "Обслуживание" (или "Читатели").
  - 2. Выберите "Читательские билеты".
  - 3. Нажмите "Создать".
  - 4. Выберите читателя.
- 5. Выберите тип "Абонемента" и "Возрастной рейтинг". Цена и дата окончания рассчитаются автоматически. (Рисунок 3.3)
  - 6. Нажмите "Провести и закрыть".
  - Выдача книги:
  - 1. Перейдите в раздел "Обслуживание".
  - 2. Выберите "Выдача книг".
  - 3. Нажмите "Создать".
  - 4. Укажите "Дату" выдачи.
- 5. Выберите "Читателя". Система проверит наличие действующего билета.
  - 6. В табличной части "Книги" нажмите "Добавить".

- 7. Выберите "Книгу" из справочника (можно использовать поиск). Система проверит доступность книги.
  - 8. Укажите "Качество книги при выдаче".
- 9. Установите планируемую "Дату возврата книги" (обычно не более месяца).
  - 10. Повторите шаги 6-9 для других книг, если необходимо. (Рисунок 3.4)
  - 11. Нажмите "Провести и закрыть".

## • Возврат книги:

- 1. Перейдите в раздел "Обслуживание".
- 2. Выберите "Возврат книг".
- 3. Нажмите "Создать".
- 4. *Рекомендуемый способ:* Нажмите "Создать на основании" и выберите документ "Выдача книг", по которому возвращаются книги. Форма заполнится автоматически.
- 5. *Альтернативный способ:* Заполните "Дату", выберите "Читателя" и "Документ основания" (выдачи) вручную. Нажмите "Заполнить" -> "Заполнить по основанию".
- 6. В табличной части "Книги" для каждой возвращаемой книги укажите "Качество книги при возврате". Если оно отличается от качества при выдаче, система выдаст сообщение. (Рисунок 3.5)
- 7. Нажмите "Провести и закрыть". Система автоматически рассчитает и зарегистрирует штраф, если возврат просрочен.

## • Оплата штрафа:

- 1. Перейдите в раздел "Обслуживание".
- 2. Выберите "Оплата штрафов".
- 3. Нажмите "Создать".
- 4. Выберите "Читателя". Табличная часть "Штрафы" заполнится его неоплаченными штрафами.
- 5. Установите флажок "Оплачено" для тех штрафов, которые оплачиваются. Сумма к оплате рассчитается автоматически. (Рисунок 3.6)

6. Нажмите "Провести и закрыть".

## 5. Основные операции (для роли Комплектатор фондов)

- Создание заказа на закупку:
- 1. Перейдите в раздел "Закупки".
- 2. Выберите "Заказы поставщикам" (или "Закупки книг").
- 3. Нажмите "Создать".
- 4. Выберите "Поставщика" или "Издательство".
- 5. Заполните табличную часть "Книги", указав книги, количество и цену.
  - 6. Установите статус заказа.
  - 7. Нажмите "Записать" или "Провести". (Рисунок 3.7)
  - Оформление прихода книг:
  - 1. Перейдите в раздел "Закупки" или "Книжный фонд".
  - 2. Выберите "Приход книг".
  - 3. Нажмите "Создать".
- 4. *Рекомендуемый способ:* Нажмите "Создать на основании" и выберите документ "Закупка книг".
- 5. *Альтернативный способ:* Заполните шапку вручную (источник, поставщик/издатель), затем табличную часть.
  - 6. Проверьте/скорректируйте количество и цены. (Рисунок 3.8)
  - 7. Нажмите "Провести и закрыть".
  - Списание книг:
  - 1. Перейдите в раздел "Книжный фонд".
  - 2. Выберите "Списание книг".
  - 3. Нажмите "Создать".
  - 4. В табличной части "Книги" нажмите "Добавить".
- 5. Выберите "Книгу" для списания, укажите "Количество" и "Причину списания". (Рисунок 3.9)
  - 6. Нажмите "Провести и закрыть".

## 6. Формирование отчетов

- Перейдите в раздел "Отчеты".
- Выберите необходимый отчет из списка (например, "Остатки книг", "Задолженности читателей").
- В форме отчета установите параметры (период, отборы по читателю, книге и т.п.).
  - Нажмите кнопку "Сформировать". (Рисунок 3.11)

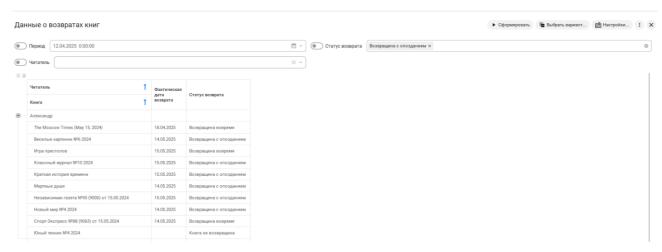


Рисунок 3.14 – Пример отчета "Задолженности читателей"

• Сформированный отчет можно распечатать или сохранить в файл (через меню "Файл" или кнопку "Еще").

## 7. Завершение работы

• Для выхода из программы закройте главное окно приложения или выберите пункт меню "Файл" -> "Выход".

#### ВЫВОДЫ ПО ГЛАВЕ 3

В третьей главе дипломного проекта была представлена экспериментально-прикладная работы, проверке посвященная часть работоспособности разработанной описанию использования автоматизированной библиотечной информационной системы (АБИС).

В рамках функционального тестирования были проверены ключевые сценарии использования системы, охватывающие основные бизнес-процессы библиотеки. Тестирование включало проверку регистрации читателей и выдачи читательских билетов, операций выдачи и возврата книг (в том числе с просрочкой и начислением штрафов), оплаты штрафов, процессов закупки и прихода книг, списания литературы, продления книг и формирования основных отчетов. По результатам тестирования можно сделать вывод о корректной работе реализованного функционала и соответствии системы требованиям, изложенным в техническом задании. Обнаруженные незначительные недочеты были устранены.

Для пользователей системы было разработано подробное руководство, описывающее порядок выполнения основных операций: от входа в систему и навигации по интерфейсу до работы со справочниками, документами и отчетами. Руководство пользователя содержит пошаговые инструкции и иллюстрировано скриншотами интерфейса программы, что облегчает освоение системы сотрудниками библиотеки с разным уровнем компьютерной грамотности.

Таким образом, экспериментально-прикладная часть работы подтвердила работоспособность разработанной АБИС и ее готовность к практическому применению. Тестирование показало стабильность системы и корректность выполнения основных функций, а подготовленное руководство пользователя обеспечивает необходимую поддержку для эффективного использования системы сотрудниками библиотеки.

#### ЗАКЛЮЧЕНИЕ

В рамках данного дипломного проекта была успешно решена задача разработки автоматизированной библиотечной информационной системы (АБИС) на платформе "1С:Предприятие 8.5. Актуальность данной разработки обусловлена необходимостью модернизации библиотечных процессов, повышения эффективности управления фондами и улучшения качества обслуживания читателей в современных условиях.

В ходе работы над проектом были выполнены все поставленные задачи. Проведен детальный анализ предметной области, позволивший выявить ключевые бизнес-процессы библиотеки и сформулировать четкие требования к функциональности информационной системы. На основе этого анализа было разработано техническое задание, которое легло в основу проектирования и разработки АБИС.

Была спроектирована архитектура системы, основанная на клиентсерверной технологии платформы "1С:Предприятие", обеспечивающая надежность, производительность и масштабируемость решения. Разработана структура базы данных, включающая необходимые справочники, документы, регистры сведений и накопления для хранения и обработки всей информации, связанной с деятельностью библиотеки.

Реализованы основные функциональные модули системы, автоматизирующие процессы учета книжного фонда (поступление, списание, инвентаризация), обслуживания читателей (регистрация, выдача и возврат книг, продление, учет билетов и штрафов), закупки литературы и учета пожертвований. Разработан интуитивно понятный пользовательский интерфейс и настроена ролевая модель доступа, соответствующая структуре персонала библиотеки. Реализован набор отчетов, предоставляющих необходимую аналитическую информацию для принятия управленческих решений.

Проведено функциональное тестирование разработанной системы, подтвердившее ее работоспособность и соответствие заявленным требованиям.

Выявленные в ходе тестирования недочеты были устранены. Подготовлено руководство пользователя, облегчающее освоение системы сотрудниками библиотеки.

Таким образом, цель дипломного проекта — создание программного обеспечения для комплексной автоматизации библиотеки — была достигнута. Разработанная АБИС представляет собой готовое к внедрению решение, способное значительно повысить эффективность работы библиотеки, оптимизировать использование ресурсов и улучшить качество предоставляемых услуг. Система обладает потенциалом для дальнейшего развития и модификации, например, путем интеграции с веб-сайтом библиотеки или добавления новых функциональных модулей.

#### СПИСОК ЛИТЕРАТУРЫ

- ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
   Введ. 1980-01-01. М.: Издательство стандартов, 1978. 9 с.
- 2. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. Введ. 1992-01-01. М.: Издательство стандартов, 1991. 27 с.
- 3. ГОСТ Р 7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления. Введ. 2009-01-01. М.: Стандартинформ, 2008. 23 с.
- 4. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. Введ. 2012-01-01. М.: Стандартинформ, 2011. 147 с.
- 5. Гартсуева, Е. А. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы / Е.А. Гартсуева. М.: ООО "1С-Паблишинг", 2020. 977 с.
- 6. Радченко, М. Г. 1С:Программирование для начинающих. Разработка в системе 1С:Предприятие 8.3 / М. Г. Радченко. М.: ООО "1С-Паблишинг", 2022.-494 с.
- 7. Хрусталева, Е. Ю. Разработка сложных отчетов в "1С:Предприятии 8". Система компоновки данных / Е.Ю. Хрусталева. М.: ООО "1С-Паблишинг", 2019. 458 с.
- 8. Ажеронок, В.А. Разработка управляемого интерфейса / В.А. Ажеронок, А.В. Островерх. М.: ООО «1С-Паблишинг», 2018. 374 с.
- 9. Насибов, И. И. Секреты профессиональной работы с «1С:Предприятие 8.3». Анализ и разработка конфигураций / И.И. Насибов. СПб.: БХВ-Петербург, 2021. 688 с.
- 10. Чистов, Д.В. Хозяйственные операции в "1С:Бухгалтерии 8". Задачи, решения, результаты / Д.В. Чистов, С.А. Харитонов. М.: ООО "1С-Паблишинг", 2021. 477 с.

- 11. Островерх А.В. Практика применения 1С:Библиотеки стандартных подсистем / А.В. Островерх. М.: ООО «1С-Паблишинг», 2020. 654 с.
- 12. Бойко, В.В. Проектирование баз данных информационных систем / В.В. Бойко, В.М. Савинков. М.: Финансы и статистика, 2002. 352 с.
- 13. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. 3-е изд. М.: Вильямс, 2016. 1436 с.
- 14. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс, Дж. Битти. 3-е изд. М.: Русская Редакция, 2014. 736 с.
- 15. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. М.: Русская Редакция, 2017. 896 с.
- 16. 1С:Предприятие 8. Руководство разработчика [Электронный ресурс]. Режим доступа: https://its.1c.ru/db/v8devdoc
- 17. Встроенный язык. Синтакс-помощник [Электронный ресурс] // 1С:Предприятие 8. Информационно-технологическое сопровождение. Режим доступа: https://its.1c.ru/db/v83doc/bookmark/dev/SyntaxHelper.
- 18. Система стандартов и методик разработки конфигураций для платформы 1С:Предприятие 8 [Электронный ресурс] // Фирма "1С". Режим доступа: https://its.1c.ru/db/v8std.
- 19. Основы работы с запросами в 1С [Электронный ресурс] // Инфостарт. Режим доступа: https://infostart.ru/public/15708/
- 20. Работа с формами в 1С: Управляемое приложение [Электронный ресурс] // Хабр. Режим доступа: https://habr.com/ru/articles/186150/.
- 21. Проведение документов в 1С [Электронный ресурс] // Программист1С.РФ. Режим доступа: https://программист1с.рф/stati/provedenie-dokumentov-1s.html
- 22. Использование регистров накопления в 1С [Электронный ресурс] // 1С: Лекторий. Режим доступа: https://its.1c.ru/lector/.
- 23. Работа с хранилищем значения в 1С [Электронный ресурс] // Клерк.ру. Режим доступа: https://www.klerk.ru/1c/articles/512367/.

- 24. Разработка печатных форм в 1С: Управляемое приложение [Электронный ресурс] // Инфостарт. Режим доступа: https://infostart.ru/public/115953/.
- 25. Моделирование бизнес-процессов. Нотация UML. Use Case Diagram [Электронный ресурс] // intuit.ru. Режим доступа: https://intuit.ru/studies/courses/6/6/lecture/174.

## приложения

# Приложение 1

# Компакт-диск с материалами проекта

На диске располагается:

- Проект программы.
- Файл дипломного проекта в формате MS Word

### Приложение 2

#### КОД ПРОГРАММЫ

Код общего модуля «ОбщийМодульФункции» РезультатЗапроса = Запрос.Выполнить(); Функция ЦеныИзРегистраСведений Выборка = РезультатЗапроса.Выбрать(); (Дата, Книги ИзРегистра) Экспорт Отбор = Новый Если Выборка. Следующий() Тогда Структура("Книги",КнигиИзРегистра); Возврат Выборка. Цена; ЗначениеЦены = Иначе РегистрыСведений.Цены.ПолучитьПоследнее(Дата, Возврат 0; // Или другое значение по умолчанию, или выбросить исключение Отбор); Возврат ЗначениеЦены.Цена; КонецЕсли; КонецФункции КонецФункции Функция ПолучитьЦенуБилета(Абонемент, Функция Получить ЦенуШтрафа (Штраф) Экспорт Возрастной Рейтинг) Экспорт Запрос = Новый Запрос; Запрос = Новый Запрос; Запрос. Текст = "ВЫБРАТЬ ПЕРВЫЕ 1 Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 1 ЦеныШтрафов.Цена |ИЗ Цены Читательских Билетов. Цена |ИЗ РегистрСведений. ЦеныШтрафов КАК ЦеныШтрафов РегистрСведений.ЦеныЧитательскихБилетов КАК ГДЕ ЦеныЧитательскихБилетов ЦеныШтрафов.Штраф = &Штраф ГДЕ |УПОРЯДОЧИТЬ ПО Цены Читательских Билетов. Абонемент = ЦеныШтрафов.Период УБЫВ"; &Абонемент ΙИ Запрос. Установить Параметр ("Штраф", Штраф); Цены Читательских Билетов. Возрастной Рейтинг = // Передаем параметр в запрос &ВозрастнойРейтинг |УПОРЯДОЧИТЬ ПО Результат = Запрос.Выполнить(); ЦеныЧитательскихБилетов.Период Выборка = Результат.Выбрать(); УБЫВ"; Если Выборка. Следующий () Тогда Запрос. Установить Параметр ("Абонемент", Возврат Выборка. Цена; Абонемент); Иначе Запрос. Установить Параметр ("Возрастной Р Возврат 0; // Или другое значение по ейтинг", ВозрастнойРейтинг); умолчанию, или выбросить исключение КонецЕсли;

Возврат Ложь; КонецФункции КонецЕсли; Если СтрНайти(СимволыНомера, Код общего модуля «ВалидацияПолей» Символ) > 0 Тогда &НаСервере Процедура ПроверитьЕтаіl(Почта, ЭлементПочты Форматированный Номер = = Неопределено) Экспорт Форматированный Номер + Символ; ИначеЕсли Шаблон = ".+@.+\..+"; // Регулярное СтрНайти(СимволыРазделители, Символ) = 0 Тогда выражение для проверки email Возврат Ложь; КонецЕсли; СтрокаАдреса = Почта; КонецЦикла; RegExp = Новый COMOбъект("VBScript.RegExp"); Если RegExp.MultiLine = Ложь; ПустаяСтрока(ФорматированныйНомер) Тогда RegExp.Global = Истина; Возврат Ложь; КонецЕсли; RegExp.IgnoreCase = Истина; RegExp.Pattern = Шаблон; Если СтрДлина(Форматированный Номер) Если НЕ RegExp.Test(Почта) Тогда < 3 Тогда ВызватьИсключение Возврат Ложь; "Некорректный email адрес. Пожалуйста, введите КонецЕсли: адрес в формате ""name@domain.com""."; КонецЕсли; Возврат Истина; КонецФункции КонецПроцедуры Код модуля объекта справочника «Книги» Функция &НаСервере КорректноВведенНомерТелефона(НомерТелефона) Процедура ПередУдалением(Отказ) Экспорт // Получаем остаток книги в регистре "ОстаткиКниг" СимволыНомера = "+1234567890"; Остаток = СимволыPазделители = "()- "; ПолучитьОстатокКниги(ЭтотОбъект.Ссылка); Форматированный Номер = ""; // Если остаток больше 0, запрещаем удаление Для Позиция = 1 По Если Остаток < 0 Тогда СтрДлина(НомерТелефона) Цикл Сообщить("Нельзя удалить книгу, так как она Символ = Сред(НомерТелефона, есть в наличии в библиотеке. Остаток: " + Остаток); Отказ = Истина: Позиция, 1);

КонецЕсли:

КонецПроцедуры

Если Символ = "+" И Не

ПустаяСтрока(ФорматированныйНомер) Тогда

Если НЕ Результат Тогда &НаСервере Возврат; Функция ПолучитьОстатокКниги(Книга) КонецЕсли; Запрос = Новый Запрос; СсылкаНаКартинку = Адрес; Запрос. Текст = "ВЫБРАТЬ КонецПроцедуры ОстаткиКниг.Количество |ИЗ &НаСервере РегистрНакопления.ОстаткиКниг КАК Процедура ПередЗаписьюНаСервере(Отказ, ОстаткиКниг ТекущийОбъект, ПараметрыЗаписи) ГДЕ ОстаткиКниг.Книги = &Книга"; Если ЭтоАдресВременногоХранилища(СсылкаНаКартин Запрос. Установить Параметр ("Книга", Книга); ку) Тогда ТекущийОбъект.ДанныеКартинки Результат = Запрос.Выполнить().Выбрать(); = Новый Если Результат. Следующий() Тогда ХранилищеЗначения(ПолучитьИзВременн Возврат Результат. Количество; огоХранилища(СсылкаНаКартинку)); Иначе КонецЕсли: Возврат 0; КонецЕсли: КонецПроцедуры КонецФункции &НаСервере Код формы элемента справочника «Книги» Процедура ПриСозданииНаСервере(Отказ, &НаКлиенте Стандартная Обработка) Процедура Вставить Картинку (Команда) СсылкаНаКартинку = Оповещение = Новый Получить Навигационную Ссылку (Объект. Ссылка, ОписаниеОповещения ("Обработать Выбор Файла", Э "ДанныеКартинки"); тотОбъект); НачатьПомещениеФайла(Оповещение,,,Ис КонецПроцедуры тина, Уникальный Идентификатор); &НаСервере КонецПроцедуры Функция ПолучитьОстаток(Книга) &НаКлиенте Результат = 0;Процедура ОбработатьВыборФайла(Результат, Отбор = Новый Структура; Отбор.Вставить("Книги", Книга); ВыбранноеИмяФайла,ДополнительныйПараметр) НайденныеЗначения = Экспорт РегистрыНакопления.ОстаткиКниг.Остатки(Текущ аяДата(), Отбор);

Если НайденныеЗначения.Количество() > 0 ЭлементОформления = Тогда УсловноеОформлениеСписка.Элементы.Добавить() Результат = НайденныеЗначения[0].Количество; КонецЕсли; Возврат Результат; ЭлементОформления.Оформление. Установ итьЗначениеПараметра("ЦветФона", КонецФункции ВыборкаДетальныеЗаписи.Цвет.Получить()); ЭлементОтбора = Код формы списка справочника «Книги» Процедура ПриСозданииНаСервере(Отказ, ЭлементОформления.Отбор.Элементы.Добавить(Т Стандартная Обработка) ип("ЭлементОтбораКомпоновкиДанных")); ОбновитьОстаткиКниг(); ЭлементОтбора.ЛевоеЗначение=Новый Установить Условное Оформление(); ПолеКомпоновкиДанных("Статус"); ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Равно; ЭлементОтбора.ПравоеЗначение = КонецПроцедуры ВыборкаДетальныеЗаписи.Ссылка; Процедура Установить Условное Оформление() КонепПикла: УсловноеОформлениеСписка=Список.Ком поновщик Настроек. Настройки. Условное Оформлен КонецПроцедуры ие; &НаСервере Запрос = Новый Запрос; Процедура ОбновитьОстаткиКниг() Запрос.Текст = "ВЫБРАТЬ Запрос = Новый Запрос; Статус.Ссылка КАК Ссылка, Запрос.Текст = Статус.Цвет КАК Цвет "ВЫБРАТЬ |ИЗ Книги.Ссылка Справочник. Статус КАК Статус"; |ИЗ Справочник.Книги КАК Книги"; // РезультатЗапроса = Запрос.Выполнить(); Выбираем все книги ВыборкаДетальныеЗаписи = Результат = Запрос.Выполнить(); РезультатЗапроса.Выбрать(); Выборка = Результат.Выбрать(); Пока Выборка.Следующий() Цикл Пока ВыборкаДетальныеЗаписи.Следующий() Цикл Книга =

Выборка. Ссылка. Получить Объект();

Книга.КоличествоНаСкладе = Возврат Результат; ПолучитьОстаток(Выборка.Ссылка); КонецФункции // Проверка и установка статуса Если Книга. Количество На Складе > Код формы элемента справочника «Статус» 0 Тогда Процедура ПередЗаписьюНаСервере(Отказ, Книга.Статус = Текущий Объект, Параметры Записи) Справочники. Статус. Найти По Наименованию ("Есть на складе"); ЗначениеЦвета = Цвет; Иначе Если ЗначениеЦвета = Новый Цвет(0,0,0) Тогда Книга.Статус = Справочники. Статус. Найти ПоНаименованию ("Отс ЗначениеЦвета = Неопределено; утствует на складе"); КонецЕсли; КонецЕсли; ТекущийОбъект.Цвет=Новый ХранилищеЗначения(ЗначениеЦвета); Попытка Книга.Записать(); КонецПроцедуры Исключение Сообщить("Ошибка при Процедура ПриЧтенииНаСервере(ТекущийОбъект) записи книги " + Книга + ": " + ОписаниеОшибки()); Цвет = ТекущийОбъект.Цвет.Получить(); КонепПопытки: КонецЦикла; КонецПроцедуры Элементы.Список.Обновить(); Код формы элемента справочника «Поставщики» КонецПроцедуры &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, &НаСервере ТекущийОбъект, ПараметрыЗаписи) Функция ПолучитьОстаток(Книга) Попытка Pезультат = 0;Отбор = Новый Структура; ВалидацияПолей.ПроверитьEmail(Текущий Отбор.Вставить("Книги", Книга); Объект.Почта); НайденныеЗначения = Исключение РегистрыНакопления.ОстаткиКниг.Остатки(Текущ Сообщить(ОписаниеОшибки()); аяДата(), Отбор); Отказ = Истина; Если НайденныеЗначения.Количество() > 0 Возврат; КонецПопытки; Тогда Результат =

НайденныеЗначения[0].Количество;

КонецЕсли;

Если НЕ	Оповещение = Новый
ВалидацияПолей.КорректноВведенНомерТелефона	ОписаниеОповещения("ОбработатьВыборФайла",Э
(ТекущийОбъект.Телефон) Тогда	тотОбъект);
ВызватьИсключение	НачатьПомещениеФайла(Оповещение,,,,Ис
"Некорректный номер телефона. Пожалуйста,	тина, УникальныйИдентификатор);
введите номер в международном формате.";	• • •
КонецЕсли;	КонецПроцедуры
КонецПроцедуры	&НаКлиенте
	Процедура ОбработатьВыборФайла(Результат,
Код формы элемента справочника «Читатели»	Адрес,
&НаСервере	ВыбранноеИмяФайла,ДополнительныйПар
Процедура ПередЗаписьюНаСервере(Отказ,	аметр) Экспорт
ТекущийОбъект, ПараметрыЗаписи)	
	Если НЕ Результат Тогда
Попытка	Возврат;
	КонецЕсли;
ВалидацияПолей.ПроверитьEmail(Текущий	СсылкаНаКартинку = Адрес;
Объект.Почта);	
Исключение	КонецПроцедуры
Сообщить(ОписаниеОшибки());	
Отказ = Истина;	&НаСервере
Возврат;	Процедура ПередЗаписьюНаСервере(Отказ,
КонецПопытки;	ТекущийОбъект, ПараметрыЗаписи)
	Если
Если НЕ	ЭтоАдресВременногоХранилища(СсылкаНаКартин
ВалидацияПолей.КорректноВведенНомерТелефона	ку) Тогда
(ТекущийОбъект.Телефон) Тогда	ТекущийОбъект.Фотография =
ВызватьИсключение	Новый
"Некорректный номер телефона. Пожалуйста,	
введите номер в международном формате.";	ХранилищеЗначения(ПолучитьИзВременн
КонецЕсли;	огоХранилища(СсылкаНаКартинку));
<i>T</i>	КонецЕсли;
КонецПроцедуры	П
TC - 1	Попытка
Код формы элемента справочника	D
«Сотрудники»	ВалидацияПолей.ПроверитьЕmail(Текущий
&НаКлиенте  Произвите Ретерит Фете (V смение)	Объект.Почта);
Процедура ВставитьФото(Команда)	Исключение
	Сообщить(ОписаниеОшибки());

Отказ = Истина;

Если НЕ Возврат; КонецПопытки; ВалидацияПолей.КорректноВведенНомерТелефона (ТекущийОбъект.Телефон) Тогда Если НЕ ВызватьИсключение Валидация Полей. Корректно Введен Номер Телефона "Некорректный номер телефона. Пожалуйста, (ТекущийОбъект.Телефон) Тогда введите номер в международном формате."; ВызватьИсключение КонецЕсли; "Некорректный номер телефона. Пожалуйста, введите номер в международном формате."; КонецПроцедуры КонецЕсли; Код формы элемента справочника «Библиотека» КонецПроцедуры &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, &НаСервере Текущий Объект, Параметры Записи) Процедура ПриСозданииНаСервере(Отказ, Стандартная Обработка) Попытка СсылкаНаКартинку = ВалидацияПолей.ПроверитьЕmail(Текущий Получить Навигационную Ссылку (Объект. Ссылка, Объект.Почта): "Фотография"); Исключение Сообщить(ОписаниеОшибки()); КонецПроцедуры Отказ = Истина; Возврат; Код формы элемента справочника КонецПопытки; «Руководители» &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, Если НЕ ТекущийОбъект, ПараметрыЗаписи) ВалидацияПолей.КорректноВведенНомерТелефона (ТекущийОбъект.Телефон) Тогда Попытка ВызватьИсключение "Некорректный номер телефона. Пожалуйста, ВалидацияПолей.ПроверитьЕmail(Текущий введите номер в международном формате."; Объект.Почта); КонецЕсли: Исключение Сообщить(ОписаниеОшибки()); Еспи Отказ = Истина; ЭтоАдресВременногоХранилища(СсылкаНаКартин Возврат; ку) Тогда КонецПопытки; ТекущийОбъект.ДанныеКартинки = Новый

	56
	КонецПроцедуры
ХранилищеЗначения(ПолучитьИзВременн	
огоХранилища(СсылкаНаКартинку));	Код модуля объекта документа «ПриходКниг»
КонецЕсли;	Процедура ОбработкаПроведения(Отказ, Режим)
	// регистр ОстаткиКниг Приход
	Движения.ОстаткиКниг.Записывать =
КонецПроцедуры	Истина;
	Для Каждого ТекСтрокаКниги Из Книги
&НаКлиенте	Цикл
Процедура Вставить Картинку (Команда)	Движение =
	Движения.ОстаткиКниг.Добавить();
Оповещение = Новый	Движение.ВидДвижения =
ОписаниеОповещения("ОбработатьВыборФайла",Э	ВидДвиженияНакопления.Приход;
тотОбъект);	Движение.Период = Дата;
НачатьПомещениеФайла(Оповещение,,,,Ис	Движение.Книги =
тина, УникальныйИдентификатор);	ТекСтрокаКниги.Книга;
	Движение.Количество =
КонецПроцедуры	ТекСтрокаКниги.Количество;
	КонецЦикла;
&НаКлиенте	
Процедура ОбработатьВыборФайла(Результат,	
Адрес,	// регистр ОборотЗаказовКниг Расход
ВыбранноеИмяФайла,ДополнительныйПараметр)	Движения.ОборотЗаказовКниг.Записывать
Экспорт	= Истина;
E HE D T	// П
Если НЕ Результат Тогда	// Проверяем источник получения
Возврат;	Если ИсточникПолучения <>
КонецЕсли;	Перечисления.ИсточникПолучения.Пожертвование
СсылкаНаКартинку = Адрес;	Тогда
КонецПроцедуры	Для Каждого ТекСтрокаКниги Из
	Книги Цикл
	// Получаем остаток по
&НаСервере	книге, поставщику и издателю

Процедура ПриСозданииНаСервере(Отказ, ОстатокКниг = Стандартная Обработка) ПолучитьОстаток(ТекСтрокаКниги.Книга, Поставщик, Издатель); СсылкаНаКартинку =

ПолучитьНавигационнуюСсылку(Объект.Ссылка,

"ДанныеКартинки");

Если ОстатокКниг < ТекСтрокаКниги.Количество Тогда

Сообщить("Не достаточно книг " + ТекСтрокаКниги.Книга + " у Движения. Обороты Пожертвований. Записы поставщика " + Поставщик + " и издателя " + вать = Истина; Издатель + ". Доступный остаток: " + Движение = ОстатокКниг); Движения. Обороты Пожертвований. Добавить(); Движение.ВидДвижения = Отказ = Истина; Возврат; // ВидДвиженияНакопления.Расход; Прерываем проведение, если не хватает книг Движение.Период = Дата; КонецЕсли; Движение.ДокументПожертвования = ДанныеОПожертвований; Движение = Движения. Оборот Заказов Книг. Добавить(); Движение.Количество = 1; Движение.ВидДвижения = КонецЕсли; ВидДвиженияНакопления.Расход; Движение.Период = Дата; // регистр ДоступныеКниги Приход Движение.Книга = Движения. Доступные Книги. Записывать = Истина; ТекСтрокаКниги.Книга; Движение.ВыборЗакупки Для Каждого ТекСтрокаКниги Из Книги = 3akynka;Цикл Движение.Поставщик = Движение = Поставщик; Движения. Доступные Книги. Добавить(); Движение.Издатель = Движение.ВидДвижения = Издатель; ВидДвиженияНакопления.Приход; Движение.Количество = Движение.Период = Дата; ТекСтрокаКниги.Количество; Движение.Книги = КонецЦикла; ТекСтрокаКниги.Книга; Движение.Количество = КонецЕсли; ТекСтрокаКниги.Количество; КонецЦикла; Если ЭтотОбъект.ИсточникПолучения = Перечисления. Источник Получения. Пожертвование Тогда Если Не Пожертвование Доступно (Этот Объект. Данные ОПо КонецПроцедуры жертвований) Тогда Сообщить("Документ Функция ПолучитьОстаток(Книга, Поставщик = пожертвования уже использован!"); Неопределено, Издатель = Неопределено) Отказ = Истина; Возврат; // Прерываем Результат = 0;проведение

// Создаем структуру отборов

КонецЕсли;

Отбор = Новый Структура("Книга", Если Не Книга); ЗначениеЗаполнено(ЭтотОбъект.Ответственный) Тогда // Добавляем отбор по поставщику, если он ЭтотОбъект.Ответственный = ПользователиИнформационнойБазы. ТекущийПоль передан Если ЗначениеЗаполнено(Поставщик) зователь(); Тогда КонецЕсли; Отбор.Вставить("Поставщик", Поставщик); Если ЭтотОбъект.ИсточникПолучения = КонецЕсли; Перечисления. Источник Получения. Пожертвование Тогда // Добавляем отбор по издателю, если он Если Не Пожертвование Доступно (Этот Объект. Данные ОПо передан Если ЗначениеЗаполнено(Издатель) Тогда жертвований) Тогда Отбор.Вставить("Издатель", Сообщить("Документ Издатель); пожертвования уже использован!"); КонецЕсли; Отказ = Истина; КонецЕсли; // Получаем остатки с учетом отбора КонецЕсли; НайденныеЗначения = РегистрыНакопления.ОборотЗаказовКниг.Остатки( КонецПроцедуры Дата, Отбор); Процедура ПриЗаписи(Отказ) Если НайденныеЗначения.Количество() > 0 Тогда Если Не Результат = ЗначениеЗаполнено(ЭтотОбъект.Ответственный) НайденныеЗначения[0].Количество; Тогда КонепЕсли: ЭтотОбъект.Ответственный = ПользователиИнформационнойБазы.ТекущийПоль Возврат Результат; зователь(); КонецЕсли; КонецФункции КонецПроцедуры Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения) Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СуммаДокумента =0; Стандартная Обработка) Для каждого Табличная Часть из Книги Если ТипЗнч(ДанныеЗаполнения) = Цикл Тип("ДокументСсылка.ЗакупкаКниг") Тогда СуммаДокумента = СуммаДокумента +ТабличнаяЧасть.Сумма; // Проверяем статус закупки

КонецЦикла;

Если ДанныеЗаполнения.Статус = // Прерываем заполнение Перечисления.СтатусЗакупки.Получен Тогда СтандартнаяОбработка = // Заполнение шапки Ложь; Закупка = Возврат; ДанныеЗаполнения.ВыборЗакупки; КонецЕсли; Издатель = ИначеЕсли ТипЗнч(ДанныеЗаполнения) = ДанныеЗаполнения. Издатель; Тип("ДокументСсылка.Пожертвование") Тогда Комментарий = // Заполнение шапки Комментарий = ДанныеЗаполнения.Комментарий; Ответственный = ДанныеЗаполнения.Комментарий; ДанныеЗаполнения.Ответственный; Ответственный = Поставщик = ДанныеЗаполнения.Ответственный; ДанныеЗаполнения.Поставщик; ДанныеОПожертвований = СуммаДокумента = ДанныеЗаполнения.Ссылка; ДанныеЗаполнения.СуммаДокумента; СуммаДокумента = ИсточникПолучения = ДанныеЗаполнения.СуммаДокумента; Перечисления. Источник Получения. Закупка; ИсточникПолучения = Основание = Перечисления. Источник Получения. Пожертвование ДанныеЗаполнения.Ссылка; Для Каждого Для Каждого ТекСтрокаКниги Из ТекСтрокаКниги Из ДанныеЗаполнения.Книги ДанныеЗаполнения.Книги Цикл Цикл НоваяСтрока = НоваяСтрока = Книги.Добавить(); Книги.Добавить(); НоваяСтрока.Книга = ТекСтрокаКниги.Книга; НоваяСтрока.Книга = НоваяСтрока.Количество ТекСтрокаКниги.Книга; = ТекСтрокаКниги.Количество; НоваяСтрока.Сумма = НоваяСтрока.Количество = ТекСтрокаКниги.Сумма; ТекСтрокаКниги.Количество; НоваяСтрока.Цена = ТекСтрокаКниги.Цена; КонецЦикла; НоваяСтрока.Сумма = ТекСтрокаКниги.Сумма; КонецЕсли; НоваяСтрока.Цена = ТекСтрокаКниги.Цена; КонецПроцедуры КонецЦикла; Иначе &НаСервере // Выводим сообщение об Функция ошибке Пожертвование Доступно (Документ Пожертвования Сообщить("Ввод на ) основании доступен только для закупок со

статусом 'Получен'!");

Отбор = Новый Сообщить("Нельзя Структура("ДокументПожертвования", удалить документ! Недостаточно книг на складе для книги """ + СтрокаТЧ.Книга + """. Остаток ДокументПожертвования); Остатки = после удаления: " + ОстатокПослеУдаления); РегистрыНакопления.ОборотыПожертвований.Ост Отказ = Истина; Возврат; // Прерываем атки(Дата, Отбор); цикл и отменяем удаление КонецЕсли; // Проверяем, что есть остатки и количество равно 1 КонецЦикла; Если Остатки. Количество() > 0 Тогда Возврат Остатки[0].Количество = КонецПроцедуры 1; КонецЕсли; Код формы документа «Приход книг» &НаКлиенте Возврат Ложь; Перем Пожертвование; Перем Закупка; КонецФункции &НаКлиенте &НаСервере Процедура Процедура ПередУдалением(Отказ) КнигиЦенаЗаШтПриИзменении(Элемент) Для Каждого СтрокаТЧ Из Книги Цикл СтрокаТабличнойЧасти = Отбор = Новый Элементы.Книги.ТекущиеДанные; Структура("Книги", СтрокаТЧ.Книга); СтрокаТабличнойЧасти.Сумма = Остатки = СтрокаТабличнойЧасти.Количество\*СтрокаТаблич РегистрыНакопления.ОстаткиКниг.Остатки(Дата, ной Части. Цена; Отбор); КонецПроцедуры Текущий Остаток = 0;Если Остатки. Количество() > 0 &НаКлиенте Тогда Процедура ТекущийОстаток = КнигиСуммаЦенаПриИзменении(Элемент) Остатки[0].Количество; КонецЕсли: СтрокаТабличнойЧасти = Элементы.Книги.ТекущиеДанные; // Вычисляем остаток после Если СтрокаТабличнойЧасти.Количество = удаления документа (вычитаем количество, так как 0 тогда удаление прихода уменьшает остаток) СтрокаТабличнойЧасти.ЦенаЗаШ ОстатокПослеУдаления = = 0: ТекущийОстаток; Иначе Если ОстатокПослеУдаления < 0

Тогда

СтрокаТабличнойЧасти.Цена = Объект. Издатель = 0; СтрокаТабличнойЧасти.Сумма / КонецЕсли; СтрокаТабличнойЧасти.Количество; ИначеЕсли Объект.ИсточникПолучения = КонецЕсли; Перечисления. Источник Получения. Закупка Тогда Элементы.Закупка.Видимость = КонецПроцедуры Истина; // Очищаем поле пожертвования, &НаКлиенте только если меняем источник получения Если Процедура КнигиКоличествоПриИзменении(Элемент) Объект. Данные ОПожертвований <>0 Тогда // Или Если ЗначениеЗаполнено(ДанныеОПожертвований) СтрокаТабличнойЧасти = Тогда Элементы.Книги.ТекущиеДанные; СтрокаТабличнойЧасти.Сумма = Объект. Данные OПожертвований = 0; СтрокаТабличнойЧасти.Количество\*СтрокаТаблич КонепЕсли: ной Части. Цена; // Устанавливаем видимость полей выбора в зависимости от значения "Закупка" КонецПроцедуры ЗакупкаПриИзмененииНаСервере(); // &НаСервере Вызываем обработчик изменения закупки КонецЕсли; Процедура ИсточникПолученияПриИзмененииНаСервере() // Управляем видимостью "Основание" // Скрываем все поля выбора сначала Элементы.Основание.Видимость = Элементы. Данные Опожертвований. Видим (Объект.ИсточникПолучения = octb = Ложь;Перечисления. Источник Получения. Закупка); Элементы. Закупка. Видимость = Ложь; Элементы. Поставщик. Видимость = Ложь; // Очищаем "Основание", если источник Элементы. Издатель. Видимость = Ложь; получения не "Закупка" Если Объект. Источник Получения <> Если Объект. Источник Получения = Перечисления. Источник Получения. Закупка И Перечисления. Источник Получения. Пожертвование ЗначениеЗаполнено(Объект.Основание) Тогда Тогда Объект.Основание = Неопределено; КонецЕсли: Элементы. Данные Опожертвований. Видим ость = Истина; // Очищаем поля, связанные с КонецПроцедуры закупкой, только если меняем источник получения Если Закупка <> 0 Тогда // Или &НаКлиенте Если ЗначениеЗаполнено(Закупка) Тогда Процедура Объект. 3акупка = 0; ИсточникПолученияПриИзменении(Элемент)

Объект. Поставщик = 0;

ИсточникПолученияПриИзмененииНаСерв // Очищаем поле издателя, только epe(); если меняем выбор закупки Если Объект. Издатель <> 0 Тогда КонецПроцедуры // Или Если ЗначениеЗаполнено(Издатель) Тогда Объект.Издатель = 0; &НаСервере КонецЕсли; Процедура ПриОткрытииНаСервере() КонецЕсли; ИсточникПолученияПриИзмененииНаСерв КонецПроцедуры epe(); &НаКлиенте КонецПроцедуры Процедура ЗакупкаПриИзменении(Элемент) &НаКлиенте ЗакупкаПриИзмененииНаСервере(); Процедура ПриОткрытии(Отказ) КонецПроцедуры ПриОткрытииНаСервере(); &НаКлиенте КонецПроцедуры Процедура КнигиКнигаПриИзменении(Элемент) &НаСервере СтрокаТЧ = Процедура ЗакупкаПриИзмененииНаСервере() Элементы.Книги.ТекущиеДанные; СтрокаТЧ.Цена=ОбщийМодульФункции.Ц // Скрываем оба поля сначала еныИзРегистраСведений(Объект.Дата,СтрокаТЧ.К Элементы. Поставщик. Видимость = Ложь; нига); Элементы. Издатель. Видимость = Ложь; КонецПроцедуры Если Объект. Закупка = Перечисления. Выбор Закупки. Издатель Тогда &НаСервере Элементы. Издатель. Видимость = Процедура Истина; ДанныеОпожертвованийПриИзмененииНаСервере( // Очищаем поле поставщика, только если меняем выбор закупки Если Объект. Поставшик <> 0 // Проверяем, выбран ли документ Тогда // Или Если ЗначениеЗаполнено(Поставщик) пожертвования Тогла Если Объект. Поставщик = 0; ЗначениеЗаполнено(Объект.ДанныеОПожертвован КонецЕсли; ий) Тогда ИначеЕсли Объект. Закупка = // Получаем документ Перечисления. Выбор Закупки. Поставщик Тогда пожертвования Элементы. Поставшик. Видимость =

Истина:

КонецЕсли; Пожертвование = Объект. Данные ОПожертвований. Получить Объект ( ); ЗакупкаДокумент = Объект.Основание.ПолучитьОбъект(); // Очищаем табличную часть ВыборЗакупкиЗначение = "Книги" в документе "Приход книг" ЗакупкаДокумент.ВыборЗакупки; // Используйте отдельную переменную Объект.Книги.Очистить(); // Копируем данные из табличной // Заполнение шапки части "Книги" документа "Пожертвование" Объект.Закупка = ВыборЗакупкиЗначение; Для Каждого // Устанавливаем перечисление в документ СтрокаПожертвования Из Пожертвование. Книги "ПриходКниг" Цикл Объект.Издатель = ЗакупкаДокумент.Издатель; Объект.Комментарий = НоваяСтрока = Объект.Книги.Добавить(); ЗакупкаДокумент.Комментарий; НоваяСтрока.Книга = Объект.Ответственный = СтрокаПожертвования.Книга; ЗакупкаДокумент.Ответственный; НоваяСтрока.Количество Объект.Поставщик = = СтрокаПожертвования.Количество; ЗакупкаДокумент.Поставщик; НоваяСтрока.Цена = Объект.СуммаДокумента = СтрокаПожертвования.Цена; ЗакупкаДокумент.СуммаДокумента; Объект.ИсточникПолучения = НоваяСтрока.Сумма = СтрокаПожертвования. Сумма; Перечисления. Источник Получения. Закупка; КонецЦикла; Объект.Основание = ЗакупкаДокумент.Ссылка; КонецЕсли; Объект.Книги.Очистить(); // Очищаем табличную часть перед заполнением КонецПроцедуры Для Каждого ТекСтрокаКниги Из &НаКлиенте ЗакупкаДокумент.Книги Цикл НоваяСтрока = Объект.Книги.Добавить(); Процедура ДанныеОпожертвованийПриИзменении(Элемент) НоваяСтрока.Книга = ТекСтрокаКниги.Книга; НоваяСтрока.Количество = ДанныеОпожертвованийПриИзмененииНа ТекСтрокаКниги.Количество; НоваяСтрока.Сумма = Сервере(); ТекСтрокаКниги.Сумма; КонецПроцедуры НоваяСтрока.Цена = ТекСтрокаКниги.Цена; КонецЦикла; &НаСервере Процедура ОснованиеПриИзмененииНаСервере() // Проверка соответствия данных \*после\* Если Объект. Основание = Неопределено Тогда заполнения Возврат; // Если основание не выбрано, ничего

не проверяем

Если Не КоличествоB3акупке = 0; Соответствуют Данные (Закупка Документ, Объект) Для Каждого СтрокаЗакупки Из Тогда // Передаем документ Закупка.Книги Цикл Сообщить("Данные прихода не соответствуют Если СтрокаПрихода.Книга = СтрокаЗакупки.Книга данным закупки!"); Объект.Основание = Неопределено; Тогда КонецЕсли; КоличествоВЗакупке = КонецПроцедуры СтрокаЗакупки.Количество; Прервать; // Книга &НаКлиенте найдена, выходим из внутреннего цикла Процедура ОснованиеПриИзменении(Элемент) КонецЕсли; КонецЦикла; ОснованиеПриИзмененииНаСервере(); Если КоличествоВЗакупке = 0 КонецПроцедуры Тогда Возврат Ложь; // Книга из прихода не найдена в закупке &НаСервере ИначеЕсли Функция Соответствуют Данные (Закупка, Приход) СтрокаПрихода. Количество > КоличествоВЗакупке Тогда // Проверка соответствия Возврат Ложь; // поставщика/издателя Количество в приходе превышает количество в Если Закупка. Выбор Закупки = закупке Перечисления. Выбор Закупки. Поставщик Тогда КонецЕсли; Если Закупка.Поставщик <> КонецЦикла; Приход.Поставщик Тогда Возврат Ложь; КонецЕсли; Возврат Истина; // Данные соответствуют ИначеЕсли Закупка.ВыборЗакупки = Перечисления. Выбор Закупки. Издатель Тогда КонецФункции Если Закупка. Издатель <> Приход. Издатель Тогда &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, Возврат Ложь; КонепЕсли: ТекущийОбъект, ПараметрыЗаписи) КонецЕсли; ТекущийПользователь = ПользователиИнформационнойБазы. ТекущийПоль // Проверяем, что все книги из прихода зователь(); есть в закупке с учетом количества Для Каждого СтрокаПрихода Из // Поиск сотрудника по логину только если

поле "Ответственный" не заполнено

Приход.Книги Цикл

Если Не ОбластьЗаголовок = ЗначениеЗаполнено(ТекущийОбъект.Ответственны Макет.ПолучитьОбласть("Заголовок"); ОбластьТекст = й) Тогда Макет.ПолучитьОбласть("Текст"); Найденный Сотрудник = Справочники.Сотрудники.НайтиПоРеквизиту("Лог ОбластьШапкаТаблици = ин", ТекущийПользователь.Имя); Макет.ПолучитьОбласть("ШапкаТаблици"); Область Таблица = Если Найденный Сотрудник <> Макет.ПолучитьОбласть("Таблица"); Неопределено Тогда ОбластьПодвалТаблици = Макет.ПолучитьОбласть("ПодвалТаблици"); ОбластьПодвал = ТекущийОбъект.Ответственный = Макет.ПолучитьОбласть("Подвал"); Найденный Сотрудник; Иначе Сообщить("Сотрудник не Область Заголовок. Параметры. Номер найден по логину текущего пользователя. Укажите =СсылкаНаДокумент.Номер; ответственного вручную."); Область Заголовок. Параметры. Дата = Формат(СсылкаНаДокумент.Дата,"ДЛФ=DD"); КонецЕсли; КонецЕсли; ТабДок.Вывести(ОбластьЗаголовок); КонецПроцедуры Запрос = Новый Запрос; Запрос. Текст = "ВЫБРАТЬ ПЕРВЫЕ 1 Код печати «РегистрацияКнигВБиблиотеке» &НаКлиенте Библиотека.ПолноеНаименование |ИЗ Процедура Обработка Команды (Параметр Команды, Справочник. Библиотека КАК Библиотека |АВТОУПОРЯДОЧИВАНИЕ"; ПараметрыВыполненияКоманды) ТабДок = Новый Табличный Документ; Результат = Запрос.Выполнить(); Печать (ТабДок, ПараметрКоманды); Выборка = Результат.Выбрать(); ТабДок.Показать("Печать Регистрации книг"); Если Выборка. Следующий () Тогда НазваниеБиблиотеки = КонецПроцедуры Выборка.ПолноеНаименование; Иначе &НаСервере НазваниеБиблиотеки = ""; Процедура Печать(ТабДок, СсылкаНаДокумент) КонецЕсли; Область Текст. Параметры. Название Бибилиотеки Макет = Документы. Приход Книг. Получить Макет ("Регистра = НазваниеБиблиотеки; цияКнигВБиблиотеке");

ТекущийПользователь = ИначеЕсли ПользователиИнформационнойБазы. ТекущийПоль ДокументПриходКниг.Закупка = Перечисления. Выбор Закупки. Поставщик Тогда зователь(); Сотрудник = НаименованиеОрганизации = Справочники. Сотрудники. Найти По Реквизиту ("Лог ДокументПриходКниг.Поставщик.Наименование; ин", ТекущийПользователь.Имя); ИначеЕсли ДокументПриходКниг.Закупка = Если Сотрудник <> Неопределено И Перечисления.ВыборЗакупки.Издатель Тогда ЗначениеЗаполнено(Сотрудник.Отдел) Тогда НаименованиеОрганизации = НазваниеОтдела = ДокументПриходКниг.Издатель.Наименование; Сотрудник.Отдел.Наименование; КонецЕсли; Иначе Область Текст. Параметры. Источник = НазваниеОтдела = ""; КонецЕсли; НаименованиеОрганизации; Область Текст. Параметры. Структурное Подразделен Если ДокументПриходКниг.Основание <> ие = НазваниеОтдела; Неопределено Тогда ДокументОснование = ДокументПриходКниг.Основание.ПолучитьОбъект ДокументПриходКниг = ();СсылкаНаДокумент.ПолучитьОбъект(); Область Текст. Параметры. Основание = "Документ №" + ДокументОснование.Номер + " от " + // Определяем наименование НаименованиеОрганизации = ""; Формат(ДокументОснование.Дата, "ДЛФ=DD");  $\Phi MO = ""$ ; Иначе КонецЕсли: Если ДокументПриходКниг.ИсточникПолучения = ТабДок.Вывести(ОбластьТекст); Перечисления. Источник Получения. Пожертвование Тогда ТабДок.Вывести(ОбластьШапкаТаблици); // Получаем данные о пожертвовании Пожертвование = Запрос = Новый Запрос; ДокументПриходКниг.ДанныеОПожертвований.По Запрос.Текст = лучитьОбъект(); "ВЫБРАТЬ ФИО = Пожертвование. Фамилия ПриходКнигКниги.НомерСтроки + " " + Пожертвование.Имя + " " + КАК НомерСтроки, Пожертвование. Отчество; ПриходКнигКниги.Книга КАК НаименованиеОрганизации = Книга, "Пожертвование №" + Пожертвование.Номер + " от ПриходКнигКниги.Количество " + ФИО: КАК Количество,

ПриходКнигКниги.Цена КАК	
Цена,	ТабДок.Вывести(ОбластьТаблица);
ПриходКнигКниги.Сумма КАК	КонецЦикла;
Сумма	
ИЗ	
Документ.ПриходКниг.Книги КАК	ОбластьПодвалТаблици.Параметры.ВсегоКоличест
ПриходКнигКниги	во = СуммаКоличство;
ГДЕ	
ПриходКнигКниги.Ссылка =	ОбластьПодвалТаблици.Параметры.ВсегоСумма =
&СсылкаНаДокумент";	СуммаВсего;
	ТабДок.Вывести(ОбластьПодвалТаблици);
Запрос. УстановитьПараметр("СсылкаНаДокумент",	
СсылкаНаДокумент);	ТабДок.Вывести(ОбластьПодвал);
РезультатЗапроса = Запрос.Выполнить();	КонецПроцедуры
ВыборкаДетальныеЗаписи =	Код модуля объекта документа «СписаниеКниг»
РезультатЗапроса.Выбрать();	
СуммаКоличство = 0;	Процедура ОбработкаПроведения(Отказ, Режим)
СуммаВсего = 0;	
_	// manyaning Octoryay Vyyyy Docycory
Пока	// регистр ОстаткиКниг Расход
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл	// регистр ОстаткиКниг Расход Движения.ОстаткиКниг.Записывать =
	Движения.ОстаткиКниг.Записывать =
ВыборкаДетальныеЗаписи.Следующий() Цикл	Движения.ОстаткиКниг.Записывать = Истина;
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки =	Движения.ОстаткиКниг.Записывать = Истина; Для Каждого ТекСтрокаКниги Из Книги
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;	Движения.ОстаткиКниг.Записывать = Истина; Для Каждого ТекСтрокаКниги Из Книги Цикл ОстатокКниг =
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки; ОбластьТаблица.Параметры.Книга =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга);
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки; ОбластьТаблица.Параметры.Книга =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если
ВыборкаДетальныеЗаписи.Следующий() Цикл ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки; ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда
ВыборкаДетальныеЗаписи.Следующий() Цикл  Область Таблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  Область Таблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  Область Таблица.Параметры.Количество =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга);  Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина;
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена = ВыборкаДетальныеЗаписи.Цена;	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина; КонецЕсли;
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена = ВыборкаДетальныеЗаписи.Цена;  ОбластьТаблица.Параметры.Сумма =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина; КонецЕсли; Движение =
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена = ВыборкаДетальныеЗаписи.Цена;  ОбластьТаблица.Параметры.Сумма = ВыборкаДетальныеЗаписи.Сумма;	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина; КонецЕсли; Движение = Движения.ОстаткиКниг.Добавить();
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена = ВыборкаДетальныеЗаписи.Цена;  ОбластьТаблица.Параметры.Сумма = ВыборкаДетальныеЗаписи.Сумма;  СуммаКоличство =	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина; КонецЕсли; Движение = Движения.ОстаткиКниг.Добавить(); Движение.ВидДвижения =
ВыборкаДетальныеЗаписи.Следующий() Цикл  ОбластьТаблица.Параметры.НомерСтроки = ВыборкаДетальныеЗаписи.НомерСтроки;  ОбластьТаблица.Параметры.Книга = ВыборкаДетальныеЗаписи.Книга;  ОбластьТаблица.Параметры.Количество = ВыборкаДетальныеЗаписи.Количество;  ОбластьТаблица.Параметры.Цена = ВыборкаДетальныеЗаписи.Цена;  ОбластьТаблица.Параметры.Сумма = ВыборкаДетальныеЗаписи.Сумма;  СуммаКоличство = СуммаКоличство+	Движения.ОстаткиКниг.Записывать = Истина;  Для Каждого ТекСтрокаКниги Из Книги Цикл  ОстатокКниг = ПолучитьОстаток(ТекСтрокаКниги.Книга); Если ОстатокКниг<ТекСтрокаКниги.Количество Тогда Сообщить(" Не достаточно книг "+ТекСтрокаКниги.Книга+" Доступный остаток: "+ОстатокКниг);  Отказ = Истина; КонецЕсли; Движение = Движения.ОстаткиКниг.Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Количество = Если Не ТекСтрокаКниги.Количество; ЗначениеЗаполнено(ЭтотОбъект.Ответственный) КонецЦикла; Тогда ЭтотОбъект.Ответственный = // регистр ДоступныеКниги Расход ПользователиИнформационнойБазы. ТекущийПоль Движения. Доступные Книги. Записывать = зователь(); Истина; Для Каждого ТекСтрокаКниги Из Книги КонепЕсли: Цикл КонецПроцедуры Движение = Движения. Доступные Книги. Добавить(); Процедура ОбработкаЗаполнения(ДанныеЗаполнения, Движение.ВидДвижения = ВидДвиженияНакопления.Расход; Стандартная Обработка) Движение.Период = Дата; Движение.Книги = Если ТипЗнч(ДанныеЗаполнения) = ТекСтрокаКниги.Книга; Тип("ДокументСсылка.ВыкупУтеряннойКниги") Движение.Количество = Тогда ТекСтрокаКниги.Количество; // Заполнение шапки КонецЦикла; Для Каждого ТекСтрокаКниги Из ДанныеЗаполнения.Книги Цикл НоваяСтрока = КонецПроцедуры Книги.Добавить(); НоваяСтрока.Книга = Функция ПолучитьОстаток(Книга) ТекСтрокаКниги.Книга; НоваяСтрока.Количество Результат =0; = ТекСтрокаКниги.Количество; Отбор = Новый Структура; Отбор.Вставить("Книги", Книга); НоваяСтрока.ПричинаСписания = НайденныеЗначения = Перечисления. Причина Списания. Утрата; РегистрыНакопления.ОстаткиКниг.Остатки(Дата,О КонецЦикла; тбор); КонецЕсли; Если НайденныеЗначения.Количество()>0 Тогда КонецПроцедуры Результат = НайденныеЗначения[0].Количество; Код формы документа «СписаниеКниг» КонецЕсли; Возврат Результат; &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, КонецФункции Текущий Объект, Параметры Записи)

Процедура ПриЗаписи(Отказ)

ТекущийПользователь = ПользователиИнформационнойБазы. ТекущийПоль Макет = зователь(); Документы.СписаниеКниг.ПолучитьМакет("Списа ниеКниг"); Область Чердак = // Поиск сотрудника по логину только если поле "Ответственный" не заполнено Макет.ПолучитьОбласть("Чердак"); Если Не ОбластьЗаголовок = ЗначениеЗаполнено(ТекущийОбъект.Ответственны Макет.ПолучитьОбласть("Заголовок"); ОбластьТекст = й) Тогда Найденный Сотрудник = Макет.ПолучитьОбласть("Текст"); Справочники. Сотрудники. Найти По Реквизиту ("Лог ОбластьШапкаТаблици = Макет.ПолучитьОбласть("ШапкаТаблици"); ин", ТекущийПользователь.Имя); ОбластьТаблица = Если Найденный Сотрудник <> Макет.ПолучитьОбласть("Таблица"); ОбластьПодвалТаблици = Неопределено Тогда Макет.ПолучитьОбласть("ПодвалТаблици"); ОбластьПодвал = ТекущийОбъект.Ответственный = Найденный Сотрудник; Макет.ПолучитьОбласть("Подвал"); Иначе Сообщить ("Сотрудник не Область Чердак. Параметры. Дата = найден по логину текущего пользователя. Укажите Формат(СсылкаНаДокумент.Дата,"ДЛФ=DD"); ответственного вручную."); Запрос = Новый Запрос; КонецЕсли; Запрос.Текст = КонецЕсли; "ВЫБРАТЬ ПЕРВЫЕ 1 Библиотека. Директор. Фамилия КАК КонецПроцедуры ДиректорФамилия, Библиотека. Директор. Наименование Код печати «ПечатьСписанияКниг» КАК ДиректорИмя, Библиотека. Директор. Отчество КАК &НаКлиенте ДиректорОтчество |ИЗ Процедура Обработка Команды (Параметр Команды, ПараметрыВыполненияКоманды) Справочник. Библиотека КАК Библиотека ТабДок = Новый Табличный Документ; |АВТОУПОРЯДОЧИВАНИЕ"; Печать (ТабДок, ПараметрКоманды); ТабДок.Показать("Печать Акта Результат = Запрос.Выполнить(); Выборка = Результат.Выбрать(); Пожертвования"); КонецПроцедуры Если Выборка. Следующий () Тогда ФИОРуководителя = &НаСервере Выборка. Директор Фамилия + " " +

Процедура Печать(ТабДок, СсылкаНаДокумент)

```
Выборка. ДиректорИмя + " " +
                                                             Пока
Выборка. Директор Отчество;
                                                     ВыборкаДетальныеЗаписи.Следующий() Цикл
       Область Чердак. Параметры. Руководитель У
                                                             Область Таблица. Параметры. Количество =
чреждения = ФИОРуководителя;
                                                     ВыборкаДетальныеЗаписи.Количество;
       Иначе
                                                                    СуммаКоличство =
                                                     СуммаКоличство+
       Область Чердак. Параметры. Руководитель У
                                                     ВыборкаДетальныеЗаписи.Количество;
чреждения = "";
                                                             КонецЦикла;
       КонецЕсли;
                                                             // Получаем данные библиотеки
       ТабДок.Вывести(ОбластьЧердак);
                                                             Запрос = Новый Запрос;
       ОбластьЗаголовок.Параметры.Номер =
                                                             Запрос.Текст =
СсылкаНаДокумент.Номер;
                                                             "ВЫБРАТЬ ПЕРВЫЕ 1
       ОбластьЗаголовок.Параметры.Дата =
                                                             Библиотека.ПолноеНаименование
Формат(СсылкаНаДокумент.Дата,"ДЛФ=DD");
                                                             |ИЗ
       ТабДок.Вывести(ОбластьЗаголовок);
                                                             Справочник. Библиотека КАК
                                                     Библиотека
       Запрос = Новый Запрос;
                                                             |АВТОУПОРЯДОЧИВАНИЕ";
       Запрос.Текст =
       "ВЫБРАТЬ
                                                             Результат = Запрос.Выполнить();
              СписаниеКнигКниги.НомерСтроки
                                                             Выборка = Результат.Выбрать();
КАК НомерСтроки,
              СписаниеКнигКниги.Количество
                                                             Если Выборка. Следующий () Тогда
КАК Количество
                                                                    НазваниеБиблиотеки =
       |ИЗ
                                                     Выборка.ПолноеНаименование;
              Документ.СписаниеКниг.Книги
                                                             Иначе
КАК СписаниеКнигКниги
                                                                    НазваниеБиблиотеки = ""; // Или
       ГДЕ
                                                     другое значение по умолчанию
              СписаниеКнигКниги.Ссылка =
                                                             КонецЕсли;
&СсылкаНаДокумент";
                                                             Область Текст. Параметры. Название Бибили
       Запрос. Установить Параметр ("Ссылка На До
                                                     отеки = НазваниеБиблиотеки:
кумент", СсылкаНаДокумент);
                                                             ТекущийПользователь =
                                                     ПользователиИнформационнойБазы. ТекущийПоль
       РезультатЗапроса = Запрос.Выполнить();
                                                     зователь();
       ВыборкаДетальныеЗаписи =
                                                             Сотрудник =
РезультатЗапроса.Выбрать();
                                                     Справочники. Сотрудники. Найти ПоРеквизиту ("Лог
       СуммаКоличство = 0;
                                                     ин", ТекущийПользователь.Имя);
```

Если Сотрудник <> Неопределено И ЗначениеЗаполнено(Сотрудник.Отдел) Тогда Запрос. Установить Параметр ("Ссылка На До НазваниеОтдела = кумент", СсылкаНаДокумент); Сотрудник.Отдел.Наименование; Иначе РезультатЗапроса = Запрос.Выполнить(); НазваниеОтдела = ""; КонецЕсли; ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать(); Область Текст. Параметры. Структурное Под СуммаКоличство = 0;разделение = НазваниеОтдела; СуммаВсего = 0; Пока ВыборкаДетальныеЗаписи.Следующий() Цикл Область Текст. Параметры. Дата = Формат(СсылкаНаДокумент.Дата,"ДЛФ=D"); Область Таблица. Параметры. Номер Строки Область Текст. Параметры. Номер = = ВыборкаДетальныеЗаписи.НомерСтроки; СсылкаНаДокумент.Номер; Область Таблица. Параметры. Книга Область Текст. Параметры. Количество = = ВыборкаДетальныеЗаписи.Книга; СуммаКоличство; Область Таблица. Параметры. Количество = Выборка Детальные Записи. Количество; ТабДок.Вывести(ОбластьТекст); СуммаКоличство = СуммаКоличство+ ТабДок.Вывести(ОбластьШапкаТаблици); ВыборкаДетальныеЗаписи.Количество; Запрос = Новый Запрос; Область Таблица. Параметры. Причина Спис Запрос.Текст = ания = "ВЫБРАТЬ Выборка Детальные Записи. Причина Списания; СписаниеКнигКниги.НомерСтроки ТабДок.Вывести(ОбластьТаблица); КАК НомерСтроки, КонецЦикла; СписаниеКнигКниги.Книга КАК Книга, Область Подвал Таблици. Параметры. Всего К оличество = СуммаКоличство; СписаниеКнигКниги.ПричинаСписания КАК ПричинаСписания, ТабДок.Вывести(ОбластьПодвалТаблици); СписаниеКнигКниги.Количество КАК Количество ТабДок.Вывести(ОбластьПодвал); |ИЗ Документ.СписаниеКниг.Книги КонецПроцедуры КАК СписаниеКнигКниги ГДЕ Код модуля объекта «ЗакупкаКниг» СписаниеКнигКниги.Ссылка = Процедура ПриЗаписи(Отказ) &СсылкаНаДокумент";

Если Не Движение.Книга = ЗначениеЗаполнено(ЭтотОбъект.Ответственный) ТекСтрокаКниги.Книга; Тогда Движение.ВыборЗакупки = ЭтотОбъект.Ответственный = ВыборЗакупки; ПользователиИнформационнойБазы. ТекущийПоль Движение.Поставщик = зователь(); Поставщик; КонецЕсли; Движение.Издатель = Издатель; Движение.Количество = КонецПроцедуры ТекСтрокаКниги.Количество; КонецЦикла; Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения) // регистр СредняяСобестоимостьТоваров СуммаДокумента =0; Приход Для каждого Табличная Часть из Книги Движения.СредняяСобестоимостьТоваров. Цикл Записывать = Истина; СуммаДокумента = Для Каждого ТекСтрокаКниги Из Книги СуммаДокумента +ТабличнаяЧасть.Сумма; Цикл КонецЦикла; Движение = Если Не Движения.СредняяСобестоимостьТоваров.Добавит ЗначениеЗаполнено(ЭтотОбъект.Ответственный) ь(); Тогда Движение.ВидДвижения = ЭтотОбъект.Ответственный = ВидДвиженияНакопления.Приход; ПользователиИнформационнойБазы. ТекущийПоль Движение.Период = Дата; зователь(); Движение.Книга = КонецЕсли; ТекСтрокаКниги.Книга; Движение.Сумма = КонецПроцедуры ТекСтрокаКниги.Сумма; Движение.Количество = Процедура ОбработкаПроведения(Отказ, Режим) ТекСтрокаКниги.Количество; КонецЦикла; // регистр ОборотЗаказовКниг Приход Движения. Оборот Заказов Книг. Записывать // регистр Взаиморасчеты = Истина: Движения.Взаиморасчеты.Записывать=Ист Для Каждого ТекСтрокаКниги Из Книги ина; Цикл Движение = Движение = Движения.Взаиморасчеты.Добавить(); Движения. Оборот Заказов Книг. Добавить(); Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Движение.Период=Дата; Движение.Период = Дата; Движение.Поставщик = Поставщик;

Движение.Сумма = СуммаДокумента;

Элементы. Излатель. Вилимость =

КонецПроцедуры Ложь; ИначеЕсли Объект.ВыборЗакупки = Перечисления.ВыборЗакупки.Издатель тогда Код формы документа «ЗакупкаКниг» &НаСервере Элементы.Поставщик.Видимость = Процедура Ложь; ВыборЗакупкиПриИзмененииНаСервере() Элементы.Издатель.Видимость = Истина: // Очищаем оба поля перед изменением КонецЕсли; видимости Объект. Поставщик = 0; КонецПроцедуры Объект. Издатель = 0; &НаКлиенте Если Объект.ВыборЗакупки = Процедура ПриОткрытии(Отказ) Перечисления. Выбор Закупки. Поставщик Тогда Элементы.Поставщик.Видимость = ПриОткрытииНаСервере(); Истина; КонецПроцедуры Элементы. Издатель. Видимость = Ложь; ИначеЕсли Объект.ВыборЗакупки = &НаКлиенте Перечисления. Выбор Закупки. Издатель Тогда Процедура КнигиЦенаПриИзменении(Элемент) Элементы.Поставщик.Видимость = Ложь; СтрокаТабличнойЧасти = Элементы. Издатель. Видимость = Элементы.Книги.ТекущиеДанные; Истина; СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество\*СтрокаТаблич КонецЕсли; ной Части. Цена; КонецПроцедуры &НаКлиенте КонецПроцедуры Процедура ВыборЗакупкиПриИзменении(Элемент) &НаКлиенте ВыборЗакупкиПриИзмененииНаСервере(); Процедура КнигиСуммаПриИзменении(Элемент) КонецПроцедуры СтрокаТабличнойЧасти = Элементы.Книги.ТекущиеДанные; &НаСервере Если СтрокаТабличнойЧасти.Количество = Процедура ПриОткрытииНаСервере() 0 тогда СтрокаТабличнойЧасти.ЦенаЗаШ Если Объект.ВыборЗакупки = = 0:

Иначе

Перечисления.ВыборЗакупки.Поставщик тогда

Истина:

Элементы. Поставшик. Видимость =

СтрокаТабличнойЧасти.Цена = СтрокаТабличнойЧасти.Сумма / СтрокаТабличнойЧасти.Количество; КонецЕсли; ТекущийПользователь = ПользователиИнформационнойБазы. ТекущийПоль КонецПроцедуры зователь(); &НаКлиенте // Поиск сотрудника по логину только если поле "Ответственный" не заполнено Процедура КнигиКоличествоПриИзменении(Элемент) Еспи Не ЗначениеЗаполнено(ТекущийОбъект.Ответственны СтрокаТабличнойЧасти = й) Тогда Элементы.Книги.ТекущиеДанные; НайденныйСотрудник = СтрокаТабличнойЧасти.Сумма = Справочники.Сотрудники.НайтиПоРеквизиту("Лог СтрокаТабличнойЧасти.Количество\*СтрокаТаблич ин", ТекущийПользователь.Имя); ной Части. Цена; Если Найденный Сотрудник <> КонецПроцедуры Неопределено Тогда &НаКлиенте ТекущийОбъект.Ответственный = Процедура КнигиКнигаПриИзменении(Элемент) Найденный Сотрудник; Иначе СтрокаТЧ = Сообщить("Сотрудник не Элементы.Книги.ТекущиеДанные; найден по логину текущего пользователя. Укажите СтрокаТЧ.Цена=ОбщийМодульФункции.Ц ответственного вручную."); еныИзРегистраСведений(Объект.Дата,СтрокаТЧ.К КонецЕсли; КонецЕсли: нига); КонецПроцедуры КонецПроцедуры Код печати «ТоварнаяНакладная» &НаСервере &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, Процедура ТекущийОбъект, ПараметрыЗаписи) ВыборЗакупкиПриИзмененииНаСервере() // Проверяем, заполнено ли поле "Статус" // Очищаем оба поля перед изменением Если Не видимости ЗначениеЗаполнено(ТекущийОбъект.Статус) Тогда Объект. Поставщик = 0; // Устанавливаем статус Объект.Издатель = 0; "Сформирован" ТекущийОбъект.Статус = Если Объект.ВыборЗакупки =

Перечисления. Выбор Закупки. Поставщик Тогда

Перечисления. Статус Закупки. Сформирован;

КонецЕсли;

Элементы. Поставшик. Вилимость = ПриОткрытииНаСервере(); Истина; Элементы. Издатель. Видимость = КонецПроцедуры Ложь; ИначеЕсли Объект.ВыборЗакупки = &НаКлиенте Перечисления. Выбор Закупки. Издатель Тогда Процедура КнигиЦенаПриИзменении(Элемент) Элементы.Поставщик.Видимость = Ложь: СтрокаТабличнойЧасти = Элементы.Издатель.Видимость = Элементы.Книги.ТекущиеДанные; Истина; СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество\*СтрокаТаблич КонецЕсли; ной Части. Цена; КонецПроцедуры &НаКлиенте КонецПроцедуры Процедура ВыборЗакупкиПриИзменении(Элемент) &НаКлиенте ВыборЗакупкиПриИзмененииНаСервере(); Процедура КнигиСуммаПриИзменении(Элемент) КонецПроцедуры СтрокаТабличнойЧасти = Элементы.Книги.ТекущиеДанные; &НаСервере Если Строка Табличной Части. Количество = Процедура ПриОткрытииНаСервере() 0 тогда СтрокаТабличнойЧасти.ЦенаЗаШ = 0;Если Объект.ВыборЗакупки = Перечисления. Выбор Закупки. Поставщик тогда Иначе Элементы. Поставщик. Видимость = СтрокаТабличнойЧасти.Цена = Истина: СтрокаТабличнойЧасти.Сумма / Элементы.Издатель.Видимость = СтрокаТабличнойЧасти.Количество; Ложь: КонецЕсли: ИначеЕсли Объект.ВыборЗакупки = Перечисления.ВыборЗакупки.Издатель тогда КонецПроцедуры Элементы. Поставщик. Видимость = Ложь; &НаКлиенте Элементы. Издатель. Видимость = Процедура Истина: КнигиКоличествоПриИзменении(Элемент) КонепЕсли: СтрокаТабличнойЧасти = КонецПроцедуры Элементы.Книги.ТекущиеДанные; СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество\*СтрокаТаблич &НаКлиенте

ной Части. Цена;

Процедура ПриОткрытии(Отказ)

Если НайденныйСотрудник <> КонецПроцедуры Неопределено Тогда &НаКлиенте Процедура КнигиКнигаПриИзменении(Элемент) ТекущийОбъект.Ответственный = Найденный Сотрудник; СтрокаТЧ = Иначе Элементы.Книги.ТекущиеДанные; Сообщить ("Сотрудник не СтрокаТЧ.Цена=ОбщийМодульФункции.Ц найден по логину текущего пользователя. Укажите еныИзРегистраСведений(Объект.Дата,СтрокаТЧ.К ответственного вручную."); КонецЕсли; нига); КонецЕсли; КонецПроцедуры КонецПроцедуры &НаСервере Код модуля документа «Пожертвования» Процедура ПередЗаписьюНаСервере(Отказ, Процедура ПриЗаписи(Отказ) ТекущийОбъект, ПараметрыЗаписи) Если Не // Проверяем, заполнено ли поле "Статус" ЗначениеЗаполнено(ЭтотОбъект.Ответственный) Если Не ЗначениеЗаполнено(ТекущийОбъект.Статус) Тогда ЭтотОбъект.Ответственный = // Устанавливаем статус ПользователиИнформационнойБазы. ТекущийПоль "Сформирован" зователь(); ТекущийОбъект.Статус = КонецЕсли; Перечисления. Статус Закупки. Сформирован; КонецЕсли; КонецПроцедуры Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения) ТекущийПользователь = СуммаДокумента =0; ПользователиИнформационнойБазы. ТекущийПоль Для каждого Табличная Часть из Книги зователь(); Цикл СуммаДокумента = // Поиск сотрудника по логину только если СуммаДокумента +ТабличнаяЧасть.Сумма; поле "Ответственный" не заполнено КонепПикла: Если Не Если Не ЗначениеЗаполнено(ТекущийОбъект.Ответственны ЗначениеЗаполнено(ЭтотОбъект.Ответственный) й) Тогда Тогда Найденный Сотрудник = Справочники.Сотрудники.НайтиПоРеквизиту("Лог ЭтотОбъект.Ответственный = ин", ТекущийПользователь.Имя); ПользователиИнформационнойБазы. ТекущийПоль зователь();

КонецЕсли;

КонецПроцедуры КонецПроцедуры Процедура ОбработкаПроведения(Отказ, Режим) &НаСервере Движения. Обороты Пожертвований. Записы Процедура ПриОткрытииНаСервере() вать = Истина; Движение = Если Объект.ВыборЗакупки = Движения. Обороты Пожертвований. Добавить(); Перечисления.ВыборЗакупки.Поставщик тогда Движение.ВидДвижения = Элементы.Поставщик.Видимость = ВидДвиженияНакопления.Приход; Истина; Движение.Период = Дата; Элементы.Издатель.Видимость = Движение. Документ Пожертвования = Ложь; ЭтотОбъект.Ссылка; ИначеЕсли Объект.ВыборЗакупки = Движение.Количество = 1; Перечисления.ВыборЗакупки.Издатель тогда Элементы.Поставщик.Видимость = КонецПроцедуры Ложь; Элементы. Издатель. Видимость = Код формы документа «Пожертвования» Истина; &НаСервере КонецЕсли; Процедура ВыборЗакупкиПриИзмененииНаСервере() КонецПроцедуры Если Объект.ВыборЗакупки = &НаКлиенте Перечисления. Выбор Закупки. Поставщик тогда Процедура ПриОткрытии(Отказ) Элементы. Поставщик. Видимость = Истина; ПриОткрытииНаСервере(); Элементы.Издатель.Видимость = Ложь: КонецПроцедуры ИначеЕсли Объект.ВыборЗакупки = Перечисления.ВыборЗакупки.Издатель тогда &НаКлиенте Элементы.Поставщик.Видимость = Процедура КнигиЦенаПриИзменении(Элемент) Ложь; Элементы. Издатель. Видимость = СтрокаТабличнойЧасти = Истина: Элементы.Книги.ТекущиеДанные; КонецЕсли: СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество\*СтрокаТаблич КонецПроцедуры ной Части. Цена; &НаКлиенте КонецПроцедуры Процедура ВыборЗакупкиПриИзменении(Элемент) &НаКлиенте

Процедура КнигиСуммаПриИзменении(Элемент)

ВыборЗакупкиПриИзмененииНаСервере();

ТекущийПользователь = СтрокаТабличнойЧасти = ПользователиИнформационнойБазы.ТекущийПоль Элементы.Книги.ТекущиеДанные; зователь(); Если Строка Табличной Части. Количество = 0 тогда // Поиск сотрудника по логину только если СтрокаТабличнойЧасти.ЦенаЗаШ поле "Ответственный" не заполнено = 0;Если Не ЗначениеЗаполнено(ТекущийОбъект.Ответственны Иначе СтрокаТабличнойЧасти.Цена = й) Тогда СтрокаТабличнойЧасти.Сумма / НайденныйСотрудник = СтрокаТабличнойЧасти.Количество; Справочники.Сотрудники.НайтиПоРеквизиту("Лог ин", ТекущийПользователь.Имя); КонецЕсли; КонецПроцедуры Если Найденный Сотрудник <> Неопределено Тогда &НаКлиенте ТекущийОбъект.Ответственный = Процедура КнигиКоличествоПриИзменении(Элемент) Найденный Сотрудник; Иначе СтрокаТабличнойЧасти = Сообщить ("Сотрудник не Элементы.Книги.ТекущиеДанные; найден по логину текущего пользователя. Укажите СтрокаТабличнойЧасти.Сумма = ответственного вручную."); СтрокаТабличнойЧасти.Количество\*СтрокаТаблич КонецЕсли; КонецЕсли; ной Части. Цена; КонецПроцедуры Попытка &НаКлиенте ВалидацияПолей.ПроверитьЕmail(Текущий Объект.Почта); Процедура КнигиКнигаПриИзменении(Элемент) Исключение СтрокаТЧ = Сообщить(ОписаниеОшибки()); Отказ = Истина; Элементы.Книги.ТекущиеДанные; СтрокаТЧ.Цена=ОбщийМодульФункции.Ц Возврат; еныИзРегистраСведений(Объект.Дата,СтрокаТЧ.К КонецПопытки; нига); Если НЕ КонецПроцедуры ВалидацияПолей.КорректноВведенНомерТелефона (ТекущийОбъект.Телефон) Тогда &НаСервере ВызватьИсключение

"Некорректный номер телефона. Пожалуйста,

введите номер в международном формате.";

КонецЕсли;

Процедура ПередЗаписьюНаСервере(Отказ,

ТекущийОбъект, ПараметрыЗаписи)

Библиотека. Директор. Наименование КАК

КонецПроцедуры ДиректорИмя, Код печати «ПечатьАктаПожертвования» Библиотека. Директор. Отчество КАК &НаКлиенте ДиректорОтчество Процедура Обработка Команды (Параметр Команды, |ИЗ ПараметрыВыполненияКоманды) Справочник. Библиотека КАК Библиотека |АВТОУПОРЯДОЧИВАНИЕ"; ТабДок = Новый Табличный Документ; Печать (ТабДок, ПараметрКоманды); Результат = Запрос.Выполнить(); ТабДок.Показать("Печать Акта Выборка = Результат.Выбрать(); Пожертвования"); Если Выборка. Следующий () Тогда КонецПроцедуры ФИОРуководителя = Выборка. Директор Фамилия + " " + &НаСервере Выборка. ДиректорИмя + " " + Процедура Печать(ТабДок, СсылкаНаДокумент) Выборка. Директор Отчество; Макет = Область Чердак. Параметры. Руководитель Учрежден Документы.Пожертвование.ПолучитьМакет("АктП ия =  $\Phi$ ИОРуководителя; ожертвования"); Иначе Область Чердак = Макет.ПолучитьОбласть("Чердак"); Область Чердак. Параметры. Руководитель Учрежден ОбластьЗаголовок = :''' = ки Макет.ПолучитьОбласть("Заголовок"); КонецЕсли; ОбластьТекст = Макет.ПолучитьОбласть("Текст"); ТабДок.Вывести(ОбластьЧердак); ОбластьШапкаТаблици = Макет.ПолучитьОбласть("ШапкаТаблици"); ОбластьЗаголовок.Параметры.Номер = ОбластьТаблица = СсылкаНаДокумент.Номер; Макет.ПолучитьОбласть("Таблица"); Область Заголовок. Параметры. Дата = ОбластьПодвалТаблици = Формат(СсылкаНаДокумент.Дата,"ДЛФ=DD"); Макет.ПолучитьОбласть("ПодвалТаблици"); ТабДок.Вывести(ОбластьЗаголовок); ОбластьПодвал = Макет.ПолучитьОбласть("Подвал"); Запрос = Новый Запрос; Запрос.Текст = Область Чердак. Параметры. Дата = "ВЫБРАТЬ Формат(СсылкаНаДокумент.Дата,"ДЛФ=DD"); ПожертвованиеКниги.Количество Запрос = Новый Запрос; КАК Количество, Запрос.Текст = ПожертвованиеКниги.Цена КАК "ВЫБРАТЬ ПЕРВЫЕ 1 Цена, Библиотека. Директор. Фамилия КАК ПожертвованиеКниги.Сумма КАК ДиректорФамилия, Сумма

|ИЗ |АВТОУПОРЯДОЧИВАНИЕ"; Документ.Пожертвование.Книги КАК ПожертвованиеКниги Результат = Запрос.Выполнить(); ГДЕ Выборка = Результат.Выбрать(); ПожертвованиеКниги.Ссылка = &СсылкаНаДокумент"; Если Выборка. Следующий () Тогда НазваниеБиблиотеки = Запрос. Установить Параметр ("Ссылка На До Выборка.ПолноеНаименование; кумент", СсылкаНаДокумент); Иначе НазваниеБиблиотеки = ""; // Или другое РезультатЗапроса = Запрос.Выполнить(); значение по умолчанию КонецЕсли; ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать(); Область Текст. Параметры. Название Бибилиотеки СуммаКоличство = 0; = НазваниеБиблиотеки: СуммаВсего = 0; Пока ВыборкаДетальныеЗаписи.Следующий() Цикл ТекущийПользователь = ПользователиИнформационнойБазы. ТекущийПоль Область Таблица. Параметры. Количество = зователь(); ВыборкаДетальныеЗаписи.Количество; Сотрудник = Область Таблица. Параметры. Цена Справочники.Сотрудники.НайтиПоРеквизиту("Лог = ВыборкаДетальныеЗаписи.Цена; ин", ТекущийПользователь.Имя); Область Таблица. Параметры. Сумма = Если Сотрудник <> Неопределено И ВыборкаДетальныеЗаписи.Сумма; ЗначениеЗаполнено(Сотрудник.Отдел) Тогда СуммаКоличство = НазваниеОтдела = СуммаКоличство+ Сотрудник.Отдел.Наименование; ВыборкаДетальныеЗаписи.Количество; Иначе НазваниеОтдела = ""; // Или другое значение CуммаBсего = CуммаBсего + ВыборкаДетальныеЗаписи.Сумма; по умолчанию, например, "Общий отдел" КонецЦикла; КонецЕсли; Область Текст. Параметры. Структурное Подразделен // Получаем данные библиотеки ие = НазваниеОтдела; Запрос = Новый Запрос; Запрос.Текст = "ВЫБРАТЬ ПЕРВЫЕ 1 Библиотека.ПолноеНаименование Область Текст. Параметры. ФИО = И3 СсылкаНаДокумент.Фамилия + " " +

Справочник. Библиотека КАК Библиотека

СсылкаНаДокумент.Имя + " " + Запрос. Установить Параметр ("Ссылка На До СсылкаНаДокумент.Отчество; кумент", СсылкаНаДокумент); Область Текст. Параметры. Дата = Формат(СсылкаНаДокумент.Дата,"ДЛФ=D"); РезультатЗапроса = Запрос.Выполнить(); Область Текст. Параметры. Номер = СсылкаНаДокумент.Номер; ВыборкаДетальныеЗаписи = Область Текст. Параметры. Количество = РезультатЗапроса.Выбрать(); СуммаКоличство = 0; СуммаКоличство; Область Текст. Параметры. Сумма CуммаBсе $\Gamma$ о = 0; =СуммаВсего; Пока ОбластьТекст.Параметры.СуммаСловами = ВыборкаДетальныеЗаписи.Следующий() Цикл ЧислоПрописью(СуммаВсего, "Л=ru RU;ДП=Истина", Область Таблица. Параметры. Номер Строки "рубль,рубля,рублей,м,копейка,копейки,копеек,ж,2 = ВыборкаДетальныеЗаписи.НомерСтроки; Область Таблица. Параметры. Книга "); = ВыборкаДетальныеЗаписи.Книга; ТабДок.Вывести(ОбластьТекст); ОбластьТаблица.Параметры.Количество = ТабДок.Вывести(ОбластьШапкаТаблици); ВыборкаДетальныеЗаписи.Количество; Область Таблица. Параметры. Цена Запрос = Новый Запрос; = ВыборкаДетальныеЗаписи.Цена; Запрос. Текст = "ВЫБРАТЬ Область Таблица. Параметры. Сумма = ВыборкаДетальныеЗаписи.Сумма; СуммаКоличство = ПожертвованиеКниги. НомерСтроки КАК СуммаКоличство+ НомерСтроки, ПожертвованиеКниги.Книга КАК Выборка Детальные Записи. Количество; Книга. СуммаВсего = СуммаВсего + ПожертвованиеКниги.Количество ВыборкаДетальныеЗаписи.Сумма; КАК Количество, ТабДок.Вывести(ОбластьТаблица); ПожертвованиеКниги.Цена КАК КонецЦикла; Цена, Пожертвование Книги. Сумма КАК Область Подвал Таблици. Параметры. Всего К Сумма оличество = СуммаКоличство; |ИЗ Область Подвал Таблици. Параметры. Всего С Документ.Пожертвование.Книги умма = СуммаВсего; КАК ПожертвованиеКниги ТабДок.Вывести(ОбластьПодвалТаблици); ГДЕ ПожертвованиеКниги.Ссылка = ТабДок.Вывести(ОбластьПодвал); &СсылкаНаДокумент";

КонецПроцедуры

// регистр ДоступныеКниги Расход Код модуля объекта документа «ВыдачаКниг» Движения. Доступные Книги. Записывать = Функция ПолучитьОстаток(Книга) Истина; Результат =0; Для Каждого ТекСтрокаКниги Из Книги Отбор = Новый Структура; Цикл Отбор.Вставить("Книги", Книга); ДоступныеКниги = НайленныеЗначения = ПолучитьОстаток(ТекСтрокаКниги.Книга); РегистрыНакопления.ДоступныеКниги.Остатки(Да Если та,Отбор); ДоступныеКниги<ТекСтрокаКниги.Количество Если НайденныеЗначения.Количество()>0 Тогда Тогда Сообщить(" Не достаточно Результат = книг "+ТекСтрокаКниги.Книга+" Доступный НайденныеЗначения[0].Количество; остаток: "+ДоступныеКниги); КонепЕсли: Отказ = Истина: Возврат Результат; КонецЕсли; Движение = КонецФункции Движения. Доступные Книги. Добавить(); Движение.ВидДвижения = Процедура Обработка Проведения (Отказ, Режим) ВидДвиженияНакопления.Расход; Движение.Период = Дата; // регистр СрокВозвратаКниг **Движение.Книги** = Движения. Срок Возврата Книг. Записывать ТекСтрокаКниги.Книга; = Истина; Движение.Количество = Для Каждого ТекСтрокаКниги Из Книги ТекСтрокаКниги.Количество; Цикл КонецЦикла; Движение = // регистр ИсторияВыдачКниг Приход Движения.СрокВозвратаКниг.Добавить(); Движения. История Выдач Книг. Записывать Движение.Период = Дата; //МС() = Истина: Движение.Книга = Для Каждого ТекСтрокаКниги Из Книги ТекСтрокаКниги.Книга; Цикл Движение. Читатель = Читатель; Движение = Движение.ВремяДействия = Движения. История Выдач Книг. Добавить (); ЭтотОбъект.Дата; Движение.ВидДвижения = Движение.Количество = ВидДвиженияНакопления.Приход; Движение.Период = Дата; ТекСтрокаКниги.Количество; Движение.ДокументВыдачи = Движение.Книга = ТекСтрокаКниги.Книга; ЭтотОбъект.Ссылка; Движение. Читатель = Читатель; Движение.ПланируемаяДатаВозврата = Движение.Количество = ТекСтрокаКниги.ДатаВозвратаКниги; ТекСтрокаКниги.Количество;

КонецЦикла;

КонецЦикла;

// регистр ПопулярностьКниги Приход Если Не Движения.ПопулярностьКниги.Записывать ЗначениеЗаполнено(ЭтотОбъект.Ответственный) = Истина; Тогда Для Каждого ТекСтрокаКниги Из Книги ЭтотОбъект.Ответственный = Цикл ПользователиИнформационнойБазы. ТекущийПоль Движение = зователь(); Движения.ПопулярностьКниги.Добавить(); КонецЕсли; Движение.ВидДвижения = ВидДвиженияНакопления.Приход; КонецПроцедуры Движение.Период = Дата; Движение.Книга = &НаСервере ТекСтрокаКниги.Книга; Функция Проверить Читательский Билет (Читатель, Движение.Количество = Книги) ТекСтрокаКниги.Количество; КонецЦикла; // Проверка срока действия билета Запрос = Новый Запрос; Запрос.Текст = // Проверка читательского билета "ВЫБРАТЬ ПЕРВЫЕ 1 Если Не Проверить Читательский Билет (Читатель, Книги) Тогда Сроки Действия Читательских Билетов. Дата Окончан Отказ = Истина; ия Возврат; |ИЗ КонецЕсли; РегистрСведений.СрокиДействияЧитательскихБил КонецПроцедуры етов КАК СрокиДействия Читательских Билетов ГДЕ Процедура ПриЗаписи(Отказ) Сроки Действия Читательских Билетов. Читательский Билет. Читатель = & Читатель Если Не ЗначениеЗаполнено(ЭтотОбъект.Ответственный) УПОРЯДОЧИТЬ ПО Тогда ЭтотОбъект.Ответственный = СрокиДействияЧитательскихБилетов.Период ПользователиИнформационнойБазы. ТекущийПоль УБЫВ"; зователь(); КонецЕсли: Запрос. Установить Параметр ("Читатель", Читатель); КонецПроцедуры Результат = Запрос.Выполнить(); Выборка = Результат.Выбрать(); Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения) Если Выборка. Следующий () Тогда ДатаОкончанияБилета = Выборка. Дата Окончания;

Если ДатаОкончанияБилета < Дата КонецЕсли; Тогла Сообщить("Срок действия // Проверка возрастного рейтинга книг читательского билета истек!"); Для Каждого СтрокаКниги Из Книги Цикл Возврат Ложь; Если Не Проверить Возрастной Рейтинг (Читатель, КонецЕсли; Иначе СтрокаКниги.Книга, ВозрастнойРейтингБилета) Сообщить("У читателя нет Тогла читательского билета!"); Сообщить("Возрастной Возврат Ложь; рейтинг книги """ + СтрокаКниги.Книга + """ не соответствует возрастному рейтингу читательского КонецЕсли; // Получение возрастного рейтинга билета."); читательского билета Возврат Ложь; Запрос = Новый Запрос; КонецЕсли; Запрос.Текст = КонепПикла: "ВЫБРАТЬ ПЕРВЫЕ 1 Возврат Истина; Читательский Билет. Возрастной Рейтинг КонецФункции КАК Возрастной Рейтинг |ИЗ &НаСервере Документ. Читательский Билет КАК Функция Проверить Возрастной Рейтинг (Читатель, **Читательский**Билет Книга, Возрастной Рейтинг Билета) ГДЕ Читательский Билет. Читатель = ВозрастЧитателя = &Читатель СколькоЛетТебе(ТекущаяДата(), УПОРЯДОЧИТЬ ПО Читатель. Дата Рождения); Читательский Билет. Дата УБЫВ"; РейтингКниги = Книга.ВозрастнойРейтинг; Запрос. Установить Параметр ("Читатель", Если Возрастной Рейтинг Билета = Читатель); Перечисления. Возрастной Рейтинг. Без Ограничений Результат = Запрос.Выполнить(); Тогда Выборка = Результат.Выбрать(); Возврат Истина; ИначеЕсли ВозрастнойРейтингБилета = Если Выборка. Следующий () Тогда Перечисления. Возрастной Рейтинг. Для Детей Отблет ВозрастнойРейтингБилета = И (РейтингКниги = Выборка. Возрастной Рейтинг; Перечисления.ВозрастнойРейтинг.ДляДетейОт6лет Иначе ИЛИ РейтингКниги = Сообщить("У читателя нет Перечисления. Возрастной Рейтинг. Без Ограничений читательского билета или у него не указан ) Тогда возрастной рейтинг!"); // Более информативное Возврат Истина; сообшение

Возврат Ложь;

ИначеЕсли ВозрастнойРейтингБилета = Возврат 0; Перечисления.ВозрастнойРейтинг.ДляДетейОт12Л КонецЕсли; ет И (РейтингКниги = Возраст =  $\Gamma$ од(ТекДата) -Перечисления. Возрастной Рейтинг. Для Детей Отблет Год(ДатаРождения); ИЛИ РейтингКниги = Если Месяц(ТекДата) < Перечисления. Возрастной Рейтинг. Для Детей От 12 Л Месяц(ДатаРождения) Или (Месяц(ТекДата) = Месяц(ДатаРождения) И День(ТекДата) < ет ИЛИ РейтингКниги = День(ДатаРождения)) Тогда Перечисления. Возрастной Рейтинг. Без Ограничений Возраст = Возраст - 1;) Тогда КонецЕсли; Возврат Истина; ИначеЕсли ВозрастнойРейтингБилета = Возврат Возраст; Перечисления. Возрастной Рейтинг. Для Подростков От16Лет КонецФункции И (РейтингКниги = Перечисления. Возрастной Рейтинг. Для Детей Отблет Код формы документа «ВыдачаКниг» ИЛИ РейтингКниги = &НаСервере Перечисления. Возрастной Рейтинг. Для Детей От 12 Л Процедура ПередЗаписьюНаСервере(Отказ, Текущий Объект, Параметры Записи) ет ИЛИ РейтингКниги = Перечисления. Возрастной Рейтинг. Для Подростков Для Каждого СтрокаТЧ Из От16Лет ТекущийОбъект.Книги Цикл ИЛИ РейтингКниги = Если Перечисления. Возрастной Рейтинг. Без Ограничений СтрокаТЧ.ДатаВозвратаКниги > ) Тогда Добавить Месяц (Текущий Объект. Дата, 1) Тогда Возврат Истина; Сообщить("Срок выдачи ИначеЕсли ВозрастнойРейтингБилета = книги не может превышать один месяц!"); Перечисления. Возрастной Рейтинг. Для В зрослых Отказ = Истина; Тогда Возврат; // Прерываем Возврат Истина; // Взрослым выполнение, если ошибка найдена можно выдавать любые книги КонецЕсли; Иначе КонецЦикла; Возврат Ложь; КонецЕсли; Если Не ЗначениеЗаполнено(Объект.Ответственный) Тогда КонецФункции НайденныйСотрудник = Справочники. Сотрудники. Найти По Наименованию ( &НаСервере ПользователиИнформационнойБазы. ТекущийПоль

зователь().Имя);

Неопределено Тогда

Если Найденный Сотрудник <>

Функция СколькоЛетТебе(ТекДата, ДатаРождения)

Если ДатаРождения = Неопределено Тогда

Объект.Ответственный = // Поиск записи о выдаче данной книги этому читателю Найденный Сотрудник; КонецЕсли; ЗаписьВыдачи = КонецЕсли; НайтиЗаписьВыдачи(ТекСтрокаКниги.Книга, Читатель); Если ЗаписьВыдачи = ТекущийПользователь = Неопределено Тогда ПользователиИнформационнойБазы. ТекущийПоль Сообщить("Не найдена зователь(); запись о выдаче книги " + ТекСтрокаКниги.Книга + " читателю " + Читатель); Отказ = Истина; // Поиск сотрудника по логину только если поле "Ответственный" не заполнено Продолжить; // Переходим Если Не к следующей книге ЗначениеЗаполнено(ТекущийОбъект.Ответственны КонецЕсли; й) Тогда // Проверка количества Найденный Сотрудник = Если ЗаписьВыдачи.Количество < Справочники. Сотрудники. Найти По Реквизиту ("Лог ТекСтрокаКниги.Количество Тогда ин", ТекущийПользователь.Имя); Сообщить("Попытка вернуть больше книг, чем было выдано. Доступно Если Найденный Сотрудник <> для возврата: " + ЗаписьВыдачи.Количество); Неопределено Тогда Отказ = Истина: Продолжить; ТекущийОбъект.Ответственный = КонецЕсли; Найденный Сотрудник; // Создание движения по регистру Иначе Движение = Сообщить ("Сотрудник не Движения. История Выдач Книг. Добавить(); найден по логину текущего пользователя. Укажите Движение.ВидДвижения = ответственного вручную."); ВидДвиженияНакопления.Расход; КонецЕсли; Движение.Период = Дата; КонецЕсли; Движение.Книга = ТекСтрокаКниги.Книга; КонецПроцедуры Движение. Читатель = Читатель; Движение.Количество = Код модуля объекта документа «ВозвратКниг» ТекСтрокаКниги.Количество; КонецЦикла; &НаСервере Процедура ОбработкаПроведения(Отказ, Режим) // регистр ДоступныеКниги Приход Движения.ДоступныеКниги.Записывать = Движения. История Выдач Книг. Записывать Истина: = Истина; Для Каждого ТекСтрокаКниги Из Книги

Цикл

Для Каждого ТекСтрокаКниги Из Книги

Цикл

Движение = Движения. Доступные Книги. Добавить(); // регистр ШтрафыЗаПросрочку ЦенаЗаДеньПросрочки = Движение.ВидДвижения = ВидДвиженияНакопления.Приход; Общий Модуль Функции. Получить Цену Штрафа (Пе Движение.Период = Дата; речисления. Название Штрафа. Просрочка); Движения. ШтрафыЗаПросрочку. Записыват Движение.Книги = ТекСтрокаКниги.Книга;  $_{\rm b} =$ Истина; Движение.Количество = Для Каждого ТекСтрокаКниги Из Книги ТекСтрокаКниги.Количество; Цикл КонецЦикла; ЗаписьВыдачи = // регистр СрокВозвратаКниг НайтиДатуВыдачи(ТекСтрокаКниги.Книга); Движения.СрокВозвратаКниг.Записывать = Истина; Если Не ЗаписьВыдачи = Для Каждого ТекСтрокаКниги Из Книги Неопределено Тогда Цикл ПланируемаяДатаВозврата Движение = = ЗаписьВыдачи.ПланируемаяДатаВозврата; Движения.СрокВозвратаКниг.Добавить(); Движение. Период = Дата; //MC()// Расчет просроченных Движение.Книга = дней ТекСтрокаКниги.Книга; ПросроченныеДни = Движение. Читатель = Читатель; ?(Дата > ПланируемаяДатаВозврата, Цел((Дата -Движение.ВремяДействия = ПланируемаяДатаВозврата) / 86400), 0); ЭтотОбъект.Дата; Если ПросроченныеДни > 0 Тогла Движение.Количество = ТекСтрокаКниги.Количество; Движение = Движения. ШтрафыЗаПросрочку. Добавить(); Движение. Фактическая Дата Возврата = Движение.Период Дата; = Дата; Движение.ДокументВыдачи = ДокументОснования; Движение.Основание = ЭтотОбъект.Ссылка; // Заполнение основания ЗаписьВыдачи = ссылкой на текущий документ НайтиДатуВыдачи(ТекСтрокаКниги.Книга); Движение.Книга = ТекСтрокаКниги.Книга; Если Не ЗаписьВыдачи = Неопределено Тогда Движение. Читатель = Читатель; Движение.ПланируемаяДатаВозврата = Движение.Количество = ЗаписьВыдачи.ПланируемаяДатаВозврата; ТекСтрокаКниги.Количество; КонецЕсли; Движение.СуммаШтрафа =

КонецЦикла;

Просроченные Дни \* Цена За День Просрочки \* Возврат НайденныеЗначения; ТекСтрокаКниги.Количество; КонецФункции Движение.Оплачено = Ложь; КонецЕсли; Процедура КонецЕсли; ОбработкаЗаполнения(ДанныеЗаполнения, КонецЦикла; Стандартная Обработка) КонецПроцедуры Если ТипЗнч(ДанныеЗаполнения) = Тип("ДокументСсылка.ВыдачаКниг") Тогда // Функция для поиска записи о выдаче // Заполнение шапки Функция НайтиЗаписьВыдачи(Книга, Читатель) ДокументОснования = ДанныеЗаполнения.Ссылка; Отбор = Новый Структура("Книга, Читатель = Читатель", Книга, Читатель); ДанныеЗаполнения. Читатель; // Ищем запись с нужной книгой, Для Каждого ТекСтрокаКниги Из ДанныеЗаполнения.Книги Цикл читателем // и ПУСТЫМ ДокументВыдачи - то есть, НоваяСтрока = не возвращенную ранее Книги.Добавить(); Найденные Значения = НоваяСтрока.Книга = РегистрыНакопления.ИсторияВыдачКниг.Остатки( ТекСтрокаКниги.Книга; Дата, Отбор); НоваяСтрока.Количество = ТекСтрокаКниги.Количество; Если Найденные Значения. Количество() > 0 Тогда НоваяСтрока.КачествоКнигиПриВозврате Возврат НайденныеЗначения[0]; =ТекСтрокаКниги.КачествоКнигиПриВыдаче; Иначе КонепПикла: Возврат Неопределено; ИначеЕсли ТипЗнч(ДанныеЗаполнения) = КонецЕсли; Тип("ДокументСсылка.ПродлениеКниги") Тогда // Заполнение шапки КонецФункции ДокументОснования = ДанныеЗаполнения. Документ Основания; //Функция для поиска даты выдачи Читатель = Функция НайтиДатуВыдачи(Книга) ДанныеЗаполнения. Читатель; Для Каждого ТекСтрокаКниги Из Отбор = Новый Структура("Книга, ДанныеЗаполнения.Книги Цикл Читатель, ДокументВыдачи", Книга, Читатель, НоваяСтрока = ДокументОснования); Книги.Добавить(); НайденныеЗначения = НоваяСтрока.Книга = РегистрыСведений.СрокВозвратаКниг.ПолучитьПо ТекСтрокаКниги.Книга; следнее(Дата, Отбор); НоваяСтрока.Количество

= ТекСтрокаКниги.Количество;

КонецЦикла; Движение.Период = Дата; КонецЕсли; Движение.Книга = ТекСтрокаКниги.Книга; КонецПроцедуры Движение. Читатель = Читатель; Движение.Количество = Код модуля объекта документа «ВозвратКниг» ТекСтрокаКниги.Количество; &НаСервере КонецЦикла; Процедура ОбработкаПроведения(Отказ, Режим) // регистр ДоступныеКниги Приход Движения.ДоступныеКниги.Записывать = Движения. История Выдач Книг. Записывать Истина; = Истина; Для Каждого ТекСтрокаКниги Из Книги Для Каждого ТекСтрокаКниги Из Книги Цикл Цикл Движение = // Поиск записи о выдаче данной Движения. Доступные Книги. Добавить(); книги этому читателю Движение.ВидДвижения = ЗаписьВыдачи = ВидДвиженияНакопления.Приход; НайтиЗаписьВыдачи(ТекСтрокаКниги.Книга, Движение.Период = Дата; Читатель); Движение.Книги = ТекСтрокаКниги.Книга; Если ЗаписьВылачи = **Движение.Количество** = Неопределено Тогда ТекСтрокаКниги.Количество; Сообщить("Не найдена КонецЦикла; запись о выдаче книги " + ТекСтрокаКниги.Книга + // регистр СрокВозвратаКниг " читателю " + Читатель); Движения. Срок Возврата Книг. Записывать Отказ = Истина; = Истина; Продолжить; // Переходим Для Каждого ТекСтрокаКниги Из Книги к следующей книге Цикл КонецЕсли; Движение = // Проверка количества Движения.СрокВозвратаКниг.Добавить(); Если ЗаписьВыдачи.Количество < Движение.Период = Дата; //МС() ТекСтрокаКниги.Количество Тогда Движение.Книга = ТекСтрокаКниги.Книга; Сообщить("Попытка вернуть больше книг, чем было выдано. Доступно Движение. Читатель = Читатель; для возврата: " + ЗаписьВыдачи.Количество); Движение.ВремяДействия = Отказ = Истина: ЭтотОбъект.Дата; Движение.Количество = Продолжить; КонецЕсли; ТекСтрокаКниги.Количество; // Создание движения по регистру Движение = Движение. Фактическая Дата Возврата = Движения. История Выдач Книг. Добавить(); Дата; Движение.ВидДвижения = Движение.ДокументВыдачи = ВидДвиженияНакопления.Расход; ДокументОснования;

ЭтотОбъект.Ссылка; // Заполнение основания ЗаписьВыдачи = ссылкой на текущий документ НайтиДатуВыдачи(ТекСтрокаКниги.Книга); Движение.Книга = ТекСтрокаКниги.Книга; Если Не ЗаписьВыдачи = Неопределено Тогда Движение. Читатель = Читатель; Движение.ПланируемаяДатаВозврата = Движение.Количество = ЗаписьВыдачи.ПланируемаяДатаВозврата; ТекСтрокаКниги.Количество; КонецЕсли; Движение.СуммаШтрафа = Просроченные Дни \* ЦенаЗа День Просрочки \* КонецЦикла; ТекСтрокаКниги.Количество; // регистр ШтрафыЗаПросрочку ЦенаЗаДеньПросрочки = Движение.Оплачено = Ложь; Общий Модуль Функции. Получить Цену Штрафа (Пе КонецЕсли; КонецЕсли; речисления. Название Штрафа. Просрочка); Движения.ШтрафыЗаПросрочку.Записыват КонецЦикла; ь = Истина; Для Каждого ТекСтрокаКниги Из Книги КонецПроцедуры Цикл ЗаписьВыдачи = // Функция для поиска записи о выдаче НайтиДатуВыдачи(ТекСтрокаКниги.Книга); Функция НайтиЗаписьВыдачи(Книга, Читатель) Если Не ЗаписьВыдачи = Отбор = Новый Структура("Книга, Неопределено Тогда Читатель", Книга, Читатель); ПланируемаяДатаВозврата // Ищем запись с нужной книгой, = ЗаписьВыдачи.ПланируемаяДатаВозврата; читателем // и ПУСТЫМ ДокументВыдачи - то есть, не возвращенную ранее // Расчет просроченных дней НайденныеЗначения = ПросроченныеДни = РегистрыНакопления.ИсторияВыдачКниг.Остатки( ?(Дата > ПланируемаяДатаВозврата, Цел((Дата -Дата, Отбор); ПланируемаяДатаВозврата) / 86400), 0); Если Просроченные Дни > Если НайденныеЗначения.Количество() > 0 0 Тогда Тогда Движение = Возврат НайденныеЗначения[0]; Движения. ШтрафыЗаПросрочку. Добавить(); Иначе Возврат Неопределено; Движение.Период = Дата; КонецЕсли;

КонецФункции

Движение.Основание =

ДокументОснования = //Функция для поиска даты выдачи ДанныеЗаполнения. ДокументОснования; Читатель = Функция Найти Дату Выдачи (Книга) ДанныеЗаполнения. Читатель; Отбор = Новый Структура("Книга, Для Каждого ТекСтрокаКниги Из Читатель, ДокументВыдачи", Книга, Читатель, ДанныеЗаполнения.Книги Цикл ДокументОснования); НоваяСтрока = НайленныеЗначения = Книги.Добавить(); РегистрыСведений.СрокВозвратаКниг.ПолучитьПо НоваяСтрока.Книга = следнее(Дата, Отбор); ТекСтрокаКниги.Книга; НоваяСтрока.Количество = ТекСтрокаКниги.Количество; Возврат НайденныеЗначения; КонецЦикла; КонецФункции КонецЕсли; Процедура КонецПроцедуры ОбработкаЗаполнения(ДанныеЗаполнения, Стандартная Обработка) Код формы документа «ВозвратКниг» Если ТипЗнч(ДанныеЗаполнения) = Тип("ДокументСсылка.ВыдачаКниг") Тогда &НаСервере // Заполнение шапки Процедура ДокументОснования = ДокументОснованияПриИзмененииНаСервере() ДанныеЗаполнения.Ссылка; Читатель = Если Объект. Документ Основания = ДанныеЗаполнения. Читатель; Неопределено Тогда Для Каждого ТекСтрокаКниги Из Объект.Книги.Очистить(); // ДанныеЗаполнения.Книги Цикл Очищаем табличную часть, если основание не НоваяСтрока = выбрано Книги.Добавить(); Возврат; НоваяСтрока.Книга = КонецЕсли; ТекСтрокаКниги.Книга; НоваяСтрока.Количество Выдача = = ТекСтрокаКниги.Количество; Объект.ДокументОснования.ПолучитьОбъект(); НоваяСтрока.КачествоКнигиПриВозврате Объект. Читатель = Выдача. Читатель; =ТекСтрокаКниги.КачествоКнигиПриВыдаче; Объект.Книги.Очистить(); КонецЦикла; ИначеЕсли ТипЗнч(ДанныеЗаполнения) = Для Каждого СтрокаВыдачи Из Тип("ДокументСсылка.ПродлениеКниги") Тогда Выдача.Книги Цикл // Заполнение шапки НоваяСтрока =

Объект.Книги.Добавить();

НоваяСтрока.Книга = КонецПроцедуры СтрокаВыдачи.Книга; НоваяСтрока.Количество = СтрокаВыдачи.Количество; &НаКлиенте КонецЦикла; Процедура ЧитательПриИзменении(Элемент) КонецПроцедуры ЧитательПриИзмененииНаСервере(); КонецПроцедуры &НаКлиенте Процедура ДокументОснованияПриИзменении(Элемент) &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, ДокументОснованияПриИзмененииНаСерв Текущий Объект, Параметры Записи) epe(); ТекущийПользователь = КонецПроцедуры ПользователиИнформационнойБазы. ТекущийПоль зователь(); &НаСервере // Поиск сотрудника по логину только если Процедура ЧитательПриИзмененииНаСервере() поле "Ответственный" не заполнено Если Не Если Объект. Документ Основания = ЗначениеЗаполнено(ТекущийОбъект.Ответственны Неопределено Тогда й) Тогда Возврат; // Если основание не Найденный Сотрудник = Справочники.Сотрудники.НайтиПоРеквизиту("Лог выбрано, ничего не проверяем КонепЕсли: ин", ТекущийПользователь.Имя); ЧитательИзОснования = Если НайденныйСотрудник <> Объект. Документ Основания. Получить Объект (). Чит Неопределено Тогда атель; ТекущийОбъект.Ответственный = Если Объект. Читатель <> Найденный Сотрудник; ЧитательИзОснования Тогла Иначе Сообщить ("Читатель не Сообщить("Сотрудник не соответствует выбранному основанию! Читатель найден по логину текущего пользователя. Укажите должен быть: " + ЧитательИзОснования); ответственного вручную."); Объект. Читатель = КонецЕсли; ЧитательИзОснования; // Восстанавливаем КонецЕсли; читателя из основания

КонецПроцедуры

КонецЕсли:

&НаСервере Процедура КнигиКачествоКнигиПриВозвратеПриИзменении( Процедура Сравнить Состояние Книги На Сервере (Ссылка На Кн Элемент) игу, НовоеСостояние) // Получаем текущую строку табличной части СтрокаТЧ = ДокументОснованияСсылка = Объект. Документ Основания; Элементы.Книги.ТекущиеДанные; Если НЕ // Проверяем, что выбрана книга и указан ЗначениеЗаполнено(ДокументОснованияСсылка) документ основания Тогда Если СтрокаТЧ = Неопределено ИЛИ НЕ ЗначениеЗаполнено(СтрокаТЧ.Книга) ИЛИ НЕ Возврат; КонецЕсли; ЗначениеЗаполнено(Объект. ДокументОснования) Тогда ДокументВыдачиОбъект = Возврат; // Нечего проверять ДокументОснованияСсылка.ПолучитьОбъект(); КонецЕсли; СтрокаВыдачи = ДокументВыдачиОбъект.Книги.Найти(СсылкаНаК // Получаем новое состояние из текущей нигу, "Книга"); строки НовоеСостояние = Если СтрокаВыдачи = Неопределено Тогда СтрокаТЧ.КачествоКнигиПриВозврате; Сообщить("В документе выдачи не найдена информация о книге: " + СсылкаНаКнигу); // Вызываем серверную функцию для Возврат; получения состояния при выдаче и сравнения КонецЕсли; Сравнить Состояние Книги На Сервере (Стро каТЧ.Книга, НовоеСостояние); СостояниеПриВыдаче = СтрокаВыдачи.КачествоКнигиПриВыдаче; КонецПроцедуры Если СостояниеПриВыдаче <> НовоеСостояние Код модуля объекта «ПродлениеКниги» Тогда Процедура ТекстСообщения = "Внимание! Состояние ОбработкаЗаполнения(ДанныеЗаполнения, книги "" + СсылкаНаКнигу + Стандартная Обработка) " изменилось. При выдаче: " + СостояниеПриВыдаче + Если ТипЗнч(ДанныеЗаполнения) = ", при возврате: " + НовоеСостояние Тип("ДокументСсылка.ВыдачаКниг") Тогда + "."; // Заполнение шапки Сообщить(ТекстСообщения); ДокументОснования = КонецЕсли; ДанныеЗаполнения.Ссылка; Читатель = КонецПроцедуры ДанныеЗаполнения. Читатель;

&НаКлиенте

Для Каждого ТекСтрокаКниги Из Для Каждого ТекСтрокаКниги Из Книги ДанныеЗаполнения.Книги Цикл Цикл НоваяСтрока = Движение = Движения.СрокВозвратаКниг.Добавить(); Книги.Добавить(); НоваяСтрока.Книга = Движение.Период = Дата; Движение.Книга = ТекСтрокаКниги.Книга; НоваяСтрока.Количество ТекСтрокаКниги.Книга; = ТекСтрокаКниги.Количество; Движение. Читатель = Читатель; КонецЦикла; Движение.ВремяДействия = ИначеЕсли ТипЗнч(ДанныеЗаполнения) = ЭтотОбъект.Дата; Тип("ДокументСсылка.ПродлениеКниги") Тогда Движение.ДокументВыдачи = // Заполнение шапки ДокументОснования; ДокументОснования = Движение.Количество = ДанныеЗаполнения. ДокументОснования; ТекСтрокаКниги.Количество; Комментарий = ДанныеЗаполнения.Комментарий; Движение.ПланируемаяДатаВозврата = Ответственный = ТекСтрокаКниги. Новая Дата Сдачи; ДанныеЗаполнения.Ответственный; КонецЦикла; Читатель = ДанныеЗаполнения. Читатель; КонецПроцедуры Для Каждого ТекСтрокаКниги Из ДанныеЗаполнения.Книги Цикл Процедура ПриЗаписи(Отказ) НоваяСтрока = Если Не Книги.Добавить(); ЗначениеЗаполнено(ЭтотОбъект.Ответственный) НоваяСтрока.Книга = ТекСтрокаКниги.Книга; Тогда НоваяСтрока.Количество ЭтотОбъект.Ответственный = = ТекСтрокаКниги.Количество; ПользователиИнформационнойБазы. ТекущийПоль зователь(); НоваяСтрока. НоваяДатаСдачи = КонецЕсли; ТекСтрокаКниги.НоваяДатаСдачи; КонецЦикла; КонецПроцедуры КонецЕсли; Код формы документа «ПродлениеКниги» КонецПроцедуры &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, Процедура ОбработкаПроведения(Отказ, Режим) ТекущийОбъект, ПараметрыЗаписи) // регистр СрокВозвратаКниг Для Каждого СтрокаТЧ Из Движения. Срок Возврата Книг. Записывать ТекущийОбъект.Книги Цикл = Истина:

Если СтрокаТЧ.НоваяДатаСдачи > ТекущийОбъект. Читатель); // Читатель из объекта Добавить Месяц (Текущий Объект. Дата, 1) Тогда // документа Проверка новой даты сдачи Результат = 3 апрос. Выполнить();Сообщить("Срок продления книги не может превышать один месяц Если Не Результат. Пустой() Тогда от текущей даты!"); Выборка = Отказ = Истина; Результат.Выбрать(); Возврат; // Прерываем Выборка.Следующий(); выполнение, если ошибка найдена ТекущаяДатаВозврата = КонецЕсли; Выборка.ПланируемаяДатаВозврата; КонецЦикла; // Проверяем, что новая Для Каждого СтрокаТЧ Из дата возврата не меньше текущей ТекущийОбъект.Книги Цикл СтрокаТЧ.НоваяДатаСдачи < // Получаем дату последнего Текущая Дата Возврата Тогда Сообщить("Новая продления или выдачи из регистра "СрокВозвратаКниг" дата возврата не может быть раньше текущей даты Запрос = Новый Запрос; возврата!"); Запрос.Текст = Отказ = Истина: "ВЫБРАТЬ ПЕРВЫЕ 1 Возврат; // Прерываем процедуру, если ошибка найдена СрокВозвратаКниг.ПланируемаяДатаВозврата КонецЕсли; |ИЗ Иначе РегистрСведений.СрокВозвратаКниг КАК Сообщить("Не найдена СрокВозвратаКниг запись о выдаче книги " + СтрокаТЧ.Книга); Отказ = Истина; СрокВозвратаКниг.Книга = Возврат; &Книга КонецЕсли; И СрокВозвратаКниг. Читатель КонецЦикла; = &Читатель УПОРЯДОЧИТЬ ПО КонецПроцедуры СрокВозвратаКниг.Период УБЫВ"; &НаСервере Процедура ДокументОснованияПриИзмененииНаСервере() Запрос. Установить Параметр ("Книга", СтрокаТЧ.Книга); Если Объект. Документ Основания = Неопределено Тогда

Запрос. Установить Параметр ("Читатель",

ЧитательИзОснования = Объект.Книги.Очистить(); // Очищаем табличную часть, если основание не Объект. Документ Основания. Получить Объект (). Чит выбрано атель; Возврат; КонецЕсли; Если Объект. Читатель <> ЧитательИзОснования Тогда Выдача = Сообщить("Читатель не Объект.ДокументОснования.ПолучитьОбъект(); соответствует выбранному основанию! Читатель должен быть: " + ЧитательИзОснования); Объект. Читатель = Выдача. Читатель; Объект. Читатель = ЧитательИзОснования; // Восстанавливаем Объект.Книги.Очистить(); читателя из основания Для Каждого СтрокаВыдачи Из КонецЕсли; Выдача.Книги Цикл НоваяСтрока = КонецПроцедуры Объект.Книги.Добавить(); НоваяСтрока.Книга = &НаКлиенте СтрокаВыдачи.Книга; Процедура ЧитательПриИзменении(Элемент) НоваяСтрока.Количество = СтрокаВыдачи.Количество; ЧитательПриИзмененииНаСервере(); КонецЦикла; КонецПроцедуры КонецПроцедуры Код модуля объекта «ОплатаШтрафов» &НаКлиенте &НаСервере Процедура ОбработкаПроведения(Отказ, Процедура ДокументОснованияПриИзменении(Элемент) РежимПроведения) ДокументОснованияПриИзмененииНаСерв Для Каждого СтрокаТЧ Из ЭтотОбъект.Штрафы Цикл epe(); Если СтрокаТЧ.Оплачено Тогда КонецПроцедуры &НаСервере НаборЗаписей = РегистрыСведений.ШтрафыЗаПросрочку.СоздатьН Процедура ЧитательПриИзмененииНаСервере() аборЗаписей(); Если Объект. Документ Основания = Неопределено Тогда // Отбор ТОЛЬКО по Возврат; // Если основание не регистратору (Основание) выбрано, ничего не проверяем КонецЕсли: НаборЗаписей.Отбор.Регистратор.Установ

ить(СтрокаТЧ.Основание);

Заполнить Табличную Часть Штрафов();

НаборЗаписей.Прочитать(); КонецПроцедуры Для Каждого Запись Из &НаСервере НаборЗаписей Цикл Процедура Заполнить Табличную Часть Штрафов() // Проверяем соответствие Книга и Читатель внутри цикла // Очищаем табличную часть перед Если Запись. Книга заполнением = СтрокаТЧ.Книга И Запись.Читатель = Объект.Штрафы.Очистить(); ЭтотОбъект. Читатель Тогда // Выходим, если читатель не выбран Запись. Оплачено = Истина; Если Объект. Читатель = Неопределено КонецЕсли; Тогда КонецЦикла; Возврат; КонецЕсли; НаборЗаписей.Записать(); ТекущаяДата = Дата(Год(ТекущаяДата()), Месяц(ТекущаяДата()), День(ТекущаяДата())); // КонецЕсли; Только дата, без времени КонецЦикла; Движения.РегистрБухУчет.Записывать = Запрос = Новый Запрос; Истина; Запрос.Текст = "ВЫБРАТЬ Движение = Движения.РегистрБухУчет.Добавить(); СрокВозвратаКниг.Книга КАК Движение.Период = Дата; Книга. Движение.СчетДт = ПланыСчетов.БухгалтерскийУчет.НайтиПоКоду("0 СрокВозвратаКниг.ПланируемаяДатаВозвр 1.3"); // Расчетный счет (дебет) ата КАК Планируемая Дата Возврата, Движение.СчетКт = СрокВозвратаКниг.Количество ПланыСчетов. Бухгалтерский Учет. Найти ПоКоду("0 КАК Количество, 2.2"); // Выручка (кредит) Вложенный Запрос. Сумма Штрафа Движение.Сумма = СуммаОплаты; // Цена КАК СуммаШтрафа, билета из реквизита документа Вложенный Запрос. Оплачено КАК Оплачено. КонецПроцедуры Вложенный Запрос. Основание КАК Основание И3 Код формы документа «ОплатаШтрафов» &НаКлиенте Процедура ЧитательПриИзменении(Элемент) РегистрСведений.СрокВозвратаКниг КАК СрокВозвратаКниг

ЛЕВОЕ СОЕДИНЕНИЕ	Запрос. Установить Параметр ("Читатель",
(ВЫБРАТЬ	Объект. Читатель);
	Запрос. УстановитьПараметр("ТекущаяДата
ШтрафыЗаПросрочку.СуммаШтрафа КАК	", ТекущаяДата);
СуммаШтрафа,	
	Результат = Запрос.Выполнить();
ШтрафыЗаПросрочку.Оплачено КАК	Выборка = Результат.Выбрать();
Оплачено,	
	Пока Выборка.Следующий() Цикл
ШтрафыЗаПросрочку.Основание КАК	НоваяСтрока =
Основание,	Объект.Штрафы.Добавить();
	НоваяСтрока.Книга =
ШтрафыЗаПросрочку.Книга КАК Книга,	Выборка.Книга;
	НоваяСтрока.Количество =
ШтрафыЗаПросрочку.Читатель КАК	Выборка.Количество;
Читатель	НоваяСтрока.СуммаШтрафа =
ИЗ	?(Выборка.СуммаШтрафа = NULL, 0,
	Выборка.СуммаШтрафа);
РегистрСведений.ШтрафыЗаПросрочку	НоваяСтрока.Оплачено =
КАК ШтрафыЗаПросрочку	Выборка.Оплачено;
ГДЕ	// Заполняем Основание из запроса
	НоваяСтрока.Основание =
ШтрафыЗаПросрочку.Оплачено = ЛОЖЬ)	Выборка.Основание;
КАК ВложенныйЗапрос	// Расчет суммы штрафа за книгу
ПО	ЦенаЗаДеньПросрочки =
СрокВозвратаКниг.Книга =	ОбщийМодульФункции.ПолучитьЦенуШтрафа(Пе
Вложенный Запрос. Книга	речисления. Название Штрафа. Просрочка);
И	ПросроченныеДни =
СрокВозвратаКниг. Читатель =	?(ТекущаяДата() >
Вложенный Запрос. Читатель	Выборка.ПланируемаяДатаВозврата,
ГДЕ	Цел((ТекущаяДата() -
СрокВозвратаКниг.Читатель =	Выборка.ПланируемаяДатаВозврата) / 86400), 0);
&Читатель	НоваяСтрока.СуммаШтрафа =
l N	ПросроченныеДни * ЦенаЗаДеньПросрочки *
СрокВозвратаКниг.ПланируемаяДатаВозврата <	Выборка.Количество;
&ТекущаяДата	КонецЦикла;
l N	
СрокВозвратаКниг.ФактическаяДатаВозврата =	КонецПроцедуры
ДАТАВРЕМЯ(1, 1, 1)	
И Вложенный Запрос. Оплачено =	&НаКлиенте
ложь";	Процедура ШтрафыПриИзменении(Элемент)

ПересчитатьСуммуОплаты();

КонецПроцедуры

&НаКлиенте

Процедура ПересчитатьСуммуОплаты()

СуммаОплаты = 0;

Для Каждого СтрокаШтрафа Из

Объект.Штрафы Цикл

Если СтрокаШтрафа.Оплачено

Тогда

СуммаОплаты =

CуммаOплаты + CтрокаШтрафа.CуммаШтрафа;

КонецЕсли;

КонецЦикла;

Объект.СуммаОплаты = СуммаОплаты;

КонецПроцедуры

&НаСервере

Процедура ПередЗаписьюНаСервере(Отказ,

ТекущийОбъект, ПараметрыЗаписи)

ТекущийПользователь =

ПользователиИнформационнойБазы.ТекущийПоль

зователь();

// Поиск сотрудника по логину только если

поле "Ответственный" не заполнено

Если Не

ЗначениеЗаполнено(ТекущийОбъект.Ответственны

й) Тогда

Найденный Сотрудник =

Справочники. Сотрудники. Найти По Реквизиту ("Лог

ин", ТекущийПользователь.Имя);

Если Найденный Сотрудник <>

Неопределено Тогда

ТекущийОбъект.Ответственный =

Найденный Сотрудник;

Иначе

Сообщить ("Сотрудник не

найден по логину текущего пользователя. Укажите

ответственного вручную.");

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Кол печати «ЧекОплаты»

&НаКлиенте

Процедура Обработка Команды (Параметр Команды,

ПараметрыВыполненияКоманды)

ТабДок = Новый Табличный Документ;

Печать (ТабДок, ПараметрКоманды);

ТабДок.Показать("Чек оплаты штрафа");

КонецПроцедуры

&НаСервере

Процедура Печать(ТабДок, СсылкаНаДокумент)

Макет =

Документы.ОплатаШтрафов.ПолучитьМакет("Чек

ОплатыШтрафа");

ОбластьШапка =

Макет.ПолучитьОбласть("Шапка");

ОбластьШапкаТЧ =

Макет.ПолучитьОбласть("ШапкаТЧ");

ОбластьТЧ =

Макет.ПолучитьОбласть("ТЧ");

ОбластьПодвал =

Макет.ПолучитьОбласть("Подвал");

ОбластьШапка.Параметры.НомерДок =

СсылкаНаДокумент.Номер;

ОбластьШапка.Параметры.Дата =

Формат(СсылкаНаДокумент.Дата,"ДЛФ=D");

ТабДок.Вывести(ОбластьШапка); ОбластьТЧ.Параметры.Количество = Выборка.Количество; ОбластьТЧ.Параметры.Сумма = ТабДок.Вывести(ОбластьШапкаТЧ); Выборка.Сумма; СуммаДока = СуммаДока+ Выборка.Сумма; Запрос = Новый Запрос; СуммаКоличство = Запрос.Текст = СуммаКоличство+ Выборка.Количество; "ВЫБРАТЬ ТабДок.Вывести(ОбластьТЧ); ОплатаШтрафовШтрафы.НомерСтроки КонецЦикла; КАК НомерСтроки, ОплатаШтрафовШтрафы.Книга Ответственный = КАК Книга, СсылкаНаДокумент.Ответственный.Фамилия + " " +СсылкаНаДокумент.Ответственный.Наименовани e + " " ОплатаШтрафовШтрафы.Количество КАК Количество, +СсылкаНаДокумент.Ответственный.Отчество; ОбластьПодвал.Параметры.КолНаимен = ОплатаШтрафовШтрафы.СуммаШтрафа СуммаКоличство; КАК Сумма Область Подвал. Параметры. Ответственный |ИЗ = Ответственный; ОбластьПодвал.Параметры.СуммаВсего = Документ.ОплатаШтрафов.Штрафы КАК СуммаДока; ОплатаШтрафовШтрафы ОбластьПодвал.Параметры.Итого = ГДЕ СуммаДока; ОплатаШтрафовШтрафы.Ссылка = &СсылкаНаДокумент"; Запрос. Установить Параметр ("Ссылка На До ТабДок.Вывести(ОбластьПодвал); кумент", СсылкаНаДокумент); КонецПроцедуры РезультатЗапроса = Запрос.Выполнить(); Код модуля объекта «ВыкупУтеряннойКниги» &НаСервере Выборка = РезультатЗапроса.Выбрать(); Процедура ОбработкаПроведения(Отказ, Режим) СуммаКоличство = 0; СуммаДока = 0;// === Движения по регистрам === Пока Выборка.Следующий() Цикл Движения. История Выдач Книг. Записывать = Истина; ОбластьТЧ.Параметры.Номер = Движения.ОстаткиКниг.Записывать = Выборка. НомерСтроки; Истина; // Если нужно списывать с остатков при выкупе ОбластьТЧ.Параметры.Наименование = Выборка.Книга;

```
Для Каждого ТекСтрокаКниги Из Книги
                                                                    ДвижениеБУ.Сумма =
Цикл
                                                     ТекСтрокаКниги.Сумма;
              // === Списание книги с читателя
                                                                    // Движение по регистру
(всегда, так как только утеря) ===
                                                     взаиморасчетов
                                                             Движения.Взаиморасчеты.Записывать =
       СписатьКнигуСЧитателя(ТекСтрокаКниги.
                                                     Истина;
Книга, Читатель, ТекСтрокаКниги.Количество,
                                                                    ДвижениеВР =
Отказ);
                                                     Движения.Взаиморасчеты.Добавить();
                                                                    ДвижениеВР.ВидДвижения =
              // === Движения по регистру
                                                     ВидДвиженияНакопления.Приход;
                                                                    ДвижениеВР.Период = Дата;
бухгалтерии ===
                                                                    ДвижениеВР. Читатель = Читатель;
       Движения.РегистрБухУчет.Записывать =
                                                                    ДвижениеВР.Сумма =
Истина;
                                                     ТекСтрокаКниги.Сумма;
              // Проводка по получению денег от
читателя
              ДвижениеБУ =
                                                             КонецЦикла;
Движения.РегистрБухУчет.Добавить();
              ДвижениеБУ.Период = Дата;
              ДвижениеБУ.СчетДт =
ПланыСчетов.БухгалтерскийУчет.РасчетныйСчет;
                                                             Движения.СредняяСобестоимостьТоваров.
              ДвижениеБУ.СчетКт =
                                                     Записывать = Истина;
ПланыСчетов.БухгалтерскийУчет.Выручка;
                                                             Движения.ПродажаКниг.Записывать =
                                                     Истина:
              ДвижениеБУ.Сумма =
ТекСтрокаКниги.Сумма;
                                                             Запрос = Новый Запрос;
                                                             Запрос.Текст =
                                                             "ВЫБРАТЬ
              // Проводка по списанию
себестоимости книги
                                                             СредняяСобестоимостьТоваровОстатки.Кн
              ДвижениеБУ =
                                                     ига КАК Книга,
Движения.РегистрБухУчет.Добавить();
              ДвижениеБУ.Период = Дата;
                                                             СредняяСобестоимостьТоваровОстатки.Су
              ДвижениеБУ.СчетДт =
                                                     ммаОстаток КАК Сумма,
ПланыСчетов.БухгалтерскийУчет.Собестоимость;
              ДвижениеБУ.СчетКт =
                                                             Средняя Собестоимость Товаров Остатки. Ко
ПланыСчетов.БухгалтерскийУчет.Товары;
                                                     личествоОстаток КАК Количество
                                                             И3
```

Иначе РегистрНакопления.СредняяСобестоимост ьТоваров.Остатки( СобестоимостьЕдиницы=0; КонецЕсли; &МоментВремени, Книга В Движение = Движения.СредняяСобестоимостьТоваров.Добавит (ВЫБРАТЬ ь(); Движение.ВидДвижения = ВыкупУтеряннойКнигиКниги.Книга КАК ВидДвиженияНакопления.Расход; Книга Движение.Период = Дата; И3 СтрокаТЧ = Документ.ВыкупУтеряннойКниги.Книги Книги. Найти (Выборка Детальные Записи. Книга, "Кн КАК ВыкупУтеряннойКнигиКниги ига"); ГДЕ Движение.Книга = Выборка Детальные Записи. Книга; ВыкупУтеряннойКнигиКниги.Ссылка = Движение.Количество = &Ссылка)) КАК СтрокаТЧ.Количество; СредняяСобестоимостьТоваровОстатки"; СебеСтоимостьСписание = СобестоимостьЕдиницы\*СтрокаТЧ.Количество; Запрос. Установить Параметр ("Момент Врем Движение.Сумма = ени", МоментВремени()); СебеСтоимостьСписание; Запрос. Установить Параметр ("Ссылка", //движения по регистру ПродажаКниг Ссылка); Движение = РезультатЗапроса = Запрос.Выполнить(); Движения.ПродажаКниг.Добавить(); Движение.ВидДвижения = ВыборкаДетальныеЗаписи = ВидДвиженияНакопления.Расход; РезультатЗапроса.Выбрать(); Движение.Период = Дата; Движение.Книга = CуммаCобестоимости = 0; ВыборкаДетальныеЗаписи.Книга; Движение. Читатель = Читатель; Пока Движение.Количество = ВыборкаДетальныеЗаписи.Следующий() Цикл СтрокаТЧ.Количество; Движение.Сумма = СтрокаТЧ.Сумма; Если ВыборкаДетальныеЗаписи.Количество <>0 тогда Движение.Собестоимость =СебеСтоимостьСписание; СобестоимостьЕдиницы =

Выборка Детальные Записи. Сумма/Выборка Детальн

ыеЗаписи.Количество;

И

```
СуммаСобестоимости=СуммаСобестоимос
                                                      СрокВозвратаКниг.ФактическаяДатаВозврата =
ти+СебеСтоимостьСписание;
                                                      ДАТАВРЕМЯ(1, 1, 1)
                                                                     УПОРЯДОЧИТЬ ПО
                                                                     СрокВозвратаКниг.Период
       КонецЦикла;
                                                      УБЫВ";
                                                             Запрос. Установить Параметр ("Книга",
       // === Запись в регистр
                                                      СтрокаТЧ.Книга);
"СрокВозвратаКниг" ===
       Движения. Срок Возврата Книг. Записывать
                                                             Запрос. Установить Параметр ("Читатель",
                                                      Читатель);
= Истина;
       Для Каждого СтрокаТЧ Из Книги Цикл
                                                                     Результат = Запрос.Выполнить();
              // Проверяем, была ли книга ранее
                                                                     Если Не Результат.Пустой() Тогда
выдана этому читателю
              Запрос = Новый Запрос;
                                                      // Если книга была выдана
               Запрос.Текст =
                                                                            Выборка =
               "ВЫБРАТЬ ПЕРВЫЕ 1
                                                      Результат.Выбрать();
                                                                             Выборка.Следующий();
СрокВозвратаКниг.ДокументВыдачи, // Добавили
нужные поля
                                                                            // Создаем движение на
                                                      основании найденной записи
СрокВозвратаКниг.ПланируемаяДатаВозврата,
                                                                             Движение =
                                                      Движения.СрокВозвратаКниг.Добавить();
СрокВозвратаКниг.ВремяДействия,
       СрокВозвратаКниг.Количество //
                                                             Заполнить Движение Регистра (Движение,
Добавляем количество, чтобы проверить, есть ли
                                                      СтрокаТЧ, Читатель, Выборка);
еще невозвращенные книги
              |ИЗ
                                                                     КонецЕсли;
                                                             КонецЦикла;
РегистрСведений.СрокВозвратаКниг КАК
СрокВозвратаКниг
                                                      КонецПроцедуры
              | СрокВозвратаКниг.Книга =
                                                      // === Процедура для списания книги с читателя
&Книга
              И СрокВозвратаКниг. Читатель
= &Читатель
                                                      &НаСервере
                                                      Процедура СписатьКнигуСЧитателя(Книга,
                                                      Читатель, Количество, Отказ)
```

Отбор = Новый Структура; // Поиск записи о выдаче данной книги Отбор.Вставить("Книги", Книга); этому читателю НайденныеЗначения = ЗаписьВыдачи = РегистрыНакопления.ДоступныеКниги.Остатки(Да НайтиЗаписьВыдачи(Книга, Читатель); та,Отбор); Если Найденные Значения. Количество() > 0 Если ЗаписьВыдачи = Неопределено Тогда Тогда Сообщить("Не найдена запись о Результат = выдаче книги " + Книга + " читателю " + Читатель); НайденныеЗначения[0].Количество; Отказ = Истина; КонецЕсли; Возврат; // Прекращаем Возврат Результат; выполнение процедуры, если запись не найдена КонецЕсли; КонецФункции // Проверка количества // === Функция для поиска записи о выдаче == Если ЗаписьВыдачи.Количество < Функция НайтиЗаписьВыдачи(Книга, Читатель) Количество Тогда Сообщить ("Попытка списать Отбор = Новый Структура("Книга, больше книг, чем было выдано. Доступно для списания: " + ЗаписьВыдачи.Количество); Читатель", Книга, Читатель); Отказ = Истина; // Ищем запись с нужной книгой и Возврат; читателем КонецЕсли; НайденныеЗначения = РегистрыНакопления.ИсторияВыдачКниг.Остатки( // Создание движения по регистру Дата, Отбор); "ИсторияВыдачКниг" Движение = Если Найденные Значения. Количество() > 0 Движения. История Выдач Книг. Добавить(); Тогда Движение.ВидДвижения = Возврат НайденныеЗначения[0]; ВидДвиженияНакопления.Расход; Иначе Движение.Период = Дата; Возврат Неопределено; Движение.Книга = Книга; КонецЕсли; Движение. Читатель = Читатель; Движение.Количество = Количество; КонецФункции КонецПроцедуры Процедура ПередЗаписью(Отказ, РежимЗаписи, // === Функция для получения остатка книг === РежимПроведения) Функция ПолучитьОстаток(Книга) СуммаДокумента=Книги.Итог("Сумма");

КонецПроцедуры

Результат = 0;

	КонецПроцедуры
&НаСервере	
Процедура	&НаКлиенте
Заполнить Движение Регистра (Движение, Строка ТЧ,	Процедура КнигиСуммаПриИзменении(Элемент)
Читатель, Выборка)	
	СтрокаТабличнойЧасти =
Движение.Период = Дата;	Элементы.Книги.ТекущиеДанные;
Движение.Книга = СтрокаТЧ.Книга;	Если СтрокаТабличнойЧасти.Количество =
Движение. Читатель = Читатель;	0 тогда
Движение.ВремяДействия =	СтрокаТабличнойЧасти.Цена = $0$ ;
Выборка.ВремяДействия;	Иначе
Движение.ДокументВыдачи =	СтрокаТабличнойЧасти.Цена =
Выборка.ДокументВыдачи;	СтрокаТабличнойЧасти.Сумма/
Движение.Количество =	Строка Табличной Части. Количество;
СтрокаТЧ.Количество;	КонецЕсли;
Если	
ЗначениеЗаполнено(Выборка.ПланируемаяДатаВоз	
врата) Тогда	КонецПроцедуры
Движение.ПланируемаяДатаВозврата =	&НаКлиенте
Выборка.ПланируемаяДатаВозврата;	Процедура
КонецЕсли;	КнигиКоличествоПриИзменении(Элемент)
Движение.ФактическаяДатаВозврата =	
Дата;	СтрокаТабличнойЧасти =
	Элементы.Книги.ТекущиеДанные;
// Устанавливаем признак выкупа	СтрокаТабличнойЧасти.Сумма =
Движение.Выкуплена = Истина;	СтрокаТабличнойЧасти.Количество*СтрокаТаблич
	нойЧасти.Цена;
КонецПроцедуры	
	КонецПроцедуры
Код формы документа	
«ВыкупУтеряннойКниги»	&НаКлиенте
	Процедура КнигиКнигаПриИзменении(Элемент)
&НаКлиенте	
Процедура КнигиЦенаПриИзменении(Элемент)	СтрокаТЧ =
	Элементы.Книги.ТекущиеДанные;
СтрокаТабличнойЧасти =	СтрокаТЧ.Цена=ОбщийМодульФункции.Ц
Элементы.Книги.ТекущиеДанные;	еныИзРегистраСведений(Объект.Дата,СтрокаТЧ.К
СтрокаТабличнойЧасти.Сумма =	нига);

КонецПроцедуры

СтрокаТабличнойЧасти.Количество\*СтрокаТаблич

ной Части. Цена;

&НаСервере Процедура Печать(ТабДок, СсылкаНаДокумент) Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи) Макет = ТекущийПользователь = Документы.ВыкупУтеряннойКниги.ПолучитьМаке ПользователиИнформационнойБазы. ТекущийПоль т("ТоварныйЧек"); зователь(); ОбластьШапка = Макет.ПолучитьОбласть("Шапка"); Областы ПапкаТЧ = // Поиск сотрудника по логину только если поле "Ответственный" не заполнено Макет.ПолучитьОбласть("ШапкаТЧ"); Если Не ОбластьТЧ = ЗначениеЗаполнено(ТекущийОбъект.Ответственны Макет.ПолучитьОбласть("ТЧ"); й) Тогда ОбластьПодвал = Найденный Сотрудник = Макет.ПолучитьОбласть("Подвал"); Справочники. Сотрудники. Найти По Реквизиту ("Лог ин", ТекущийПользователь.Имя); ОбластьШапка.Параметры.НомерДок = СсылкаНаДокумент.Номер; ОбластьШапка.Параметры.Дата = Если Найденный Сотрудник <> Формат(СсылкаНаДокумент.Дата,"ДЛФ=D"); Неопределено Тогда ТабДок.Вывести(ОбластьШапка); ТекущийОбъект.Ответственный = Найденный Сотрудник; Иначе ТабДок.Вывести(ОбластьШапкаТЧ); Сообщить("Сотрудник не найден по логину текущего пользователя. Укажите ответственного вручную."); Запрос = Новый Запрос; КонецЕсли; Запрос.Текст = "ВЫБРАТЬ КонепЕсли: КонецПроцедуры ВыкупУтеряннойКнигиКниги.НомерСтрок и КАК НомерСтроки, Код печати «Товарный Чек» ВыкупУтеряннойКнигиКниги.Книга КАК &НаКлиенте Процедура Обработка Команды (Параметр Команды, Книга, ПараметрыВыполненияКоманды) ВыкупУтеряннойКнигиКниги.Количество ТабДок = Новый Табличный Документ; КАК Количество, Печать (ТабДок, ПараметрКоманды); ТабДок.Показать("Товарный чек"); ВыкупУтеряннойКнигиКниги. Цена КАК Цена, КонецПроцедуры ВыкупУтеряннойКнигиКниги.Сумма КАК &НаСервере Сумма

|ИЗ +СсылкаНаДокумент.Ответственный.Наименовани e + " " Документ.ВыкупУтеряннойКниги.Книги +СсылкаНаДокумент.Ответственный.Отчество; КАК ВыкупУтеряннойКнигиКниги Область Подвал. Параметры. Ответственный ГДЕ = Ответственный; ОбластьПодвал.Параметры.СуммаВсего = ВыкупУтеряннойКнигиКниги.Ссылка = СуммаДока; &СсылкаНаДокумент"; ОбластьПодвал.Параметры.Итого = Запрос. Установить Параметр ("Ссылка На До СуммаДока; кумент", СсылкаНаДокумент); РезультатЗапроса = Запрос.Выполнить(); ТабДок.Вывести(ОбластьПодвал); КонецПроцедуры Выборка = РезультатЗапроса.Выбрать(); СуммаКоличство = 0; Код модуля объекта «Читательский Билет» СуммаДока = 0;&НаСервере Пока Выборка. Следующий() Цикл Процедура ОбработкаПроведения(Отказ, РежимПроведения) ОбластьТЧ.Параметры.Номер = Выборка. НомерСтроки; //Движения.РегистрБухУчет.Записывать = Истина; ОбластьТЧ.Параметры.Наименование = //Движение = Выборка.Книга; Движения.РегистрБухУчет.Добавить(); ОбластьТЧ.Параметры.Количество //Движение.Период = Дата; = Выборка.Количество; //Движение.СчетДт = ОбластьТЧ.Параметры.Цена = ПланыСчетов.БухгалтерскийУчет.НайтиПоКоду("0 Выборка. Цена; 1.3"); // Расчетный счет (дебет) ОбластьТЧ.Параметры.Сумма = //Движение.СчетКт = Выборка.Сумма; ПланыСчетов.БухгалтерскийУчет.НайтиПоКоду("0 СуммаДока = СуммаДока+ 2.2"); // Выручка (кредит) Выборка.Сумма; //Движение.Сумма = Цена; // Цена билета СуммаКоличство = из реквизита документа СуммаКоличство+ Выборка.Количество; ТабДок.Вывести(ОбластьТЧ); КонецЦикла; ДатаОкончания = ДатаОкончанияСрокаБилета; Область Подвал. Параметры. Кол Наимен = // Запись в регистр сведений СуммаКоличство; Ответственный = РегистрыСведений.СрокиДействияЧитательскихБи СсылкаНаДокумент.Ответственный.Фамилия + " " летов.СоздатьМенеджерЗаписи();

```
Запись. Читательский Билет = Ссылка;
                                                             Рассчитать ИУстановить Цену();
  Запись. Дата Окончания = Дата Окончания;
  Запись.ТипДействия =
                                                      КонецПроцедуры
Перечисления.ТипДействия.Выдача;
  Запись.Период = ТекущаяДата();
                                                      &НаКлиенте
  Запись.Записать();
                                                      Процедура Рассчитать ИУ становить Цену()
                                                             Абонемент = Объект. Абонемент;
КонецПроцедуры
                                                             Возрастной Рейтинг =
&НаСервере
                                                      Объект.ВозрастнойРейтинг;
Процедура ОбработкаОтменыПроведения(Отказ)
                                                             Цена =
                                                      Общий Модуль Функции. Получить Цену Билета (Або
       //// Сторнирование движений
                                                      немент, ВозрастнойРейтинг);
       //Для Каждого Движение Из
                                                             Объект.Цена = Цена;
Движения.РегистрБухУчет Цикл
       //
              НовоеДвижение =
                                                      КонецПроцедуры
Движения.РегистрБухУчет.Добавить();
       //
              НовоеДвижение.Период =
                                                      &НаКлиенте
Движение.Период;
                                                      Процедура Рассчитать Дату Окончания ()
              НовоеДвижение.СчетДт =
Движение.СчетКт;
                                                             РассчитатьДатуОкончанияНаСервере();
       //
              НовоеДвижение.СчетКт =
Движение.СчетДт;
                                                      КонецПроцедуры
       //
              НовоеДвижение.Сумма =
Движение.Сумма;
                                                      &НаСервере
       //КонецЦикла;
                                                      Процедура Рассчитать Дату Окончания На Сервере()
КонецПроцедуры
                                                             Выбранный Абонемент =
                                                      Объект. Абонемент;
Код формы документа «Читательский Билет»
&НаКлиенте
                                                             Если Выбранный Абонемент =
Процедура АбонементПриИзменении(Элемент)
                                                      Перечисления.СрокДействия Абониментов. Месяц
                                                      1 Тогда
       Рассчитать ИУстановить Цену();
                                                                     КоличествоМесяцев = 1;
       РассчитатьДатуОкончания();
                                                             ИначеЕсли Выбранный Абонемент =
                                                      Перечисления.СрокДействия Абониментов. Месяц
КонецПроцедуры
                                                      3 Тогда
                                                                     КоличествоMесяцев = 3;
&НаКлиенте
                                                             ИначеЕсли Выбранный Абонемент =
Процедура
                                                      Перечисления.СрокДействия Абониментов. Месяц
Возрастной Рейтинг При Изменении (Элемент)
                                                      6 Тогда
```

КоличествоМесяцев = 6;

ИначеЕсли Выбранный Абонемент = //Запись.ДатаОкончания = ДатаОкончания; Перечисления.СрокДействия Абониментов. Месяц //Запись.ТипДействия = 12 Тогда Перечисления.ТипДействия.Выдача; //Запись.Период = ТекущаяДата(); КоличествоМесяцев = 12; Иначе //Запись.Записать(); Количество Месяцев = 0; КонецЕсли; КонецПроцедуры Если КоличествоМесяцев > 0 Тогда ДатаОкончания = &НаКлиенте Добавить Месяц (Объект. Дата, Процедура ДатаПриИзменении(Элемент) КоличествоМесяцев); // Дата окончания = дата регистрации + количество месяцев Рассчитать Дату Окончания(); Объект. Дата Окончания Срока Билета = КонецПроцедуры ДатаОкончания; Иначе &НаСервере Процедура ПередЗаписьюНаСервере(Отказ, Объект. Дата Окончания Срока Билета = Null; Текущий Объект, Параметры Записи) КонепЕсли: КонецПроцедуры ПроверитьДопустимостьВыдачиБилетаНа Сервере(Отказ); &НаКлиенте ТекущийПользователь = ПользователиИнформационнойБазы. ТекущийПоль Процедура ДатаРегистрацииБилетаПриИзменении(Элемент) зователь(); Рассчитать Дату Окончания(); // Поиск сотрудника по логину только если поле "Ответственный" не заполнено КонецПроцедуры Если Не ЗначениеЗаполнено(ТекущийОбъект.Ответственны &НаСервере й) Тогда Процедура ПриЗаписиНаСервере(Отказ, Найденный Сотрудник = ТекущийОбъект, ПараметрыЗаписи) Справочники. Сотрудники. Найти ПоРеквизиту ("Лог ин", ТекущийПользователь.Имя); //ДатаОкончания = Объект. Дата Окончания Срока Билета; Если НайденныйСотрудник <> //// Запись в регистр сведений Неопределено Тогда //Запись = РегистрыСведений.СрокиДействияЧитательскихБи ТекущийОбъект.Ответственный = летов.СоздатьМенеджерЗаписи(); Найденный Сотрудник; //Запись. Читательский Билет = Иначе Объект.Ссылка:

Сообщить("Сотрудник не ВозрастЧитателя = найден по логину текущего пользователя. Укажите СколькоЛетТебе(ТекущаяДата(), Читатель. Дата Рождения); ответственного вручную."); КонецЕсли; КонецЕсли; Если Возрастной Рейтинг = Перечисления. Возрастной Рейтинг. Без Ограничений КонецПроцедуры Тогда Возврат Истина; &НаСервереБезКонтекста ИначеЕсли ВозрастнойРейтинг = Функция СколькоЛетТебе(ТекДата, ДатаРождения) Перечисления.ВозрастнойРейтинг.ДляДетейОт6лет И Возраст Читателя >= 6 Тогда Если ДатаРождения = Неопределено Тогда Возврат Истина; Возврат 0; ИначеЕсли ВозрастнойРейтинг = КонецЕсли; Перечисления.ВозрастнойРейтинг.ДляДетейОт12Л ет И ВозрастЧитателя >= 12 Тогда Возраст =  $\Gamma$ од(ТекДата) -Возврат Истина; Год(ДатаРождения); ИначеЕсли ВозрастнойРейтинг = Перечисления. Возрастной Рейтинг. Для Подростков Если Месяц(ТекДата) < От16Лет И ВозрастЧитателя >= 16 Тогда Месяц(ДатаРождения) Или (Месяц(ТекДата) = Месяц(ДатаРождения) И День(ТекДата) < Возврат Истина; День(ДатаРождения)) Тогда ИначеЕсли ВозрастнойРейтинг = Bospact = Bospact - 1;Перечисления. Возрастной Рейтинг. Для В зрослых И КонецЕсли; ВозрастЧитателя >= 18 Тогда Возврат Истина; Возврат Возраст; Иначе Возврат Ложь; КонецФункции КонецЕсли: КонецФункции &НаСервере Функция &НаСервере Проверить Допустимость Выдачи Билета (Читатель, Процедура Абонемент, Возрастной Рейтинг) Экспорт Проверить Допустимость Выдачи Билета На Сервере ( Отказ) Если Читатель. Дата Рождения = Неопределено Тогда Если Не Сообщить("Не указана дата Проверить Допустимость Выдачи Билета (Объект. Чит атель, Объект. Абонемент, рождения читателя."); Объект.ВозрастнойРейтинг) Тогда Возврат Ложь;

Сообщить("Выдача билета

невозможна. Возраст читателя не соответствует

возрастному рейтингу абонемента.");

КонецЕсли;

Отказ = Истина: //Движение.СчетДт = КонецЕсли; ПланыСчетов.БухгалтерскийУчет.НайтиПоКоду("0 1.3"); // Расчетный счет (дебет) КонецПроцедуры //Движение.СчетКт = ПланыСчетов.БухгалтерскийУчет.НайтиПоКоду("0 &НаКлиенте 2.2"); // Выручка (кредит) Процедура ПередЗаписью(Отказ, //Движение.Сумма = Цена; // Цена ПараметрыЗаписи) продления из реквизита документа ПроверитьДопустимостьВыдачиБилетаНа КонецПроцедуры Сервере(Отказ); &НаСервере КонецПроцедуры Процедура ОбработкаОтменыПроведения(Отказ) Код модуля объекта документа // Сторнирование движений «ПродлениеЧитательскогоБилета» Для Каждого Движение Из Движения.РегистрБухУчет Цикл Процедура ОбработкаЗаполнения(ДанныеЗаполнения, НовоеДвижение = Стандартная Обработка) Движения.РегистрБухУчет.Добавить(); НовоеДвижение.Период = Если ТипЗнч(ДанныеЗаполнения) = Движение.Период; Тип("ДокументСсылка. Читательский Билет") Тогда НовоеДвижение.СчетДт = // Заполнение шапки Движение.СчетКт; // Меняем местами счета НовоеДвижение.СчетКт = Абонемент = неопределено; Возрастной Рейтинг = Движение.СчетДт; ДанныеЗаполнения.ВозрастнойРейтинг; НовоеДвижение.Сумма = Основание = Движение.Сумма; ДанныеЗаполнения.Ссылка; КонецЦикла; Цена = ДанныеЗаполнения.Цена; Читатель = КонецПроцедуры ДанныеЗаполнения. Читатель; КонецЕсли; Код формы документа «ПродлениеЧитательскогоБилета» КонецПроцедуры &НаСервере Процедура ПриЗаписиНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи) &НаСервере Процедура ОбработкаПроведения(Отказ, РежимПроведения) ДатаОкончания = Объект. Дата Окончания Срока Билета; //Движение = // Запись в регистр сведений Движения.РегистрБухУчет.Добавить();

//Движение.Период = Дата;

Возрастной Рейтинг = Запись = РегистрыСведений.СрокиДействияЧитательскихБи Объект.ВозрастнойРейтинг; летов.СоздатьМенеджерЗаписи(); Цена = Запись. Читательский Билет = Общий Модуль Функции. Получить Цену Билета (Або Объект.Основание; немент, ВозрастнойРейтинг); Объект.Цена = Цена; Запись. Дата Окончания = Дата Окончания; Запись.ТипДействия = Перечисления. Тип Действия. Продление; КонецПроцедуры Запись.Период = ТекущаяДата(); Запись.Записать(); &НаКлиенте Процедура Рассчитать Дату Окончания () КонецПроцедуры РассчитатьДатуОкончанияНаСервере(); &НаКлиенте Процедура АбонементПриИзменении(Элемент) КонецПроцедуры Рассчитать ИУстановить Цену(); &НаСервере РассчитатьДатуОкончания(); Процедура Рассчитать Дату Окончания На Сервере() КонецПроцедуры Выбранный Абонемент = Объект. Абонемент; &НаКлиенте Если Выбранный Абонемент = Перечисления.СрокДействия Абониментов. Месяц\_ Процедура Возрастной Рейтинг При Изменении (Элемент) 1 Тогда КоличествоМесяцев = 1; Рассчитать ИУстановить Цену(); ИначеЕсли Выбранный Абонемент = Перечисления.СрокДействия Абониментов. Месяц КонецПроцедуры 3 Тогда КоличествоМесяцев = 3; &НаКлиенте ИначеЕсли Выбранный Абонемент = Перечисления.СрокДействия Абониментов. Месяц Процедура ДатаРегистрацииБилетаПриИзменении(Элемент) 6 Тогда КоличествоМесяцев = 6; РассчитатьДатуОкончания(); ИначеЕсли Выбранный Абонемент = Перечисления.СрокДействия Абониментов. Месяц КонецПроцедуры 12 Тогда КоличествоМесяцев = 12; &НаКлиенте Иначе Процедура Рассчитать ИУстановить Цену() КоличествоMесяцев = 0; КонецЕсли:

Абонемент = Объект. Абонемент:

Если КоличествоМесяцев > 0 Тогда ТекущийПользователь = ДатаОкончания = ПользователиИнформационнойБазы.ТекущийПоль Добавить Месяц (Объект. Дата, зователь(); КоличествоМесяцев); // Дата окончания = дата регистрации + количество месяцев // Поиск сотрудника по логину только если поле "Ответственный" не заполнено Объект. Дата Окончания Срока Билета = Если Не ЗначениеЗаполнено(ТекущийОбъект.Ответственны ДатаОкончания; Иначе й) Тогда НайденныйСотрудник = Объект.ДатаОкончанияСрокаБилета = Null; Справочники.Сотрудники.НайтиПоРеквизиту("Лог КонецЕсли; ин", ТекущийПользователь.Имя); КонецПроцедуры Если Найденный Сотрудник <> Неопределено Тогда &НаКлиенте Процедура ДатаПриИзменении(Элемент) ТекущийОбъект.Ответственный = Найденный Сотрудник; Рассчитать ДатуОкончания(); Иначе Сообщить ("Сотрудник не КонецПроцедуры найден по логину текущего пользователя. Укажите ответственного вручную."); &НаСервере КонецЕсли; Процедура ПередЗаписьюНаСервере(Отказ, КонецЕсли; ТекущийОбъект, ПараметрыЗаписи) КонецПроцедуры ПроверитьДопустимостьВыдачиБилетаНа Сервере(Отказ); &НаСервереБезКонтекста Функция СколькоЛетТебе(ТекДата, ДатаРождения) Если ТекущийОбъект.Основание <> Неопределено Тогда Если ДатаРождения = Неопределено Тогда ЧитательИзОснования = Возврат 0; Текущий Объект. Основание. Получить Объект (). Чит КонецЕсли; атель: Если ТекущийОбъект. Читатель <> Возраст =  $\Gamma$ од(TекДата) -ЧитательИзОснования Тогда Год(ДатаРождения); Сообщить("Нельзя менять Если Месяц(ТекДата) < читателя при продлении билета!"); Месяц(ДатаРождения) Или (Месяц(ТекДата) = Отказ = Истина: // Месяц(ДатаРождения) И День(ТекДата) < Запрещаем запись документа День(ДатаРождения)) Тогда КонецЕсли: Bозраст = Возраст - 1;

КонецЕсли;

КонецЕсли;

Возврат Ложь;

Возврат Возраст;

КонецЕсли;

КонецФункции

&НаСервере

Функция

Проверить Допустимость Выдачи Билета (Читатель,

Абонемент, Возрастной Рейтинг) Экспорт

Если Читатель. Дата Рождения =

Неопределено Тогда

Сообщить ("Не указана дата

рождения читателя.");

Возврат Ложь;

КонецЕсли;

ВозрастЧитателя =

СколькоЛетТебе(ТекущаяДата(),

Читатель. ДатаРождения);

Если Возрастной Рейтинг =

Перечисления.ВозрастнойРейтинг.БезОграничений

Тогда

Возврат Истина;

ИначеЕсли ВозрастнойРейтинг =

Перечисления.ВозрастнойРейтинг.ДляДетейОт6лет

И Возраст Читателя >= 6 Тогда

Возврат Истина;

ИначеЕсли ВозрастнойРейтинг =

Перечисления.ВозрастнойРейтинг.ДляДетейОт12Л

ет И Возраст Читателя >= 12 Тогда

Возврат Истина;

ИначеЕсли ВозрастнойРейтинг =

Перечисления. Возрастной Рейтинг. Для Подростков

От 16 Лет И Возраст Читателя >= 16 Тогда

Возврат Истина;

ИначеЕсли ВозрастнойРейтинг =

Перечисления. Возрастной Рейтинг. Для В зрослых И

Возраст Читателя >= 18 Тогда

Возврат Истина;

Иначе

КонецФункции

&НаСервере

Процедура

Проверить Допустимость Выдачи Билета На Сервере (

Отказ)

Если Не

Проверить Допустимость Выдачи Билета (Объект. Чит

атель, Объект. Абонемент,

Объект.ВозрастнойРейтинг) Тогда

Сообщить("Выдача билета

невозможна. Возраст читателя не соответствует

возрастному рейтингу абонемента.");

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ПередЗаписью(Отказ,

ПараметрыЗаписи)

ПроверитьДопустимостьВыдачиБилетаНа

Сервере(Отказ);

КонецПроцедуры

&НаСервере

Процедура ОснованиеПриИзмененииНаСервере()

Если Объект. Основание = Неопределено

Тогда

// Очищаем поля, если основание

не выбрано

Объект. Абонемент =

Неопределено;

Объект.ВозрастнойРейтинг =

Неопределено;

Объект.Цена = 0; Процедура ПутьКФайлуНачалоВыбора(Элемент, ДанныеВыбора, ВыборДобавлением, Объект. Дата Окончания Срока Билета = Стандартная Обработка) Неопределено; Объект. Читатель = Неопределено; Проводник = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла. КонецЕсли; Открытие); Проводник. Заголовок = "Выберите файл"; Если Объект. Формат Файла = "XLSX" Читательский Билет = Объект.Основание.ПолучитьОбъект(); Тогда Фильтр = "СписокАвторов (\*.xlsx)|\*.xlsx"; Объект. Читатель = ИначеЕсли Объект. Формат Файла = "TXT" Читательский Билет. Читатель; Объект. Абонемент = тогда Читательский Билет. Абонемент: Фильтр = "СписокАвторов Объект.ВозрастнойРейтинг = (\*.txt)|\*.txt"; Читательский Билет. Возрастной Рейтинг; Объект.Цена = Читательский Билет.Цена; Иначе Объект. Дата Окончания Срока Билета = Возврат; Читательский Билет. Дата Окончания Срока Билета; КонецЕсли; Проводник. Фильтр = Фильтр; // Проверка читателя Если Читательский Билет. Читатель <> Оповещение = Новый Объект. Читатель Тогда ОписаниеОповещения("ПослеВыбораФайла", Сообщить("Читатель не ЭтотОбъект); соответствует документу-основанию!"); Проводник.Показать(Оповещение); Объект. Читатель = Читательский Билет. Читатель: КонепЕсли: КонецПроцедуры КонецПроцедуры &НаКлиенте Процедура &НаКлиенте ПослеВыбораФайла(ВыбранныеФайлы,Дополнител Процедура ОснованиеПриИзменении(Элемент) ьныеПараметры) Экспорт ОснованиеПриИзмененииНаСервере(); Если Выбранные Файлы = Неопределено Тогда КонецПроцедуры Возврат; КонецЕсли; Объект.ПутьКФайлу = Код формы обработки «Загрузить Авторов» &НаКлиенте ВыбранныеФайлы[0];

КонецПроцедуры НоваяСтрока.Фамилия=МассивСлов[1]; &НаКлиенте Процедура Прочитать Файл (Команда) НоваяСтрока.ГодРождения=МассивСлов[2 ]; Объект. Данные Файла. Очистить(); НоваяСтрока.Национальность=МассивСло Если Объект. Формат Файла = "XLSX" в[3]; Тогда ПрочитатьФайл XLSX(); НоваяСтрока.СсылкаНаСайтАвтора=Масси ИначеЕсли Объект. Формат Файла = "ТХТ" вСлов[4]; Тогда ПрочитатьФайл ТХТ(); ТекСтрока=Текст.ПрочитатьСтроку(); КонецЕсли; КонецЦикла; Иначе КонецПроцедуры Текст = Новый Текстовый Документ; &НаКлиенте Текст.Прочитать(Объект.ПутьКФайлу); Процедура Прочитать Файл ТХТ() Для НомерСтроки=1 По Последовательное Чтение = Истина; Текст.КоличествоСтрок() Цикл Если Последовательное Чтение ТОгда ТекСтрока = Текст = Новый ЧтениеТекста; Текст.ПолучитьСтроку(НомерСтроки); Текст.Открыть(Объект.ПутьКФайлу, МассивСлов=СтрРазделить(ТекСтрока, КодировкаТекста. UTF8); ";"); Если ТекСтрока=Текст.ПрочитатьСтроку(); МассивСлов.Количество() < 5 Тогда Пока ТекСтрока <> Неопределено Продолжить; Цикл КонецЕсли; МассивСлов = СтрРазделить(ТекСтрока,";"); НоваяСтрока=Объект. ДанныеФайла. Добав Если ить(); МассивСлов.Количество()<5 Тогда НоваяСтрока.Имя= Продолжить; МассивСлов[0];КонецЕсли;

1;

НоваяСТрока=Объект.ДанныеФайла.Добав

НоваяСтрока.Имя=

ить();

МассивСлов[0];

НоваяСтрока.Фамилия=МассивСлов[1];

НоваяСтрока.ГодРождения=МассивСлов[2

КоличествоСтрок = НоваяСтрока.Национальность=МассивСло ТабДок.ВысотаТаблицы; в[3]; Для НомерСТроки = 2 По КоличествоСтрок Цикл НоваяСтрока.СсылкаНаСайтАвтора=Масси СтрокаДанных = Объект. Данные Файла. Добавить(); вСлов[4]; КонецЦикла; СтрокаДанных.Имя = КонецЕсли; ТабДок.ПолучитьОбласть("R"+Формат(НомерСтро ки,"ЧГ=0")+"С"+1).ТекущаяОбласть.Текст; КонецПроцедуры СтрокаДанных.Фамилия = ТабДок.ПолучитьОбласть("R"+Формат(НомерСтро ки,"ЧГ=0")+"С"+2).ТекущаяОбласть.Текст; &НаСервере СтрокаДанных.ГодРождения = Процедура ПрочитатьФайл\_XLSX\_HaСервере(АдресДанных) ТабДок.ПолучитьОбласть("R"+Формат(НомерСтро ки, "ЧГ=0")+"С"+3). Текущая Область. Текст; ТабДок = Новый Табличный Документ; СтрокаДанных.Национальность = ТабДок.ПолучитьОбласть("R"+Формат(НомерСтро ки,"ЧГ=0")+"С"+4).ТекущаяОбласть.Текст; Данные = ПолучитьИзВременногоХранилища(АдресДанных) СтрокаДанных.СсылкаНаСайтАвтора = ТабДок.ПолучитьОбласть("R"+Формат(НомерСтро ПутьКФайлуНаСервере = ки, "ЧГ=0")+"С"+5). Текущая Область. Текст; ПолучитьИмяВременногоФайла("xlsx"); Данные.Записать(ПутьКФайлуНаСервере); КонецЦикла; КонецПроцедуры Попытка &НаКлиенте Процедура Прочитать Файл XLSX() ТабДок.Прочитать(ПутьКФайлуНаСервере ДанныеФайла = Новый СпособЧтенияЗначенийТабличногоДокумента.Знач ДвоичныеДанные(Объект.ПутьКФайлу); ение); АдресДанных = ПоместитьВоВременноеХранилище(ДанныеФайла) Исключение Сообщение = Новый СообшениеПользователю: Прочитать Файл XLSX На Сервере (Адрес Д Сообщение. Текст = "Не удалось анных); прочитать указанный файл по причине: " + ОписаниеОшибки(); КонецПроцедуры Сообщение.Сообщить(); &НаСервере Возврат; КонецПопытки; Процедура Записать Данные На Сервере()

```
Для каждого СтрокаДанных из
Объект. Данные Файла Цикл
                                                      &НаКлиенте
                                                      Процедура Записать Данные (Команда)
              Найденный Автор =
Справочники. Авторы. Найти ПоРеквизиту ("Фамили
яАвтора", СтрокаДанных.Фамилия);
                                                             ЗаписатьДанныеНаСервере();
              Если Найденный Автор. Пустая()
Тогда
                                                      КонецПроцедуры
                      Найденный Автор =
Справочники. Авторы. Создать Элемент();
                                                      Код формы обработки «УстановкаЦенБилетов»
                                                      &НаКлиенте
                                                      Процедура Заполнить Таблицу (Команда)
       Найденный Автор. Наименование =
СтрокаДанных.Имя;
                                                        Заполнить Таблицу На Сервере();
       Найденный Автор. Фамилия Автора =
СтрокаДанных.Фамилия;
                                                      КонецПроцедуры
                      День =
Число(Лев(СтрокаДанных.ГодРождения, 2));
                                                      &НаСервере
                      Месяц =
                                                      Процедура Заполнить Таблицу На Сервере()
Число(Сред(СтрокаДанных.ГодРождения, 4, 2));
                      Гол =
                                                        Цены.Очистить();
Число(Прав(СтрокаДанных.ГодРождения, 4));
                      ГодРожденияДата =
                                                        Для Каждого Абонемент Из
Дата(Год, Месяц, День);
                                                      Перечисления.СрокДействия Абониментов Цикл
                                                          Для Каждого Возрастной Рейтинг Из
                                                      Перечисления.ВозрастнойРейтинг Цикл
       Найденный Автор. Год Рождения =
                                                            НоваяСтрока = Цены.Добавить();
ГодРождения Дата;
                                                            НоваяСтрока. Абонемент = Абонемент;
                                                            НоваяСтрока.ВозрастнойРейтинг =
       Найденный Автор. Национальность =
                                                      Возрастной Рейтинг;
СтрокаДанных.Национальность;
                                                            // Попробуем получить текущую цену из
       Найденный Автор. Ссылка На Сайт Автора =
                                                      регистра сведений
СтрокаДанных.СсылкаНаСайтАвтора;
                                                            Запрос = Новый Запрос;
                                                            Запрос.Текст =
       Найденный Автор. Записать();
                                                              "ВЫБРАТЬ ПЕРВЫЕ 1
              КонепЕсли:
                                                              ЦеныЧитательскихБилетов.Цена
                                                              |ИЗ
                                                      РегистрСведений. Цены Читательских Билетов КАК
       КонецЦикла;
       Сообщить("Добавление прошло успешно")
                                                      ЦеныЧитательскихБилетов
                                                              ГДЕ
```

КонецПроцедуры

```
ЦеныЧитательскихБилетов. Абонемент
= &Абонемент
                                                        Для Каждого Строка Из Цены Цикл
        | И
                                                          Запись =
Цены Читательских Билетов. Возрастной Рейтинг =
                                                      РегистрыСведений. Цены Читательских Билетов. Соз
&ВозрастнойРейтинг
                                                      датьМенеджерЗаписи();
        УПОРЯДОЧИТЬ ПО
                                                          Запись. Абонемент = Строка. Абонемент;
        ЦеныЧитательскихБилетов.Период
                                                          Запись.ВозрастнойРейтинг =
УБЫВ":
                                                      Строка. Возрастной Рейтинг;
                                                          Запись.Цена = Строка.Цена;
      Запрос. Установить Параметр ("Абонемент",
Абонемент);
                                                                    // Устанавливаем период записи
                                                                    Запись.Период = ТекущаяДата;
Запрос. Установить Параметр ("Возрастной Рейтинг",
Возрастной Рейтинг);
                                                          Запись.Записать();
                                                        КонецЦикла;
     Результат = Запрос.Выполнить();
      Выборка = Результат.Выбрать();
                                                        Сообщить("Цены успешно записаны.");
                      Если
                                                      КонецПроцедуры
Выборка.Следующий() Тогда
                             НоваяСтрока.Цена
                                                      Код формы обработки «Виджеты»
= Выборка.Цена;
                                                      &НаКлиенте
      Иначе
                                                      Перем Временное Хранилище Фотографии;
        НоваяСтрока.Цена = 0;
                      КонецЕсли;
                                                      &НаСервере
КонецЦикла;
                                                      Процедура
  КонецЦикла;
                                                      Заполнить Таблицу На Сервере (Выбранный Месяц)
                                                             Если НЕ
КонецПроцедуры
                                                      ЗначениеЗаполнено(ВыбранныйМесяц) Тогда
&НаКлиенте
                                                                    Выбранный Месяц =
                                                      ТекущаяДатаСеанса();
Процедура ЗаписатьЦены(Команда)
                                                             КонецЕсли;
  ЗаписатьЦеныНаСервере();
                                                             НачалоПериода =
                                                      НачалоМесяца(ВыбранныйМесяц);
КонецПроцедуры
                                                             КонецПериода =
                                                      КонецМесяца(ВыбранныйМесяц);
&НаСервере
                                                             // --- КОНЕЦ ИЗМЕНЕНИЙ ---
Процедура ЗаписатьЦеныНаСервере()
                                                             Запрос = Новый Запрос;
       // Получаем текущую дату сервера
                                                             Запрос.Текст =
       ТекущаяДата = ТекущаяДата();
                                                             "ВЫБРАТЬ ПЕРВЫЕ 5
```

ПопулярностьКнигиОбороты.Книга,	КонецЦикла;
ПопулярностьКнигиОбороты.Период	
КАК ПериодОборота, // Переименовал, чтобы не	ЗапросЗаказы = Новый Запрос;
путать с периодом запроса	ЗапросЗаказы.Текст =
1	"ВЫБРАТЬ ПЕРВЫЕ 10
СУММА(ПопулярностьКнигиОбороты.Количество	ЗакупкаКниг. Номер КАК Номер,
Оборот) КАК КоличествоОборот	ЗакупкаКниг.Дата КАК Дата,
ИЗ	ЗакупкаКниг.Поставщик КАК
I	Поставщик,
РегистрНакопления.ПопулярностьКниги.Обороты(	ЗакупкаКниг.Издатель КАК
&НачалоПериода, &КонецПериода, ДЕНЬ) КАК	Издатель,
ПопулярностьКнигиОбороты	ЗакупкаКниг.СуммаДокумента
ГДЕ	КАК СуммаДокумента,
ПопулярностьКнигиОбороты.Книга <>	ЗакупкаКниг.Статус КАК Статус
ЗНАЧЕНИЕ(Справочник.Книги.ПустаяСсылка)	ИЗ
1	Документ.ЗакупкаКниг КАК
СГРУППИРОВАТЬ ПО	ЗакупкаКниг
ПопулярностьКнигиОбороты.Книга,	ГДЕ
ПопулярностьКнигиОбороты.Период	ЗакупкаКниг.Статус ↔
1	ЗНАЧЕНИЕ(Перечисление.СтатусЗакупки.Получен
УПОРЯДОЧИТЬ ПО	)
1	1
СУММА(ПопулярностьКнигиОбороты.Количество	УПОРЯДОЧИТЬ ПО
Оборот) УБЫВ";	ЗакупкаКниг.Дата УБЫВ";
Запрос. Установить Параметр ("Начало Пери	РезультатЗаказы =
ода", НачалоПериода);	ЗапросЗаказы.Выполнить();
Запрос. Установить Параметр ("Конец Перио	ВыборкаЗаказы =
да", КонецПериода);	РезультатЗаказы.Выбрать();
Результат = Запрос.Выполнить();	Объект.ОжидаемыеЗаказы.Очистить();
Выборка = Результат.Выбрать();	
Объект.Книги.Очистить();	Пока ВыборкаЗаказы.Следующий() Цикл
	НоваяСтрока =
Пока Выборка.Следующий() Цикл	Объект.ОжидаемыеЗаказы.Добавить();
НоваяСтрока =	НоваяСтрока.Номер =
Объект.Книги.Добавить();	ВыборкаЗаказы.Номер;
НоваяСтрока.Книга =	НоваяСтрока.Дата =
Выборка.Книга;	ВыборкаЗаказы.Дата;
НоваяСтрока.Период =	НоваяСтрока.Поставщик =
Выборка.ПериодОборота;	ВыборкаЗаказы.Поставщик;
НоваяСтрока.Количество =	НоваяСтрока.Издатель =
Выборуз КоличествоОборот	Rufonva Savasti Manatenti

НоваяСтрока.СуммаДокумента = Процедура ЗагрузкаПользователя() ВыборкаЗаказы.СуммаДокумента; НоваяСтрока.Статус = ТекущийПользователь = ВыборкаЗаказы.Статус; ПользователиИнформационнойБазы.ТекущийПоль КонецЦикла; зователь(); КонецПроцедуры // Поиск сотрудника по логину Сотрудник = &НаКлиенте Справочники.Сотрудники.НайтиПоРеквизиту("Лог Процедура Заполнить Диаграмму() ин", ТекущийПользователь.Имя); ДиаграммаПопулярности.Очистить(); Если Сотрудник <> Неопределено Тогда Для Каждого СтрокаТаблицы Из // Приветствие Объект.Книги Цикл Элементы.Приветствие.Заголовок = "Добро пожаловать в систему, " + Серия = ДиаграммаПопулярности. Установить Серию (Строк Сотрудник. Наименование + "!"; // Или другой текст аТаблицы.Книга); приветствия Точка = ДиаграммаПопулярности. УстановитьТочку(Строка // ФИО и должность Таблицы.Период);  $\Phi MO =$ Сотрудник. $\Phi$ амилия + " " + Сотрудник. Наименование + " " + Точка. Текст = Формат(СтрокаТаблицы.Период, Сотрудник.Отчество; "ДФ=dd.MM.yyyy"); Должность = Сотрудник. Должность; ДиаграммаПопулярности. Установить Значе Элементы. ФИОДолжность. Заголовок = ние(Точка, Серия, СтрокаТаблицы.Количество); КонепЦикла:  $\Phi$ ИО + ", " + Должность; КонецПроцедуры // Фотография Если &НаКлиенте ЗначениеЗаполнено(Сотрудник. Фотография) Тогда Процедура ПриОткрытии(Отказ) ЭтотОбъект.Месяц= ТекущаяДата(); ВременноеХранилищеФотографии = Заполнить Таблицу На Сервере (Этот Объект. Сотрудник. Фотография. Получить(); // Получаем Месяц); картинку из хранилища Заполнить Диаграмму(); ФотографияСотрудника = ЗагрузкаПользователя(); Получить Навигационную Ссылку (Сотрудник. Ссыл ДанныеУчереждения(); ка, "Фотография"); ЭтотОбъект.Дата = ТекущаяДата(); КонецЕсли; КонецПроцедуры Иначе

&НаСервере

Элементы. Приветствие. Заголовок = "Добро пожаловать в систему!"; // Приветствие Элементы.ПолноеНаименование.Заголовок для пользователей без привязки к сотруднику = Выборка.ПолноеНаименование; КонецЕсли; Элементы.Адрес.Заголовок = "Адрес: " + Выборка. Адрес; КонецПроцедуры Элементы.Телефон.Заголовок = "Телефон: " + Выборка. Телефон; &НаСервере Элементы.Почта.Заголовок = Процедура Данные Учереждения() "Почта: " + Выборка.Почта; Элементы.Сайт.Заголовок = "Сайт: " + Выборка.Сайт; Запрос = Новый Запрос; Запрос. Текст = ФИОДиректора = "ВЫБРАТЬ ПЕРВЫЕ 1 Выборка. Директор Фамилия + " " + Выборка. ДиректорИмя + " " + Библиотека.ПолноеНаименование КАК ПолноеНаименование. Выборка. Директор Отчество; Библиотека. Адрес КАК Адрес, Элементы. Директор. Заголовок = Библиотека. Телефон КАК "Руководитель учереждерения: " + ФИОДиректора; Телефон, КонецЕсли; Библиотека. Почта КАК Почта, Библиотека.Сайт КАК Сайт. КонецПроцедуры Библиотека.ИНН КАК ИНН, Библиотека.КПП КАК КПП. &НаКлиенте Библиотека. Директор. Фамилия Процедура МесяцПриИзменении(Элемент) КАК ДиректорФамилия, Заполнить Таблицу На Сервере (Этот Объект. Библиотека. Директор. Наименование Месяц); КАК ДиректорИмя, ЗаполнитьДиаграмму(); Библиотека. Директор. Отчество КАК КонецПроцедуры ДиректорОтчество |ИЗ &НаКлиенте Справочник. Библиотека КАК Процедура Обновить(Команда) Библиотека Заполнить Таблицу На Сервере (Этот Объект. |АВТОУПОРЯДОЧИВАНИЕ"; Месяц); Результат = Запрос.Выполнить(); ЗаполнитьДиаграмму(); Выборка = Результат.Выбрать(); КонецПроцедуры

Если Выборка. Следующий () Тогда