

§1 БАЗЫ ДААННЫХ

Грибанова-Подкина М.Ю.

UML-МОДЕЛЬ ПАРТИОННОГО УЧЕТА ТОВАРА ДЛЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Аннотация: Объектом исследования является модуль партионного учета, являющийся составной частью автоматизированной информационной системы. Разработан проект реализации модуля. Проект представляет собой техническое описание функций и особенностей программной реализации партионного учета, а также объектов, входящих в его состав. Представленная модель дает четкое представление о компонентах автоматизированной информационной системы, которые должны быть развернуты для осуществления в ней партионного учета по методикам FIFO или LIFO, об алгоритмах реализации операций по поступлению, расходу и списанию товара. Модель построена с помощью CASE-средства Sparx Enterprise Architect в нотации UML2.1 и включает в себя диаграмму данных, вариантов использования, а также другие диаграммы, которые описывают детали поведения, реализации и развертывания модуля. Рассмотренный подход к автоматизации партионного учета основан на внесении некоторой степени избыточности в логическую модель базы данных, что позволяет реализовать более удобный с точки зрения реализации механизм регистрации поставок и расхода товаров и сырья. Предложенная модель предусматривает программную клиент-серверную реализацию различных вариантов: с использованием тонкого и толстого клиента, с выделенным слоем бизнес-логики.

Ключевые слова: диаграмма развертывания, диаграмма активностей, диаграмма последовательности, диаграмма данных, диаграмма вариантов использования, база данных, проектирование информационных систем, информационная система, партионный учет, UML

Abstract: The object of research is a module for accounting of supplied goods, which is part of the automated information system. The author presents a project for module implementation. The project contains a technical description of functions and features of the software implementation of accounting of supplied goods as well as the objects included in this software. The given model gives a

clear view of the automated information system components that must be deployed for accounting of supplied goods using FIFO or LIFO methods. The model shows algorithms for the implementation of operations of incoming, expenditure and cancellation of the goods. The model is constructed using the CASE-tools in UML 2.1 notation and includes a data chart, use case diagram, and other diagrams, which describe the details of behavior, implementing and deploying the module. The used approach to automating the accounting of supplied goods is based on including a certain degree of redundancy in the logical database model, which allows using a more convenient mechanism of registration of supply and consumption of goods and raw materials. The proposed model provides a software client-server implementation of the different options: using a thin and thick client with a dedicated business logic layer.

Keywords: *UML, deployment diagram, activity diagram, sequence diagram, use case diagram, data chart, database, information systems development, information system, accounting of supplied goods*

Введение

Задача учета движения товара как составная часть присутствует во многих экономических видах деятельности. Она включает в себя вопросы учета поступления и расхода товара и его партий [1, С.163]. При этом предполагается, что цена одного и того же товара в разных партиях не обязана быть одинаковой. Для движения различных по ценам партий товара (партионного учета) используются следующие схемы:

- списание партий товара по принципу очереди. Эта схема реализуется по принципу расхода сырья, начиная с самых ранних поставок и известна под названием FIFO (first-in-first-out – первым-вошел-первым-вышел);
- списание партий по принципу стека. Здесь расход сырья осуществляется, начиная с последних поставок, а схема называется LIFO (last-in-first-out – последним-вошел-первым-вышел);
- списание товара по средней стоимости.
- Первые две схемы интересны тем, что позволяют точно определить многие учетные данные:
- количественные остатки товаров и сырья по партиям;
- суммовые остатки по партиям;
- актуальную себестоимость продукции, полученной в результате использования сырья;
- актуальную прибыль продукции.

Реализация этих схем в информационных системах является актуальной задачей для разработчиков и сопряжена с необходимостью анализа способов формирования соответствующей структуры данных. Правильно построенная логическая модель базы данных значительно упрощает работу с информационным хранилищем, она дает возможность формировать интуитивно понятные и простые запросы, выполнять необходимые операции по занесению и извлечению данных. Традиционно для таких целей используется

преобразование базы данных, называемое нормализацией. Нормализация нацелена на выделение сущностей в отдельные таблицы и уменьшение избыточности информации.

В представленной модели используется несколько другой подход. Он основан на внесении некоторой степени избыточности в логическую модель базы данных. Это позволяет реализовать более прозрачный механизм регистрации поставок и расхода товаров и сырья.

1. Модель данных и общая концепция системы

Разработка проекта автоматизированной информационной системы ставит перед собой задачу технического описание объектов, функций и особенностей программной реализации. С учетом усложняющейся архитектуры современных приложений для различных предметных областей, возникает необходимость в создании комплексного проекта информационной системы, который включает в себя объектную модель и модели бизнес-функций системы [2, 3, 4].

Будем рассматривать модель решения задачи партионного учета, являющегося составной частью информационной системы.

Задача партионного учета включает в себя следующие функциональные элементы:

- учет поступления товара;
- учет расхода товара;
- учет списания товара.

Эти элементы функционально представляют собой приходно-расходные операции модуля количественно-суммового учета и представлены прецедентами на диаграмме вариантов использования (Use case) на рис. 1. Описание проекта этого модуля и представляет собой задачу данной работы. Внешним по отношению к модулю является некоторая система-клиент, также входящая в состав информационной системы.

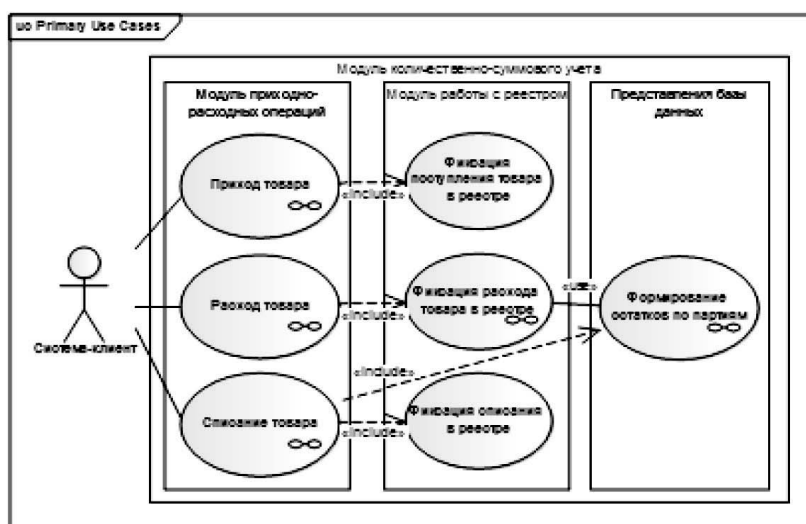


Рис.1. Диаграмма вариантов использования

Как видно из диаграммы, модуль количественно-суммового учета состоит из следующих логических частей, которые могут быть реализованы как подсистемы:

1. Модуль приходно-расходных операций.
2. Модуль работы с реестром.
3. Представления базы данных.

Операции прихода, расхода и списания товара принадлежат модулю приходно-расходных операций, модуль работы с реестром отвечает за регистрацию операций в реестре, в модуле представления базы данных реализуется формирование остатков по партиям.

Прецеденты из первого модуля включают себя соответствующие прецеденты из второго. Прецедент «Формирование остатков по партиям» включается в «Списание товара» и используется прецедентом «Фиксация расхода товара в реестре».

Для учета приходно-расходных операций используются таблицы базы данных, модель которой представлена на рисунке 2.

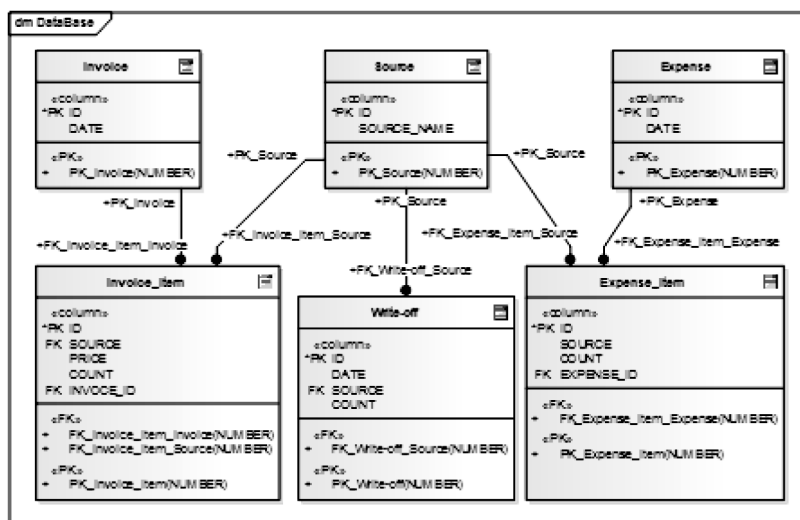


Рисунок 2. Модель базы данных

Перечень таблиц следующий:

- Source (товары, сырье) – наименования ресурсов;
- Invoice (накладная) – общие сведения по накладной: номер и дата;
- Invoice_item (позиция в накладной) – табличная часть накладной: приходуемый ресурс, его цена и количество;
- Expense (расход) – общие сведения по расходному документу: номер и дата;
- Expense_item (позиция в расходе) – табличная часть расхода: расходующий ресурс и его количество;
- Write-off (списание) – данные по списанию товара: номер и дата документа, ресурс и его количество.

Приведенный состав таблиц является необходимым для осуществления рассматриваемой модели партионного учета, но может быть расширен при реализации в конкретной информационной системе. Например, могут быть добавлены поставщики, потребители и т.д.

Рассматриваемый подход к реализации партионного учета, как уже отмечалось выше, подразумевает наличие информационной избыточности, которая реализуется с помощью дополнительной таблицы Register (реестр), представленной на рис. 3.

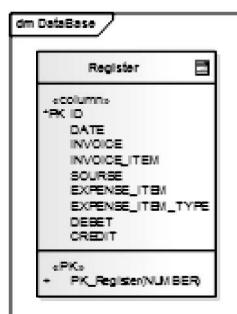


Рисунок 3. Модель реестра

Реестр аккумулирует в себе всю информацию по приходу (атрибут DEBET) и расходу (атрибут CREDIT) ресурсов. Каждая операция, помимо идентификатора и даты, имеет следующие атрибуты:

- INVOICE, INVOICE_ITEM – номер и позиция накладной, по которой осуществляется приход товара, или к которой привязывается расход;
- SOURCE – код приходуемого и расходуемого ресурса.
- Дополнительно при расходе ресурса указываются код позиции на расход (EXPENSE_ITEM), который представляет собой либо код позиции из расходного документа (таблица Expense_item), либо код документа на списание (таблица Write-off). Для однозначной идентификации этого атрибута используется дополнительный атрибут реестра – EXPENSE_ITEM_TYPE, который может принимать одно из текстовых значений: «EXPENSE» или «WRITE_OFF».

На основе данных реестра формируется представление базы данных, содержащее остатки по партиям. SQL-выборка для представления сортируется в порядке возрастания значения register.invoice_item для партионного учета FIFO, и по убыванию для учета LIFO. Соответственно, первая партия в данном представлении используется для расхода товара по любой из этих методик.

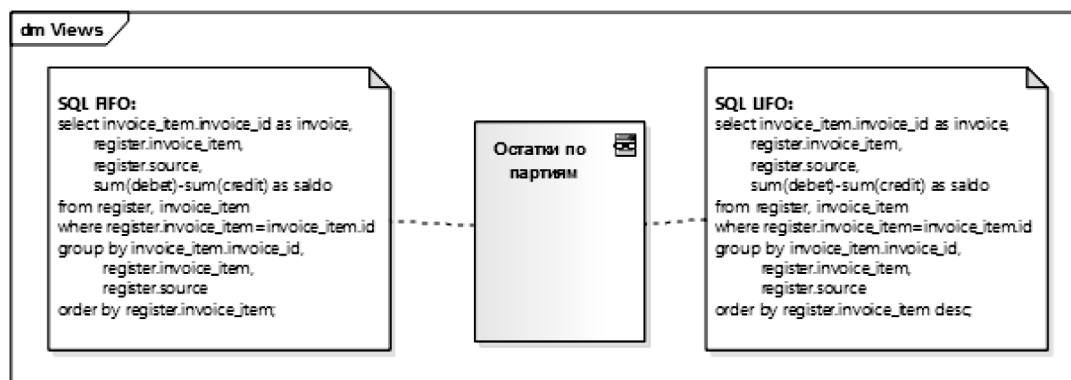


Рисунок 4. Представление базы данных

2. Описание вариантов использования

Процесс регистрации в системе поступления товара приведен на диаграмме из рис. 5. Он включает в себя следующие действия:

- создание новой приходной накладной;
- создание позиции в приходной накладной.

Эти операции вносят записи в таблицы базы данных Invoice (Приходная накладная) и Invoice_item (Позиция в приходной накладной).

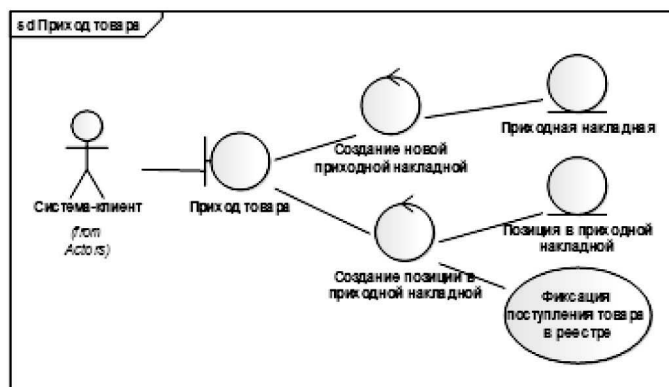


Рисунок 5. Диаграмма взаимодействия по прецеденту «Приход товара»

Создание новой позиции влечет за собой фиксацию поступления товара в реестре. Последний прецедент содержательно включает в себя операцию внесения записи в реестр о поступлении с указанием даты, номера накладной и ее позиции, кода приходного ресурса и его количества (атрибут DEBET таблицы Register).

Регистрация в системе факта расхода товара отражается в модели прецедентом «Расход товара» (см. рис. 1). Декомпозиция прецедента приведена на диаграмме взаимодействия на рис. 6.

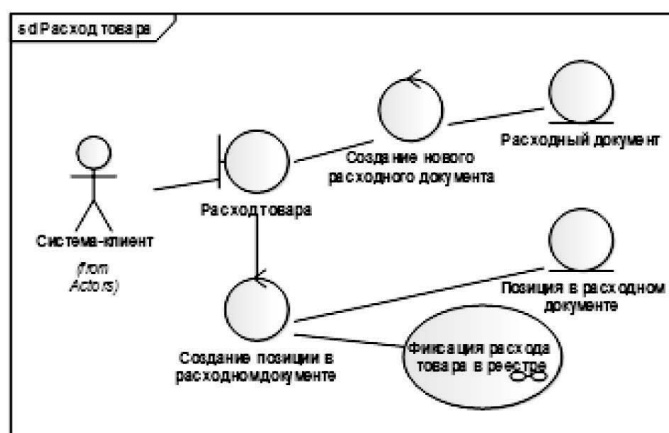


Рисунок 6. Диаграмма взаимодействия по прецеденту «Расход товара»

Таким образом, для учета расхода товара необходимо наличие расходного документа, который либо уже существует, либо вновь создается. Для этого документа создается позиция расхода и осуществляется фиксация расхода в реестре. Операция внесения записи о расходе в реестр является структурно более сложной, чем отражение поступления товара, за счет необходимости привязки расходуемого товара к имеющимся партиям. Требуемая последовательность действий приведена на диаграмме из рис. 7.

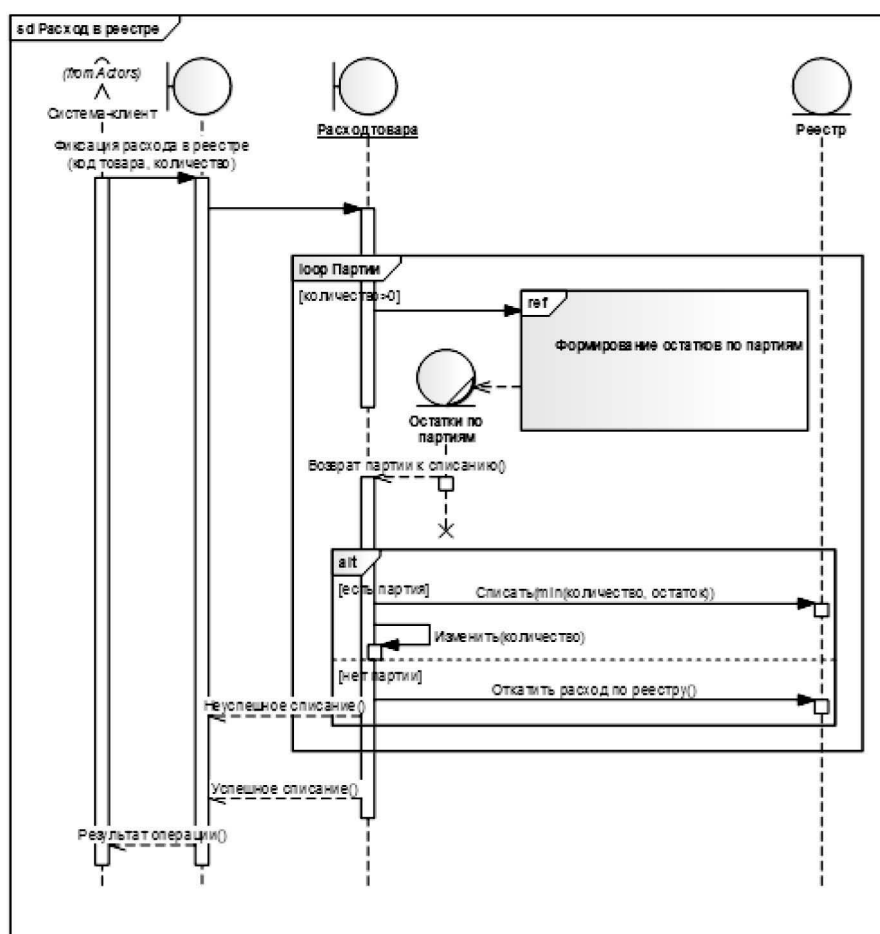


Рисунок 7. Диаграмма последовательности действий по прецеденту «Фиксация расхода товара в реестре»

Как видно из последней диаграммы, расход в реестре, в общем случае, отражается несколькими записями и является транзакцией. Пока количество товара к списанию больше нуля, необходимо выполнение следующих действий:

1. формирование остатков по партиям, в результате чего получаем временную сущность «Остатки по партиям», которая соответствует представлению из рис. 4. Этот же пункт отвечает за реализацию учета по технологии FIFO или LIFO;
2. получение из сущности «Остатки по партиям» партии на списание, которая идентифицируется номером приходной накладной и ее позицией;
3. при наличии партии товар списывается в количестве $\min(\text{количество}, \text{остаток})$ и

- уменьшается количество товара к списанию, в случае отсутствия партии все предыдущие записи по расходу откатываются;
- сведения о результате операции (успешна либо нет) передаются в вызывающий модуль.

Полная детализация одной итерации рассмотренной транзакции приведена на диаграмме активности на рис. 8.

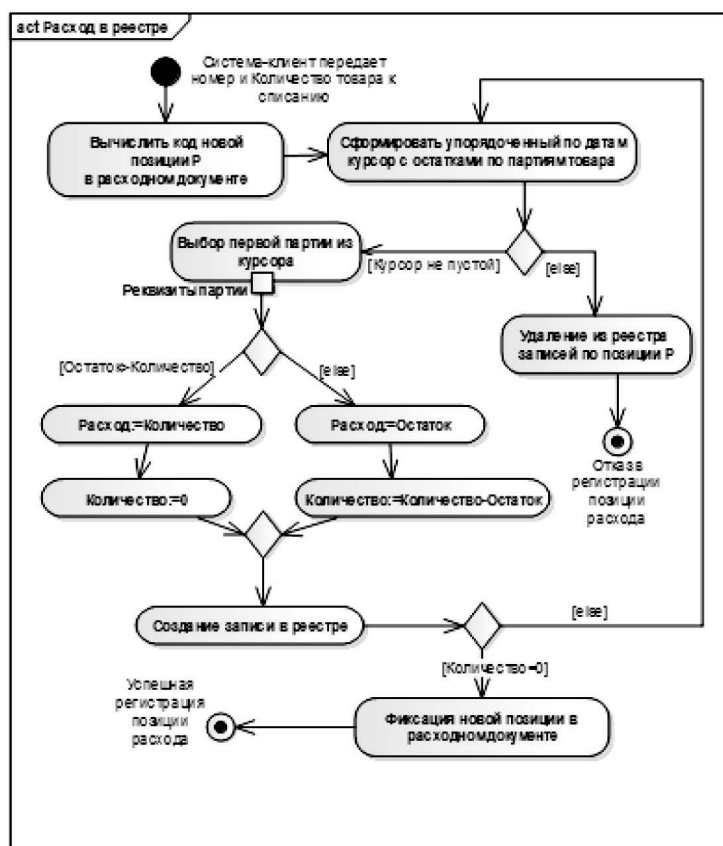


Рисунок 8. Диаграмма активности для отражения одной итерации расхода в реестре

Диаграмма активности предлагает наглядное изложение алгоритма расхода, в котором записи в реестре формируются до тех пор, пока либо количество к списанию не станет равным нулю (тогда списание успешно), либо не останется партий, к которым можно привязывать расход (в этом случае списание неуспешно и транзакция откатывается). Расходуемое количество товара в каждой итерации равно текущему количеству, если остаток превышает требуемый расход, и равно остатку в противном случае. В первом случае последующие итерации не потребуются.

Для отражения расхода в реестре атрибут EXPENSE_ITEM должен содержать код расходного документа (таблица Expense_item), а атрибут EXPENSE_ITEM_TYPE должен получить значение «EXPENSE».

Описание объектов, необходимых для реализации прецедента «Списание товара»,

приведено на диаграмме взаимодействия на рис. 9.

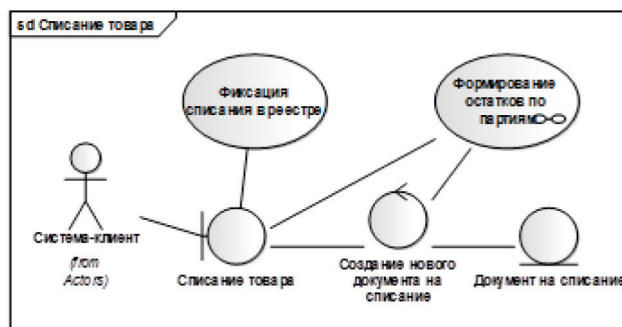


Рисунок 9. Диаграмма взаимодействия по прецеденту «Списание товара»

Существенной особенностью реализации операции списания является предоставление интерфейса для выбора партии, из которой необходимо списывать товар. Поэтому создание остатков по партиям используется для формирования общей картины остатков и возможности выбора нужной партии. Это отражено на диаграмме активности на рис. 10.

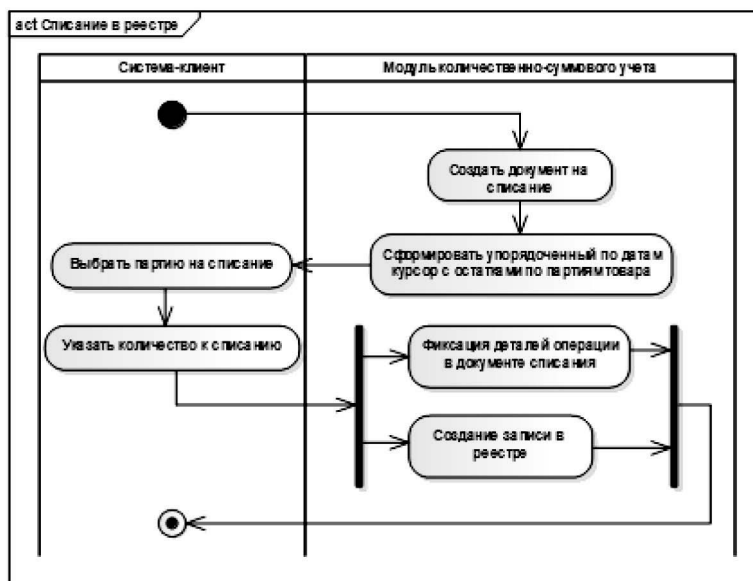


Рисунок 10. Диаграмма активности для прецедента «Списание товара»

На рис. 10 операции разделены по двум дорожкам:

1. Система-клиент: операции выполняются на клиенте проектируемого модуля;
2. Модуль количественно-суммового учета: операции выполняются в проектируемом модуле.

Таким образом, в системе-клиенте необходимо предусмотреть наличие интерфейса для отображения остатков по партиям, выбора партии и указания количества к списанию. В модуле количественно-суммового учета обеспечивается формирование остатков по

партиям (представление из рис. 4), а также создание документа на списание и записи в реестре. Необходимо лишь добавить, что для отражения списания в реестре атрибут EXPENSE_ITEM должен содержать код документа на списание (таблица Write-off), а атрибут EXPENSE_ITEM_TYPE должен получить значение «WRITE_OFF».

Рассмотренные прецеденты составляют исчерпывающий функционал модуля количественно-суммового учета.

3. Развертывание модуля

Для реализации представленных в проекте информационной системы прецедентов предлагается создание компонентов, представленных на рис. 11:

- библиотека приходно-расходных операций, включающая в себя процедуры создания записей о приходе, расходе и списании в таблицах Invoice, Invoice_item, Expense, Expense_item, Write-off;
- библиотека работы с реестром, содержащая операции по формированию в реестре записей о приходно-расходных операциях;
- база данных.

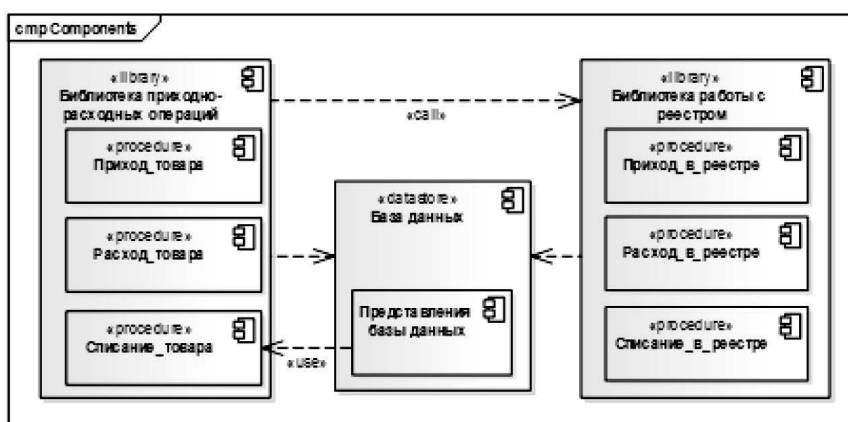


Рисунок 11. Диаграмма компонентов

Описанные компоненты при реализации модуля количественно-суммового учета должны развертываться согласно общей концепции автоматизированной информационной системы.

Для клиент-серверной системы, построенной по принципу толстого клиента, обе рассмотренные библиотеки будут располагаться на клиенте. Взаимодействие между клиентом и базой данных будет осуществляться посредством драйвера базы данных и включать в себя:

1. создание записей в таблицах базы данных согласно вызванным процедурам;
2. получение сведений по остаткам партий из базы данных.

Диаграмма разворачивания модуля для толстого клиента представлена на рис. 12.

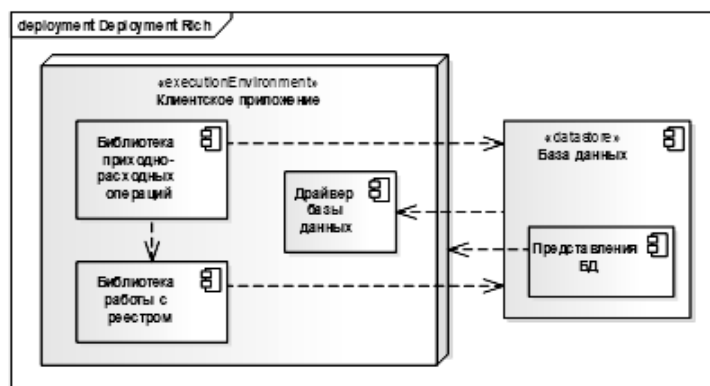


Рисунок 12. Диаграмма разворачивания для толстого клиента

Подробное описание реализации модуля количественно-суммового учета для толстого клиента, построенного на взаимодействии СУБД Microsoft Access и Delphi, представлено в [5].

Клиент-серверная система, построенная по принципу тонкого клиента, предусматривает размещение функционала на сервере (сервер СУБД), а презентационной логики – на клиенте. С целью их взаимодействия на клиент устанавливается драйвер базы данных. Функционал в этом случае реализуется с помощью хранимых процедур (см. рис. 13).

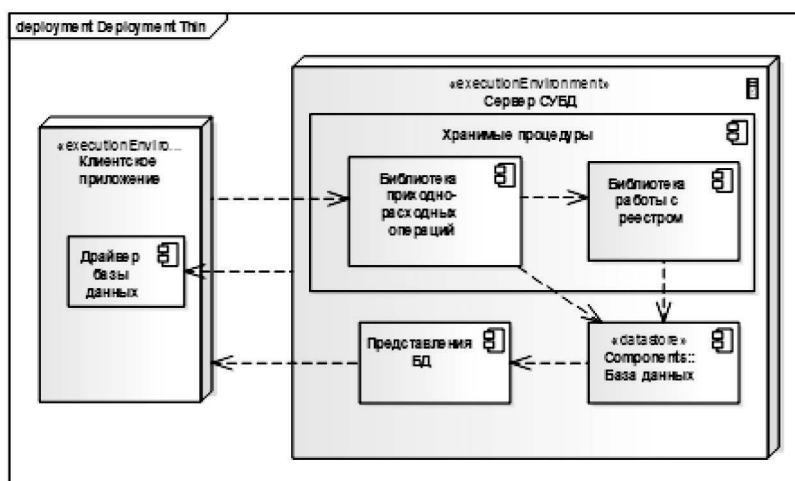


Рисунок 13. Диаграмма разворачивания для тонкого клиента

Большое распространение в последнее время получили многозвенные приложения для различных предметных областей, которые выделяют различные архитектурные слои под функции презентационной логики, бизнес-логики и системы управления данными [6,7,8]. Они могут быть реализованы двумя основными способами:

- в виде web-приложения, где промежуточным слоем между клиентом и СУБД явля-

- ется web-сервер, который обрабатывает бизнес-логику приложения;
 - в виде enterprise-приложения, где за бизнес-логику отвечает сервер приложений.
- В любом варианте на среднее звено попадают процедуры по обработке приходно-расходных операций в той реализации, которая предусмотрена используемой платформой, например, сервлеты и JSP, серверные компоненты и др.

На рис. 14 приведена диаграмма развертывания модуля количественно-суммового учета для архитектуры с выделенным сервером приложений.

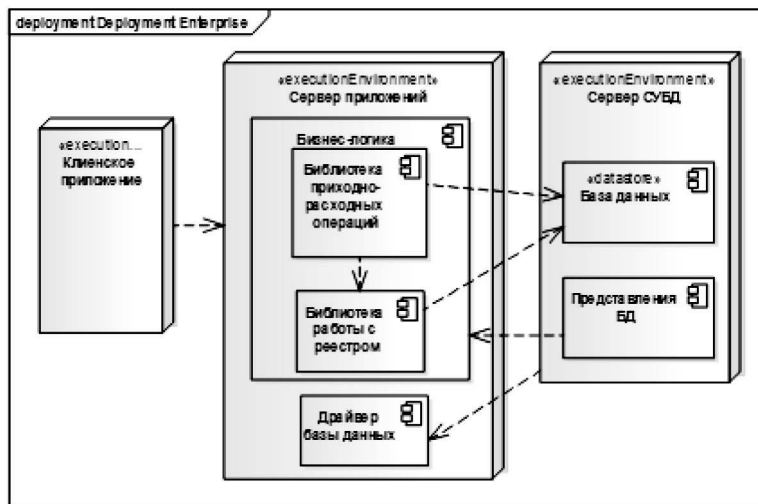


Рисунок 14. Диаграмма развертывания для выделенного сервера приложений

Заключение

Операции, связанные с учетом движения товара, присутствует во многих отраслях экономической деятельности: розничная и оптовая торговля, ресторанный бизнес, производство, сельское хозяйство и пр. Несмотря на обилие существующих систем, автоматизирующих учет в этих отраслях, всегда будет появляться необходимость в новых разработках, в том числе эксклюзивных под конкретного заказчика. Рассмотренная модель предлагает такую концепцию разработки модуля партионного учета товара, которая с успехом может быть реализована на любой платформе. А приведенный уровень детализации проекта позволяет ему выступать в качестве полного технического описания к программной реализации.

Библиография :

1. Кондраков Н.П. Бухгалтерский учет: Учебник / Н.П. Кондраков. М.: ИНФРА-М, 2007. 592 с.
2. Голосовский М.С. Информационно-логическая модель процесса разработки программного обеспечения // Программные системы и вычислительные методы. – 2015. № 1. С. 59-68.
DOI: 10.7256/2305-6061.2015.1.14119

3. Шумский Л.Д. Семантическая трассировка информационных процессов. // Программные системы и вычислительные методы. 2014. № 1. С. 80-92. DOI: 10.7256/2305-6061.2014.1.11362
4. Грибанова-Подкина М.Ю., Аннушкин А.В. Модель информационной системы учета движения товара в компьютерном магазине // Технические науки – от теории к практике. 2016. № 4(52). Новосибирск: Изд. АНС «СибАК», 2016. С. 44-51.
5. Грибанова-Подкина М.Ю. Программная реализация учета товара по технологии FIFO // Программные системы и вычислительные методы. 2014. № 4. С. 411-417. DOI: 10.7256/2305-6061.2014.4.13738
6. Сокольников А.М. Сравнительный анализ подходов к разработке архитектуры и систем управления базами данных для высоконагруженных WEB-сервисов // NB: Кибернетика и программирование. 2014. № 4. С. 1-13. DOI: 10.7256/2306-4196.2014.4.12800. URL: http://e-notabene.ru/kp/article_12800.html
7. Ле В.Н., Панченко Д.П.. Программная реализация медицинской экспертной системы дифференциальной диагностики // Программные системы и вычислительные методы. 2014. № 3. С. 291-297. DOI: 10.7256/2305-6061.2014.3.11162
8. Куракин П.В. Специализированные системы математических расчетов нового поколения // Программные системы и вычислительные методы. 2016. № 1. С. 80-94. DOI: 10.7256/2305-6061.2016.1.17997

References:

1. Kondrakov N.P. Bukhgalterskii uchet: Uchebnik / N.P. Kondrakov. M.: INFRA-M, 2007. 592 s.
2. Golosovskii M.S. Informatsionno-logicheskaya model' protsessa razrabotki programmnoho obespecheniya // Programmnye sistemy i vychislitel'nye metody. – 2015. № 1. С. 59-68. DOI: 10.7256/2305-6061.2015.1.14119
3. Shumskii L.D. Semanticheskaya trassirovka informatsionnykh protsessov. // Programmnye sistemy i vychislitel'nye metody. 2014. № 1. С. 80-92. DOI: 10.7256/2305-6061.2014.1.11362
4. Gribanova-Podkina M.Yu., Annushkin A.V. Model' informatsionnoi sistemy ucheta dvizheniya tovara v komp'yuternom magazine // Tekhnicheskie nauki – ot teorii k praktike. 2016. № 4(52). Novosibirsk: Izd. ANS «SibAK», 2016. S. 44-51.
5. Gribanova-Podkina M.Yu. Programmная realizatsiya ucheta tovara po tekhnologii FIFO // Programmnye sistemy i vychislitel'nye metody. 2014. № 4. С. 411-417. DOI: 10.7256/2305-6061.2014.4.13738
6. Sokol'nikov A.M. Sravnitel'nyi analiz podkhodov k razrabotke arkhitektury i sistem upravleniya bazami dan-nykh dlya vysokonagruzhennykh WEB-servisov // NB: Kibernetika i programmirovaniye. 2014. № 4. S. 1-13. DOI: 10.7256/2306-4196.2014.4.12800. URL: http://e-notabene.ru/kp/article_12800.html
7. Le V.N., Panchenko D.P.. Programmная realizatsiya meditsinskoi ekspertnoi sistemy differentsial'noi diagnostiki // Programmnye sistemy i vychislitel'nye metody. 2014. № 3. С. 291-297. DOI: 10.7256/2305-6061.2014.3.11162
8. Kurakin P.V. Spetsializirovannyye sistemy matematicheskikh raschetov novogo pokoleniya // Programmnye sistemy i vychislitel'nye metody. 2016. № 1. С. 80-94. DOI: 10.7256/2305-6061.2016.1.17997