

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Удмуртский государственный университет»
Институт математики, информационных технологий и физики
Кафедра теоретических основ информатики

**Методические указания по выполнению курсовой
работы для студентов направления подготовки
«Прикладная информатика»**

методическое пособие

Ижевск 2019

УДК 004.438(075.8)

ББК 32.973.22я73

М545

Рекомендовано к изданию Учебно-методическим советом УдГУ.

Составитель: А.Ю. Сапаров, кандидат технических наук, доцент кафедры теоретических основ информатики ИМИТиФ

Рецензент: В.И. Родионов, кандидат физико-математических наук, доцент, заведующий кафедрой информатики и математики

М545 Методические указания по выполнению курсовой работы для студентов направления подготовки «Прикладная информатика»: методическое пособие / Сост.: А.Ю. Сапаров, Ижевск, 2019. 69 с.

Пособие посвящается вопросам выполнения курсовой работы. Содержит краткие указания при выборе темы, решению задач, составлению отчета и защите. Знания и навыки, полученные при выполнении курсовой работы, будут полезны в дальнейшем при подготовке выпускной квалификационной работы.

Разработано студентам второго и третьего курсов направления подготовки «Прикладная информатика», а также рекомендовано студентам направлений подготовки «Информатика и вычислительная техника», «Фундаментальная информатика и информационные технологии», выполняющих курсовую работу по дисциплинам «Базы данных» и «Автоматизированные информационные системы». Также рекомендуется студентам других направлений подготовки, занимающихся проектированием баз данных и разработкой автоматизированных информационных систем средствами «1С:Предприятия».

УДК 004.438(075.8)

ББК 32.973.22я73

© Сост.: А.Ю. Сапаров, 2019

© Институт математики, информационных технологий и физики, 2019

Введение

Данное методическое пособие предназначено для студентов второго и третьего курса направления подготовки «Прикладная информатика», а также рекомендовано студентам направлений подготовки «Информатика и вычислительная техника», «Фундаментальная информатика и информационные технологии», выполняющих курсовую работу по дисциплинам «Базы данных» и «Автоматизированные информационные системы». Согласно учебным планам, курсовая работа выполняется на кафедре, которая реализует эту дисциплину. На выполнение каждой курсовой работы дается один семестр. Каждому студенту назначается научный руководитель из состава преподавателей кафедры. Цель курсовой работы заключается в освоении навыков самостоятельной работы по указанной теме. В результате выполнения курсовой работы студент должен научиться правильно ставить цели и задачи, выполнять обзор литературы, самостоятельно решать задачи, правильно оформлять отчеты по проделанной работе и уверенно выступать перед экзаменационной комиссией. Опыт написания курсовых работ должен сыграть решающую роль при выполнении выпускной квалификационной работы.

Пособие содержит краткие указания по выполнению курсовой работы, а именно: рекомендации по выбору темы, выполнению исследовательской и практической работы, составлению отчета и защите. Указания сопровождаются фрагментами из примера курсовой работы на тему «Создание автоматизированной информационной системы для ФАП в программе 1С:Предприятие».

Курсовая работа должна способствовать развитию навыков проектирования баз данных и информационных систем, что будет полезно в дальнейшем при выполнении выпускной квалификационной работы и трудоустройстве.

В результате решения поставленных задач у студента должны сформироваться некоторые профессиональные и общепрофессиональные компетенции, в том числе способность эксплуатировать и сопровождать информационные системы и сервисы; способность анализировать рынок программно-технических средств, информационных продуктов и услуг для создания и модификации информационных систем; способность осваивать методики использования программных средств для решения практических задач.

1 Выбор темы

Курсовая работа состоит из 4 основных этапов:

1. выбор темы;
2. решение задачи;
3. составление отчета по работе;
4. защита.

Во многом от правильно выполненного первого этапа зависит успешность завершения всех остальных. Правильный выбор темы первой курсовой работы, при возможности ее дальнейшего развития, может также быть залогом успешной защиты выпускной квалификационной работы. В связи с этим нужно ответственно подходить к выбору темы работы.

При выборе темы необходимо учитывать следующие критерии:

1. Актуальность. Результаты предстоящей работы должны быть востребованы в современном мире.
2. Новизна. Не нужно делать то, что уже есть.
3. Практическая либо теоретическая значимость. Результаты работы должны принести реальную пользу обществу. Если говорить о теоретической значимости, то результаты должны помочь в других исследованиях.
4. Интерес. Выбор темы нужно выполнять с учетом своих предпочтений. Работа должна быть интересна исполнителю, чтобы он не только мог, но и хотел в полной мере погрузиться в нее.

Порядок действий при выборе темы курсовой:

1. Консультация с научным руководителем. Необходимо выполнить предварительные договоренности о графике консультаций.
2. Выбор темы из списка, предложенного научным руководителем или кафедрой. Список тем может содержать только названия направлений, а не конкретные задачи. Окончательный вариант названия составляется в ходе обсуждений с научным руководителем.
3. Если студенту не подходит ни одна из тем в списке, то он может предложить свою тему, удовлетворяющую всем критериям.

4. Предварительное изучение литературы с целью оценки возможности и целесообразности выполнения работы по выбранной теме, а также составления предварительного плана.
5. Согласование с научным руководителем цели работы и списка задач. При выделении цели следует обращать внимание на то, что в ней должна быть заложена полезность. Например, неправильно указывать в качестве цели разработку ПО. Разработка ПО может быть рассмотрена как задача, направленная на достижения чего-либо, так как разработка сама по себе не несет полезности (кроме как учебной), а только требует некоторых трудозатрат. Правильно будет, например: цель — ускорить процесс ввода текста на компьютере, задача — разработать программу автоматического распознавания текстов. Таким образом, цель — это нечто, имеющее полезность, а задача — это нечто, что требует приложения некоторых усилий от исполнителя.
6. Согласование с научным руководителем итогового названия темы. Правильное название темы должно в себе содержать цель, объект и предмет исследования. В то же время название не должно быть слишком длинным.
7. Написание заявления на утверждение темы. Важно понимать, что название, указанное в заявлении, будет включено в приказ. После этого изменение темы хотя бы на одну букву не допустимо. Необходимо с момента начала выполнения и до окончания защиты использовать именно это название, в противном случае работа может быть не зачтена.

Пример. Пусть студента заинтересовала тема, которая в списке значится как «Разработка в системе 1С:Предприятие». В таком виде данное название не может быть использовано в работе, поэтому требуется конкретизация. Допустим, нам нужно автоматизировать часть работы фельдшерско-акушерского пункта (далее ФАП), а именно, упростить процесс составления отчетов по расходам и остаткам медикаментов. Перейдем к формулировке цели. Согласно критерию, цель и задачи можно сформулировать следующим образом:

«Цель работы: Автоматизация процесса составления отчетов по расходам и остаткам медикаментов в фельдшерско-акушерском пункте.

Для достижения поставленной цели необходимо решить следующие **задачи:**

1. Анализ предметной области.

2. Формулировка требований.
3. Разработка системы.
4. Тестирование системы.
5. Внедрение системы.»

Итоговое название темы может быть в виде: «Создание автоматизированной информационной системы для ФАП в программе 1С:Предприятие» или «Автоматизация отчетности ФАП в системе 1С:Предприятие».

2 Решение задачи

Для достижения поставленной цели должны быть решены все поставленные задачи.

2.1 Анализ предметной области

Многие студенты не обращают внимания на пункт анализ предметной области, считая, что это не так важно, как этап разработки. Не стоит забывать о нем, так как правильно выполненный анализ поможет в дальнейшем избежать ошибок и необходимости выполнять ненужную работу. Задачу можно разделить на несколько этапов:

1. Анализ требований и информационных потребностей. На этом этапе мы должны получить у постановщика задачи как можно больше информации о том, что конкретно нужно реализовать. Например, если задачу ставит какая-то сторонняя организация, то она должна предоставить полный список того, что они хотят получить.

2. Выявление информационных объектов и связей между ними. На этом этапе необходимо выяснить, с какими данными нужно будет работать, в каком виде будут они храниться, и как будет организована связь между данными.

3. Построение модели предметной области и проектирование схемы БД. На этом этапе необходимо построить графические схемы, которые в полной мере отражают ту часть предметной области, которая затрагивается в решаемой задаче. Например, для описания базы данных может быть построена соответствующая ER-диаграмма.

Пример.

Для того, чтобы выяснить все подробности, общаемся с сотрудником ФАП, который и будет основным пользователем разрабатываемой системы. От него получаем следующую информацию:

«Необходимо создать автоматизированную информационную систему для учета номенклатуры лекарственных препаратов в ФАП.

Периодически при необходимости сотрудник ФАП заказывает те или иные медикаменты. Поступление сопровождается накладной, которая имеет уникальный номер. В накладной указывается дата поступления, а также приводится список медикаментов с указанием количества упаковок и цены за одну упаковку.

В конце каждого отчетного периода сотрудник ФАП должен составить отчет об израсходованных медикаментах за указанный период и их остатках на конец отчетного периода. Особенностью является то, что расход не обязательно должен быть в полных упаковках (например, расход за месяц составил 2,5 упаковки).»

Исходя из полученной информации, делаем вывод, что основные объекты — это данные о имеющейся номенклатуре. Главная таблица будет называться «Номенклатура». Она будет хранить номер накладной, дату поступления, название, цену, количество единиц в каждой упаковке. Название будет указано в накладной, и с учетом того, что для одних и тех же медикаментов в разных накладных они могут отличаться, потребуется дополнительная таблица со стандартными наименованиями. Она позволит группировать номенклатуру по наименованию, если даже в накладной будут указаны разные имена. Также потребуется таблица, которая будет хранить данные о расходе номенклатуры.

Строим ER-диаграмму, в которой будет 3 таблицы: «Номенклатура», «Названия» и «ИсторияРасхода» (рис. 1). Между таблицами «Номенклатура» и «Названия» будет связь многие-к-одному, а между «Номенклатура» и «ИсторияРасхода» — один-ко-многим.



Рис. 1. ER-диаграмма

2.2 Формулировка требований

Формулировка требования — это некий документ, содержащий подробное описание всех требований: реализуемые функции и структуры входных и выходных данных. Требования должны быть сформулированы четко, в максимально формализованном виде, чтобы было единое понимание как у постановщика задачи, так и у исполнителя. В наиболее простом виде это может быть обычный текст, в котором все используемые термины описаны и каждая фраза однозначна. В идеале это должно быть развернутое техническое задание (ТЗ).

Пример.

Входные данные:

- Накладная с указанным номером и датой. Накладная содержит список медикаментов с указанием названия, цены и количества упаковок. Список может содержать как лекарственные препараты, так и другие медицинские средства (шприцы, бинты и т.д.)
- Данные о расходе. В конце каждого рабочего дня сотрудник ФАП фиксирует данные об израсходованных медикаментах: из какой партии (номер накладной) израсходованы медикаменты, сколько полных упаковок израсходовано и сколько единиц в открытых упаковках израсходовано.

Должен быть реализован следующий функционал:

- просмотр информации об имеющейся номенклатуре;
- ввод данных о новом приходе;
- ввод данных о расходе;

- составление отчета об израсходованных медикаментах за указанный период и остатках на конец отчетного периода.

Выходные данные:

Отчет об израсходованных медикаментах за указанный период и остатках на конец отчетного периода. Отчет должен быть в виде таблицы. Колонки таблицы: Название, Номер накладной, Цена, Расход полных упаковок, Расход в открытых упаковках, Стоимость расхода, Остаток полных упаковок, Остаток в открытых упаковках, Остаточная стоимость. Строки таблицы: Номенклатура, сгруппированная по стандартному наименованию.

2.3 Разработка системы

Разработка выполняется с использованием средств программирования, которые согласованы между постановщиком задачи и исполнителем, либо между научным руководителем и студентом. Разработка должна вестись с учетом всех соответствующих стандартов. Например, названия методов, переменных, классов должны быть осмысленными; тексты модулей — правильно отформатированы (использовать отступы для вложенных конструкций, каждое действие должно быть на отдельной строке, между разными функциональными блоками оставлять пустые строки); остальные требования — согласно выбранному языку программирования.

Пример.

Создадим новую базу. Для этого запускаем «1С:Предприятие». Если откроется окно с текстом «Список информационных баз пуст. Добавить в список информационную базу?», то нажимаем Да. Если окна с вопросом нет, то нажимаем на кнопку Добавить (рис. 2).

Выбираем «Создание новой информационной базы» (рис. 3) и нажимаем Далее.

Выбираем «Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы» (рис. 4) и нажимаем Далее.

Указываем имя информационной базы, которое будет отображаться в списке баз (рис. 5) и нажимаем Далее.

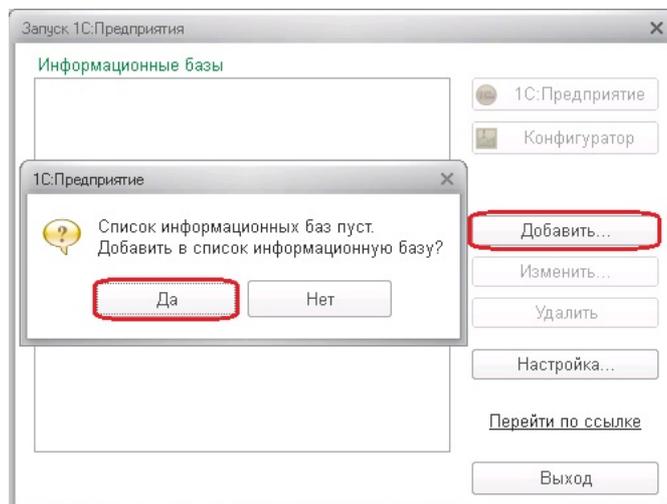


Рис. 2. Создание базы (шаг 1)

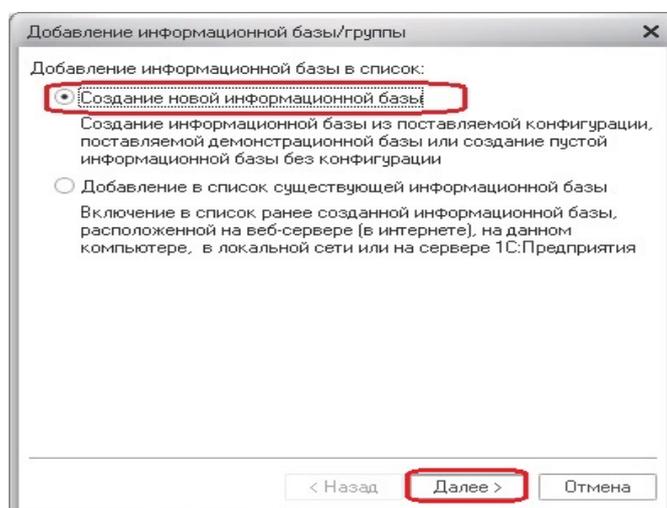


Рис. 3. Создание базы (шаг 2)

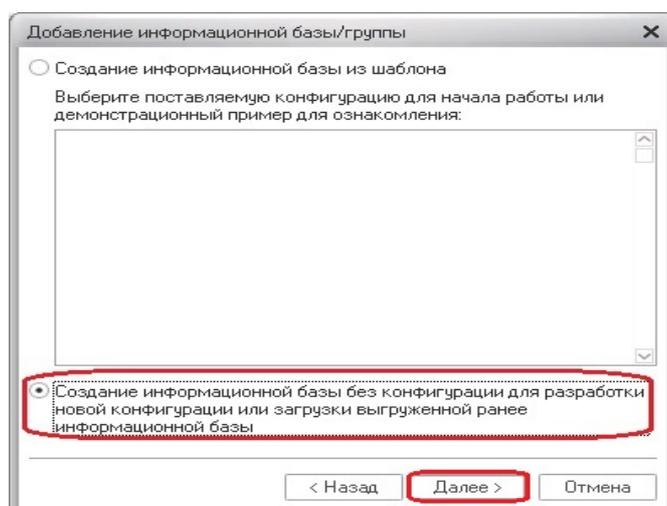


Рис. 4. Создание базы (шаг 3)

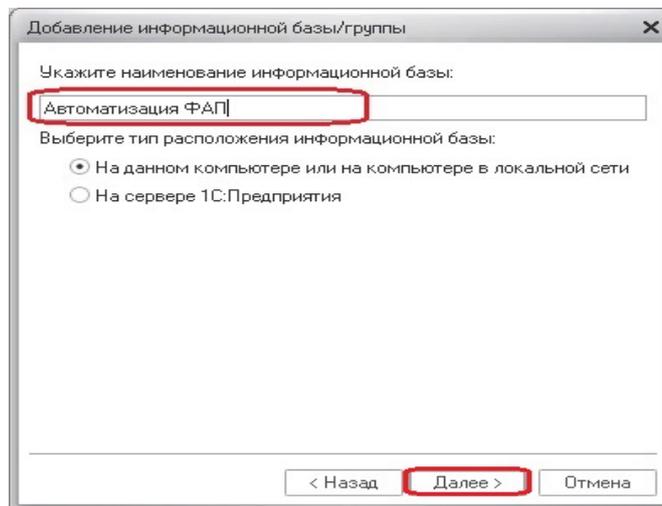


Рис. 5. Создание базы (шаг 4)

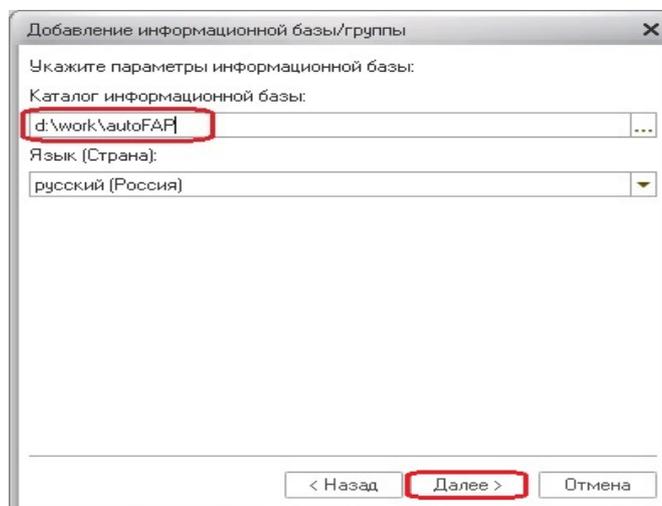


Рис. 6. Создание базы (шаг 5)

Указываем путь к каталогу на диске, в котором будут храниться файлы информационной базы (рис. 6) и нажимаем Далее.

Нажимаем Готово (рис. 7).

Чтобы приступить к разработке, запускаем информационную базу в режиме конфигуратора. Для этого в списке слева выбираем добавленную базу и нажимаем на кнопку Конфигуратор (рис. 8).

В меню открывшегося конфигуратора выбираем Конфигурация -> Открыть конфигурацию (рис. 9). Слева откроется окно конфигурации, в которой и должен быть описан весь функционал.

Указываем название и синоним конфигурации. Для этого кликаем два раза

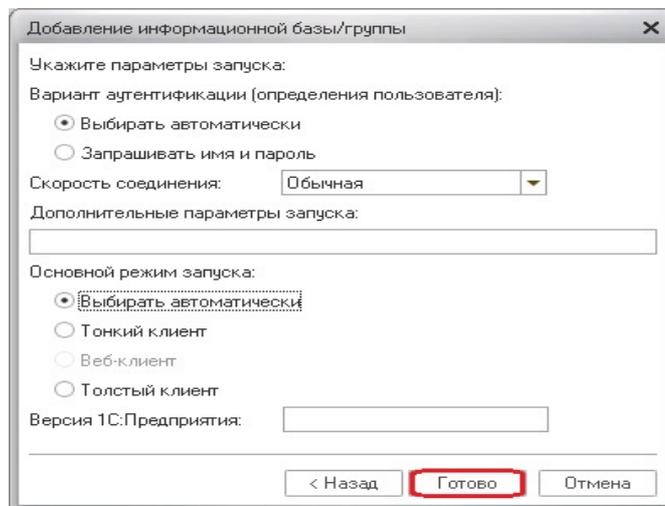


Рис. 7. Создание базы (шаг 6)

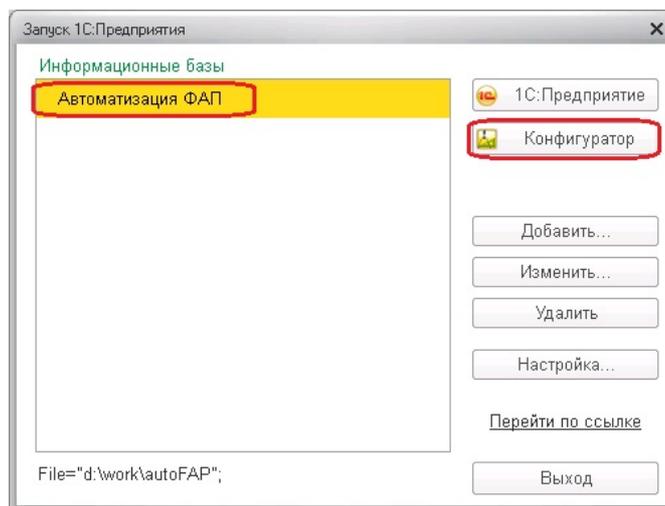


Рис. 8. Запуск конфигуратора

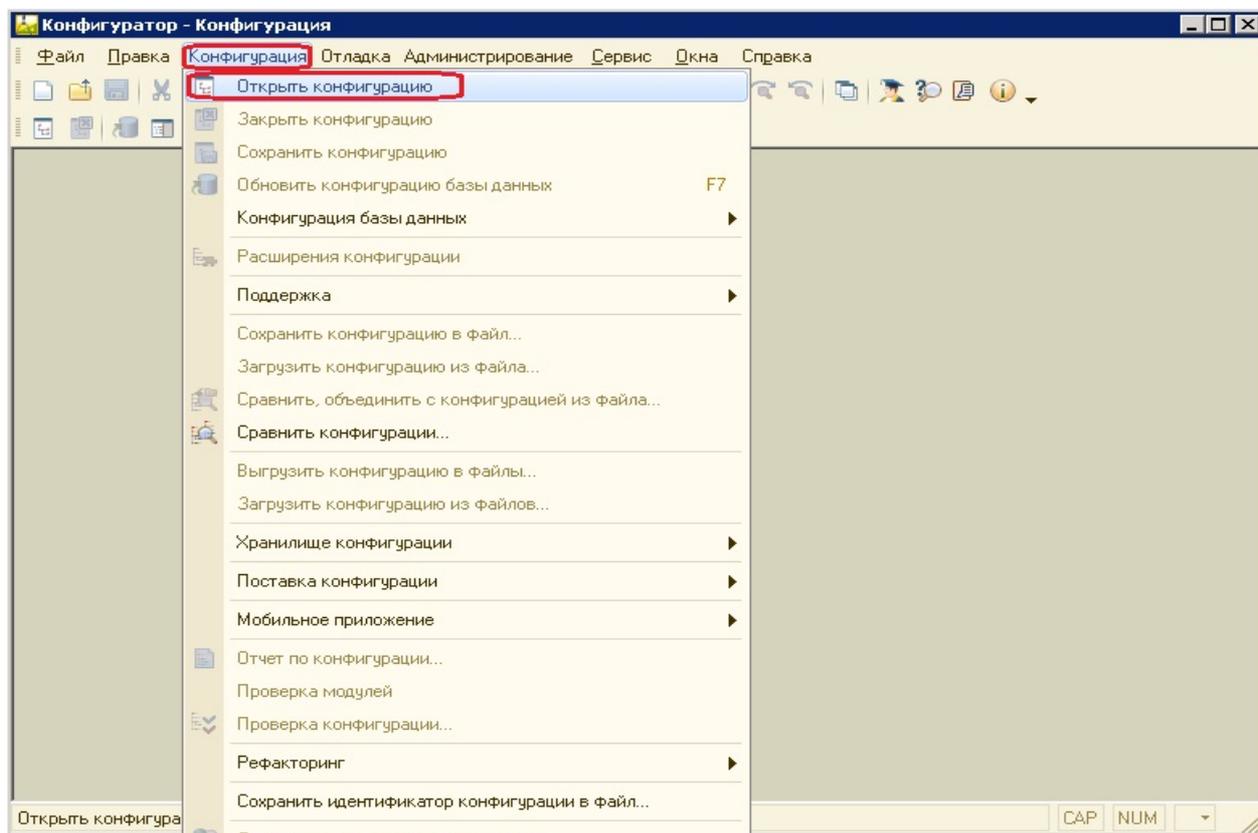


Рис. 9. Открытие конфигурации

на корень дерева конфигурации и в открывшемся окне заполняем соответствующие поля (рис. 10). Чтобы не потерять изменения, сохраняем конфигурацию нажатием соответствующей кнопки в меню либо комбинацией клавиш Ctrl+S.

Добавим справочник Номенклатура, в котором и будут храниться основные сведения. Для этого правой кнопкой мыши нажимаем на элемент «Справочники» в дереве конфигурации и выбираем пункт меню Добавить. В открывшемся окне указываем имя справочника (рис. 11). Каждый элемент справочника будет соответствовать одной строке накладной.

Сохраняем изменения в конфигурации. Чтобы применить изменения к базе данных, необходимо нажать клавишу F7 и в появившемся окне нажать на кнопку Принять (рис. 12).

Добавляем реквизиты в справочник. Для этого правой кнопкой мыши нажимаем на пункт Реквизиты у справочника Номенклатура и нажимаем на Добавить. Добавляем следующие реквизиты (рис. 13):

- *НомерНакладной* с типом строка, который будет хранить информацию о номере накладной поступления;

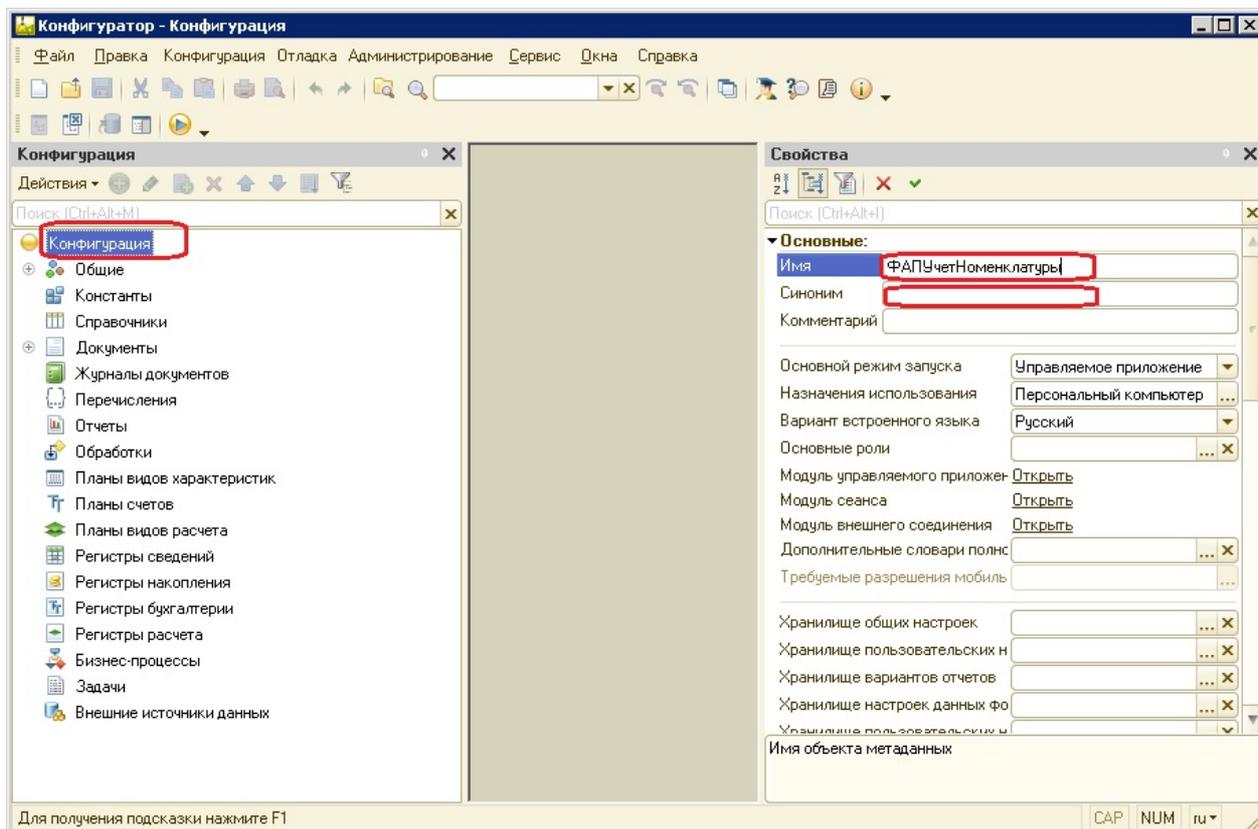


Рис. 10. Указание названия конфигурации

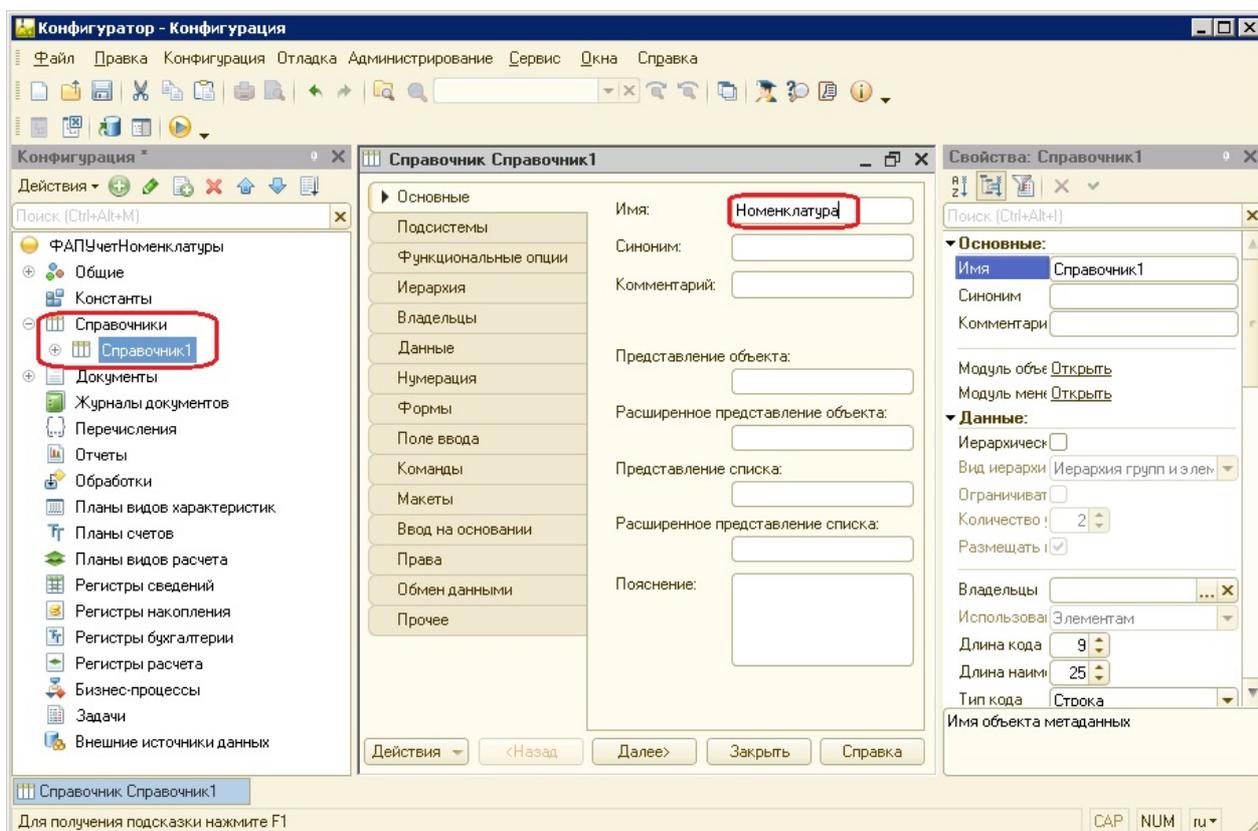


Рис. 11. Добавление справочника Номенклатура

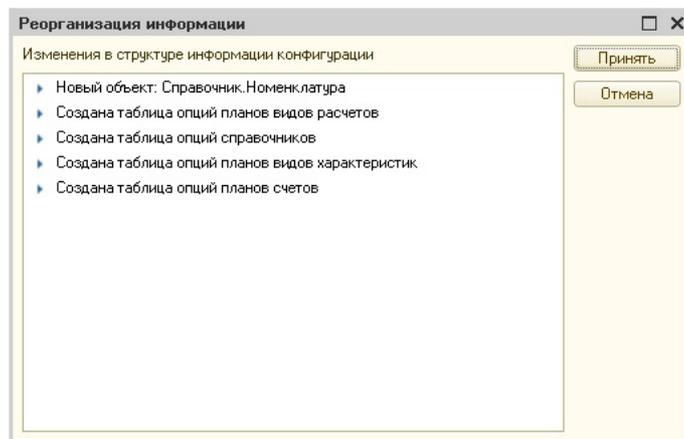


Рис. 12. Принятие изменений в базе

- *ДатаПрихода* с типом дата, который будет хранить информацию о дате поступления;
- *КоличествоУпаковок* с типом число (целое), который будет хранить информацию о количестве упаковок поступившей номенклатуры;
- *Цена* с типом число (с точностью 2 знака после запятой), который будет хранить информация о цене за одну упаковку;
- *ЕдиницВОднойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц хранится в одной упаковке (таблеток, ампул и т.д.).

Для того, чтобы была возможность группировки медикаментов с одним и тем же наименованием, но с разных поступлений, добавим в конфигурацию справочник Наименования с реквизитом ПолноеНаименование (рис. 14).

В справочник Номенклатура добавим реквизит Название с типом Справочник.Ссылка.Наименования (рис. 15). Таким образом, будет реализована связь многие-к-одному между таблицами Номенклатура и Наименования, что позволит группировать номенклатуру по наименованиям.

Основная часть данных уже описана в конфигурации, и можно заполнять базу тестовыми данными. Для этого запускаем базу в режиме 1С:Предприятия. Это можно сделать нажатием комбинации клавиш Ctrl+F5 (либо только F5, если нужно запустить в режиме отладки).

Открываем справочник Наименования и заполняем его тестовыми данными. Для этого выбираем соответствующий пункт меню и добавляем тре-

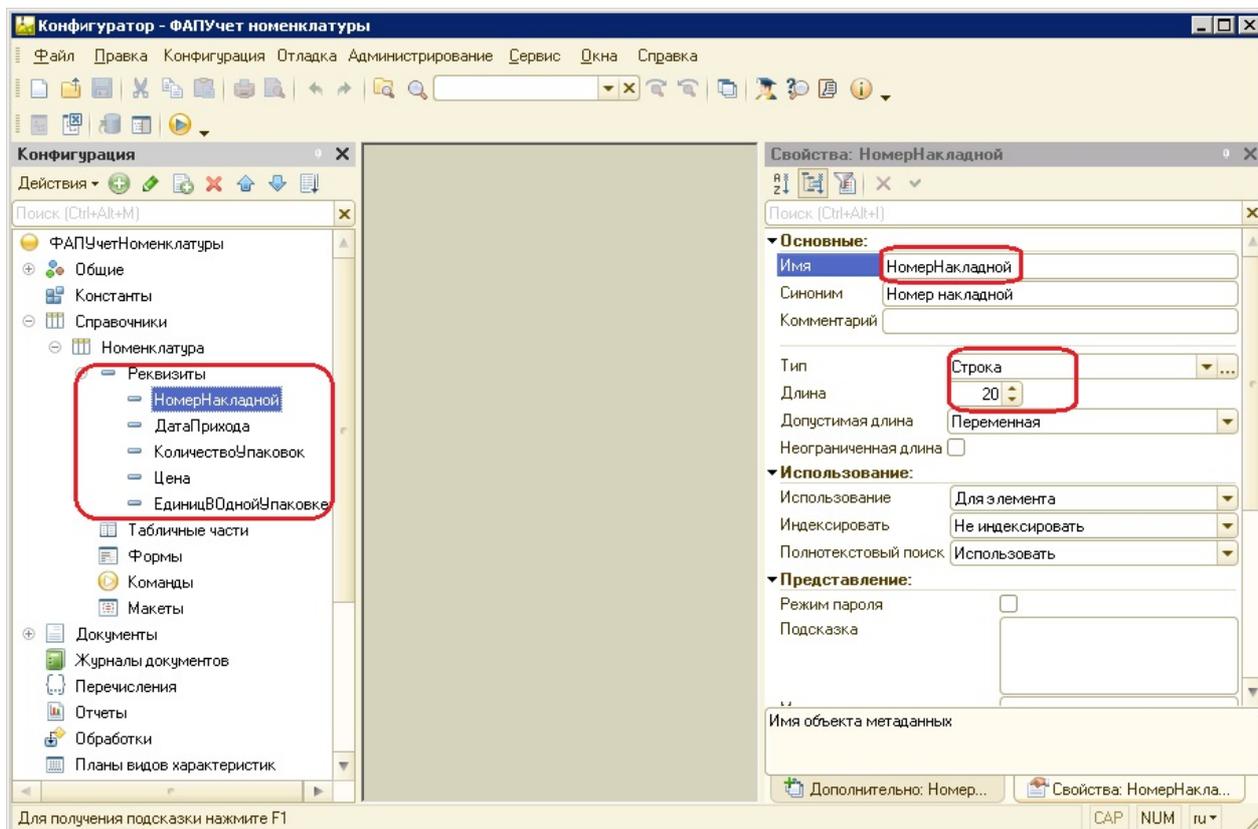


Рис. 13. Добавление реквизитов в справочник Номенклатура

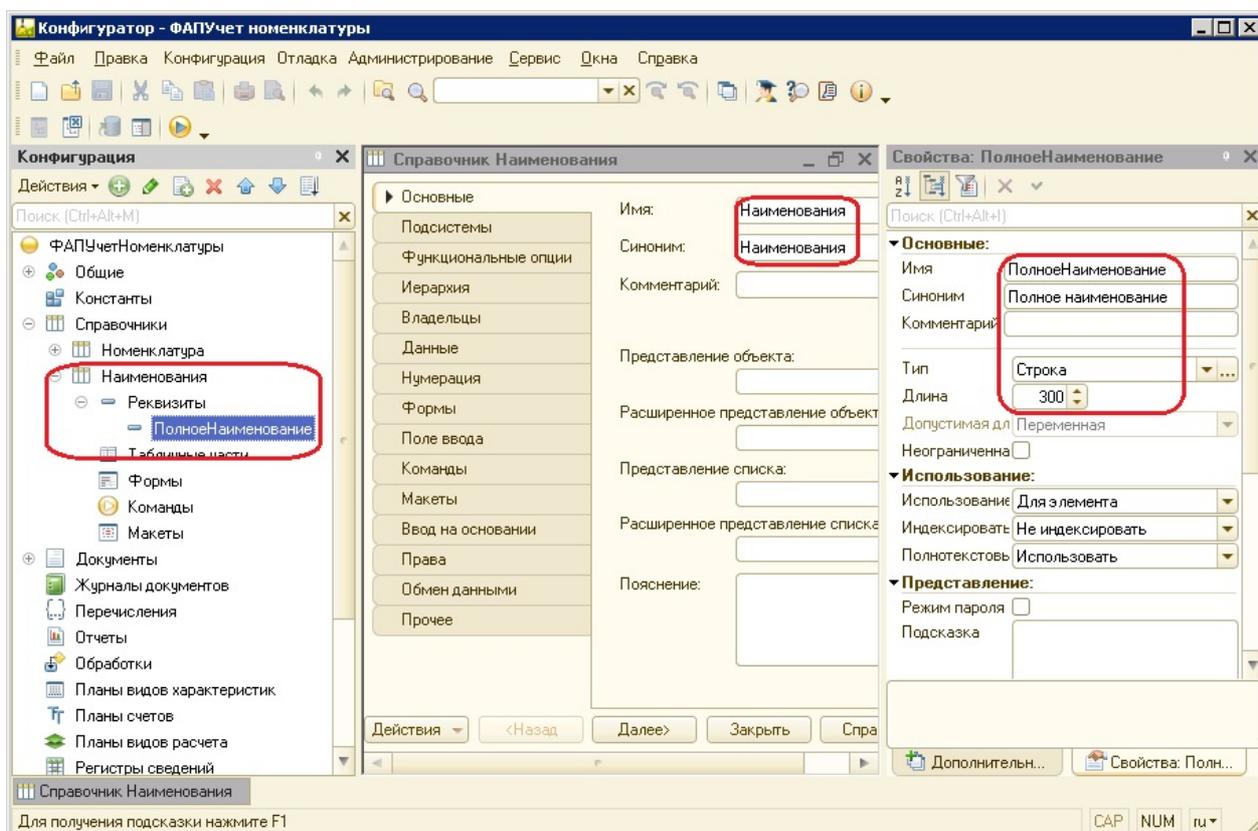


Рис. 14. Добавление справочника Наименования

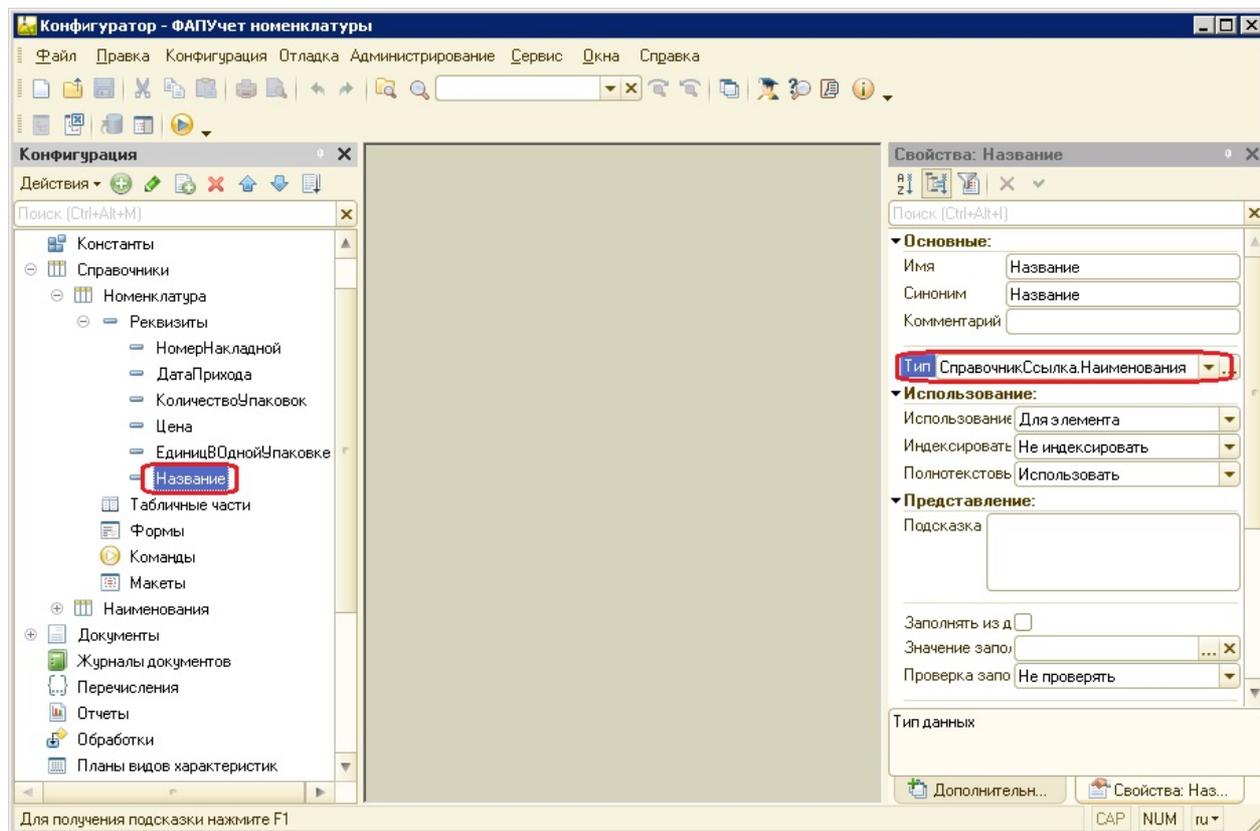


Рис. 15. Добавление реквизита Название в справочник Номенклатура

бумое количество тестовых записей (рис. 16). Обращаем внимание на то, что при добавлении элемента справочника необходимо указывать Наименование и Полное наименование. Наименование — это стандартный реквизит, и он ограничен по длине. Чтобы была возможность ввода длинных названий, добавлен реквизит ПолноеНаименование. Таким образом, стандартный реквизит Наименование будет использован в качестве краткого наименования.

Аналогично заполняется и справочник Номенклатура (рис. 17)

Таким образом, реализована возможность добавления новых данных о приходе. Для того, чтобы была возможность ввода расхода, необходимо реализовать функционал разделения каждого элемента справочника Номенклатура на отдельные части с точностью до одной единицы в каждой упаковке. Добавим в справочник Номенклатура табличную часть ИсторияРасхода, в которой будут отражаться сведения об израсходованных медикаментах за указанную дату. Для этого нажимаем правой кнопкой мыши на раздел Табличные части справочника Номенклатура, нажимаем на Добавить и в появившемся окне указываем название (рис. 18).

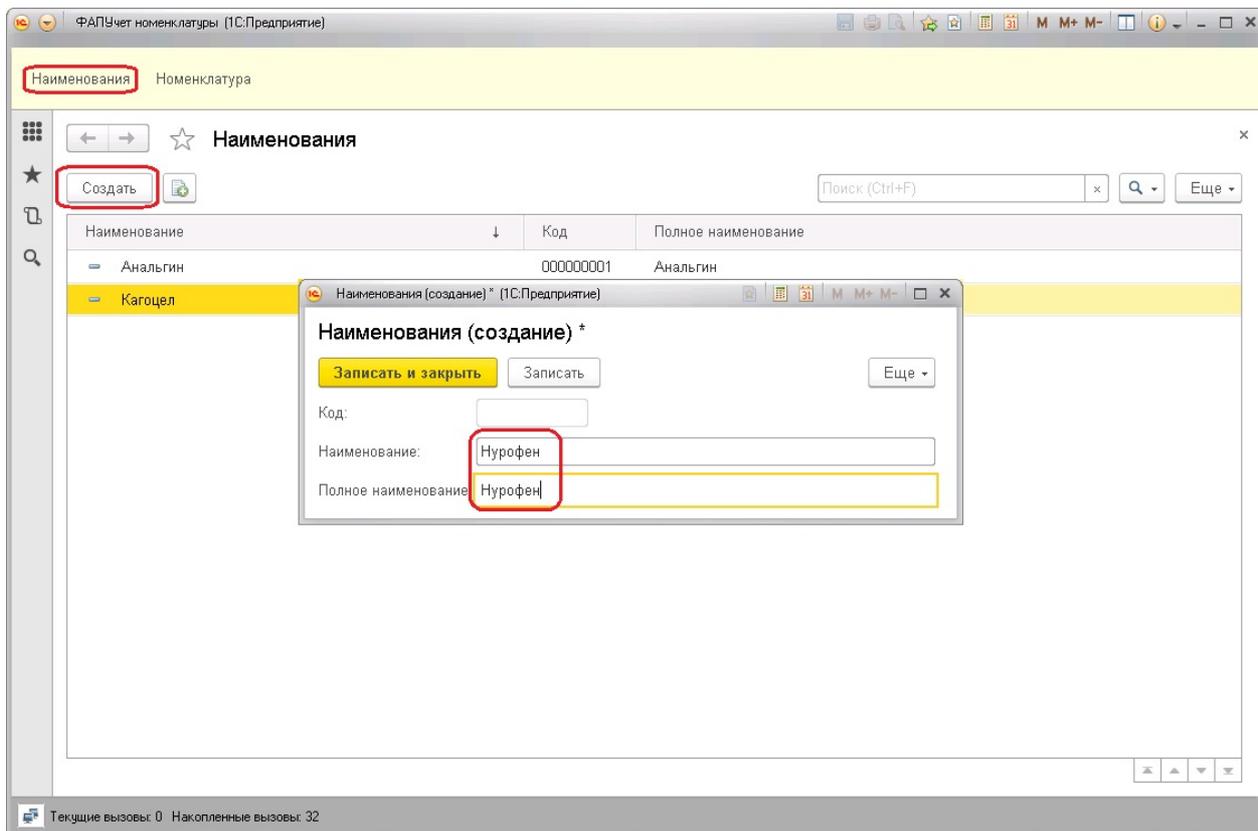


Рис. 16. Заполнение справочника Наименования

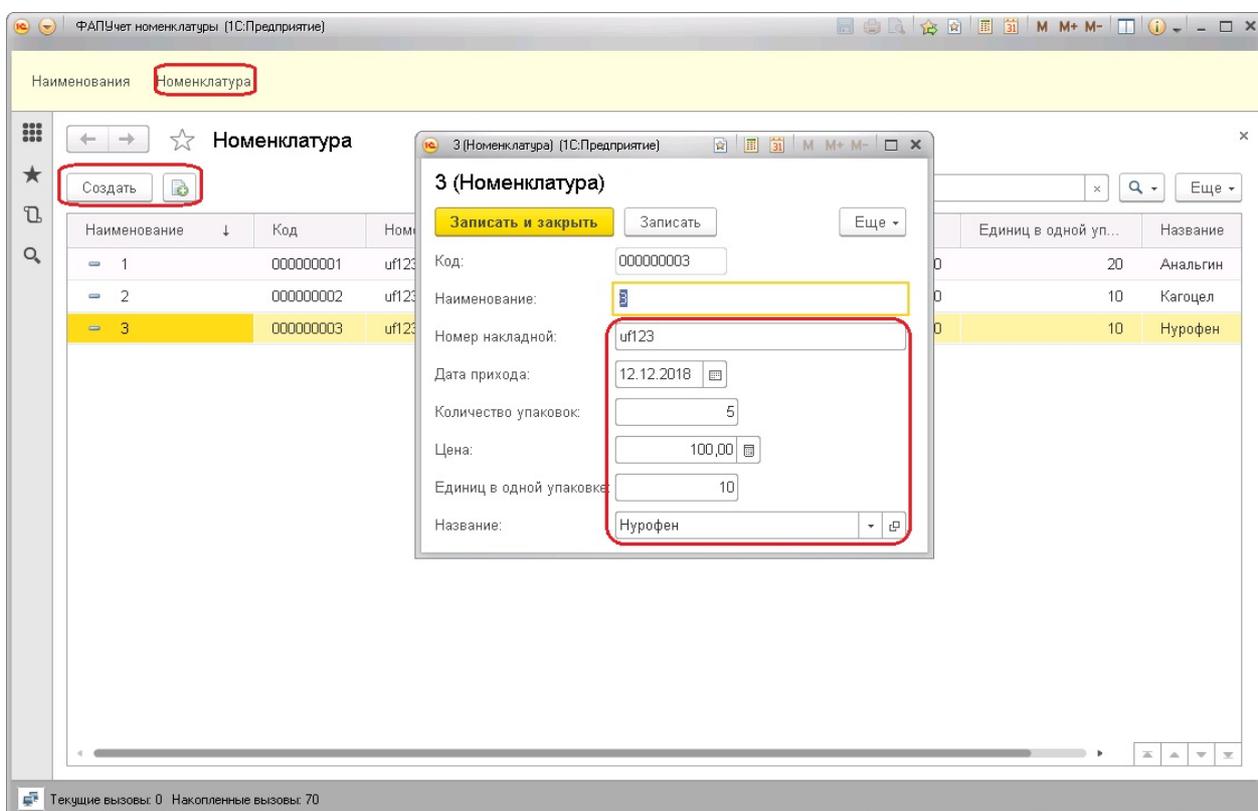


Рис. 17. Заполнение справочника Номенклатура

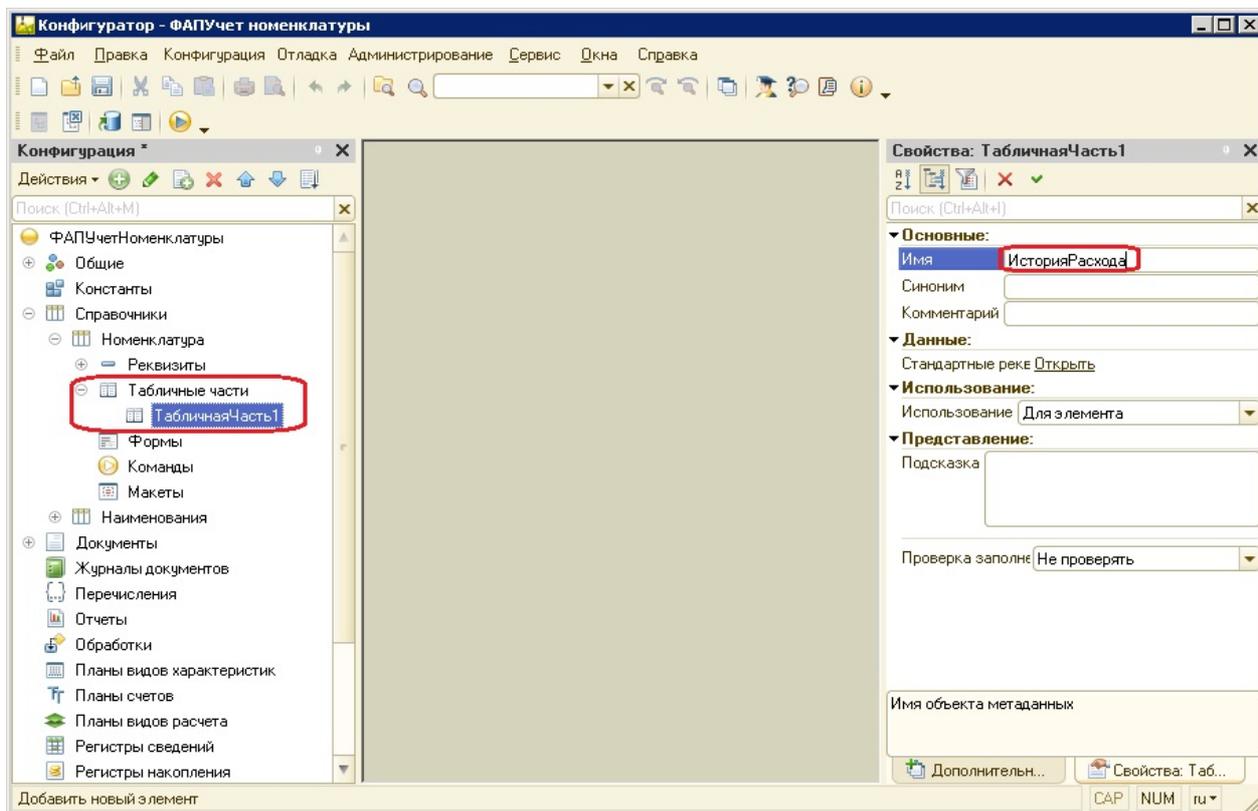


Рис. 18. Добавление табличной части ИсторияРасхода

Добавим реквизиты табличной части ИсторияРасхода. Для этого нажимаем правой кнопкой на название табличной части в дереве конфигурации и в открывшемся меню выбираем Добавить → Реквизит табличной части. В открывшемся окне указываем название и тип реквизита (рис. 19). Добавляем следующие реквизиты:

- *ДатаРасхода* с типом дата, который будет хранить информацию о дате расхода;
- *КоличествоУпаковок* с типом число (целое), который будет хранить информацию о том, сколько полных упаковок израсходовано;
- *КоличествоЕдиницВОткрытойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц использовано в открытых упаковках.

На форме элемента справочника Номенклатура теперь появилась возможность ввода данных о расходе (рис. 20).

Если вводить данные таким способом, то случайно можно ввести количество расхода больше, чем есть в наличии. Необходимо добавить какие-то

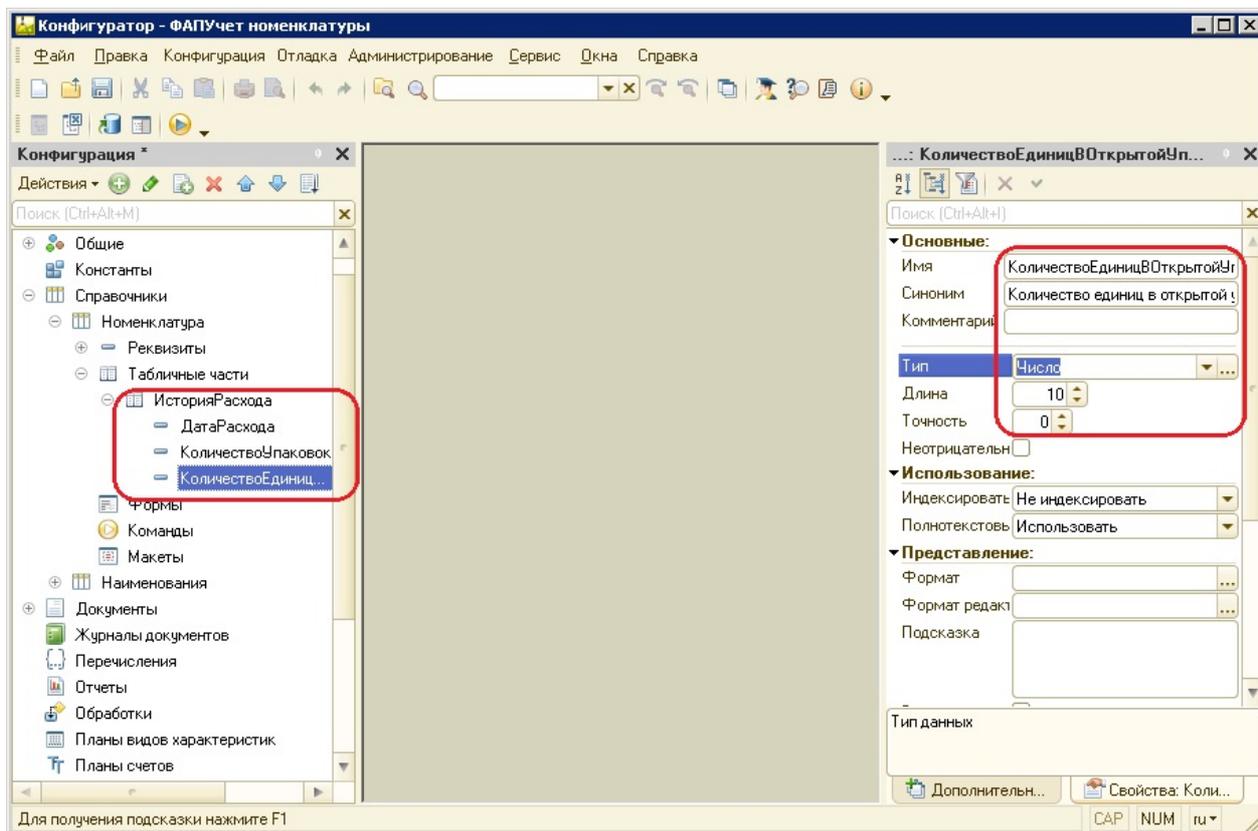


Рис. 19. Добавление реквизитов в табличную часть ИсторияРасхода

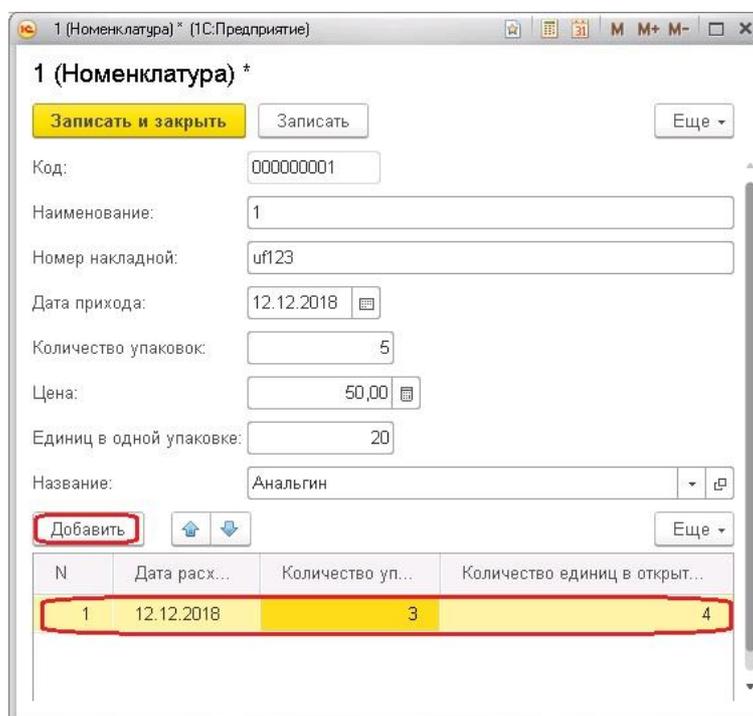


Рис. 20. Ввод данных о расходе в табличную часть ИсторияРасхода

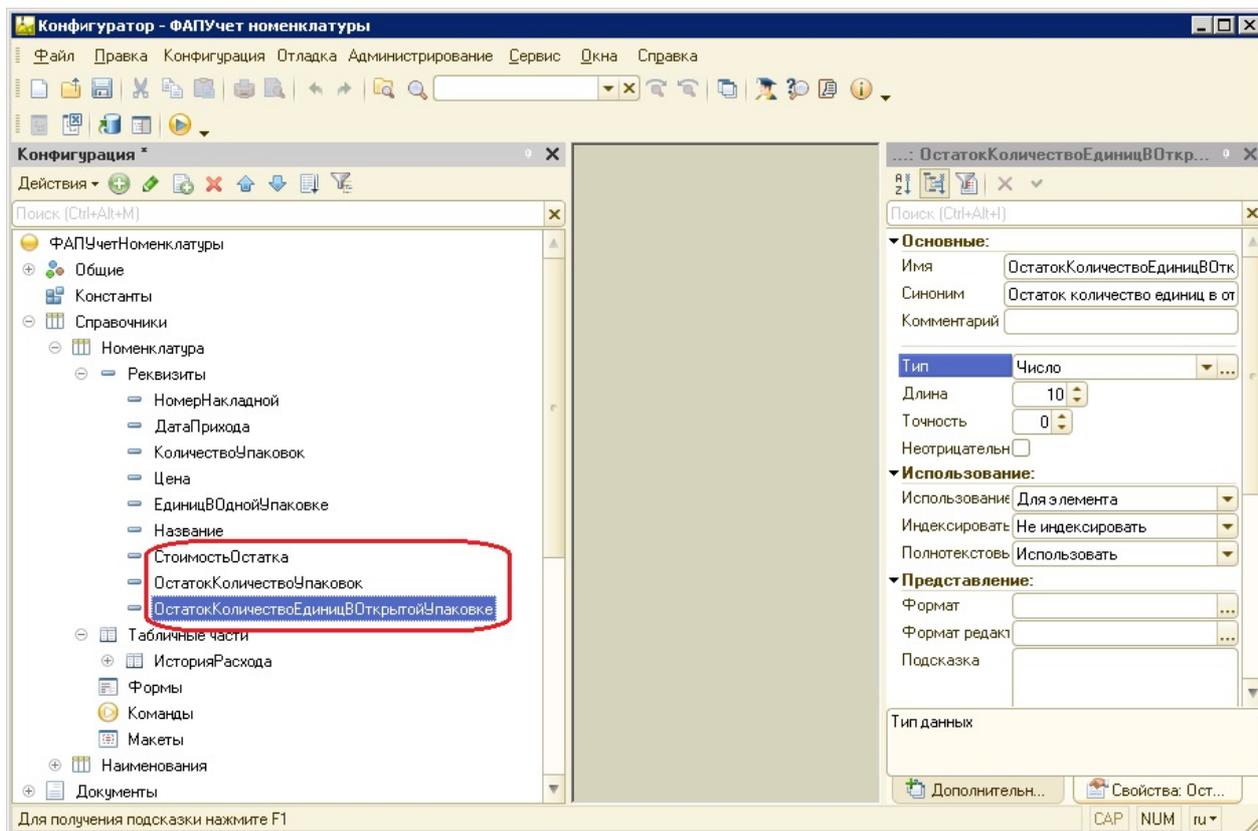


Рис. 21. Добавление в справочник Номенклатура вспомогательных реквизитов

ограничения на ввод данных. Добавим в справочник Номенклатура вспомогательные реквизиты (рис. 21):

- *СтоимостьОстатка* с типом число (с точностью 2 знака после запятой), который будет хранить информацию о стоимости оставшейся части медикаментов;
- *ОстатокКоличествоУпаковок* с типом число (целое), который будет хранить информацию о том, сколько полных упаковок осталось;
- *ОстатокКоличествоЕдиницВОткрытойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц осталось в открытых упаковках.

Значения этих реквизитов зависят от значений других реквизитов и будут использованы для упрощения вычислений. Отсюда следует, что при изменении значений одних реквизитов, должны измениться и значения этих. Добавим в модуль объекта процедуру ПередЗаписью. Очевидно, что значения зависимых реквизитов можно пересчитать в этой процедуре:

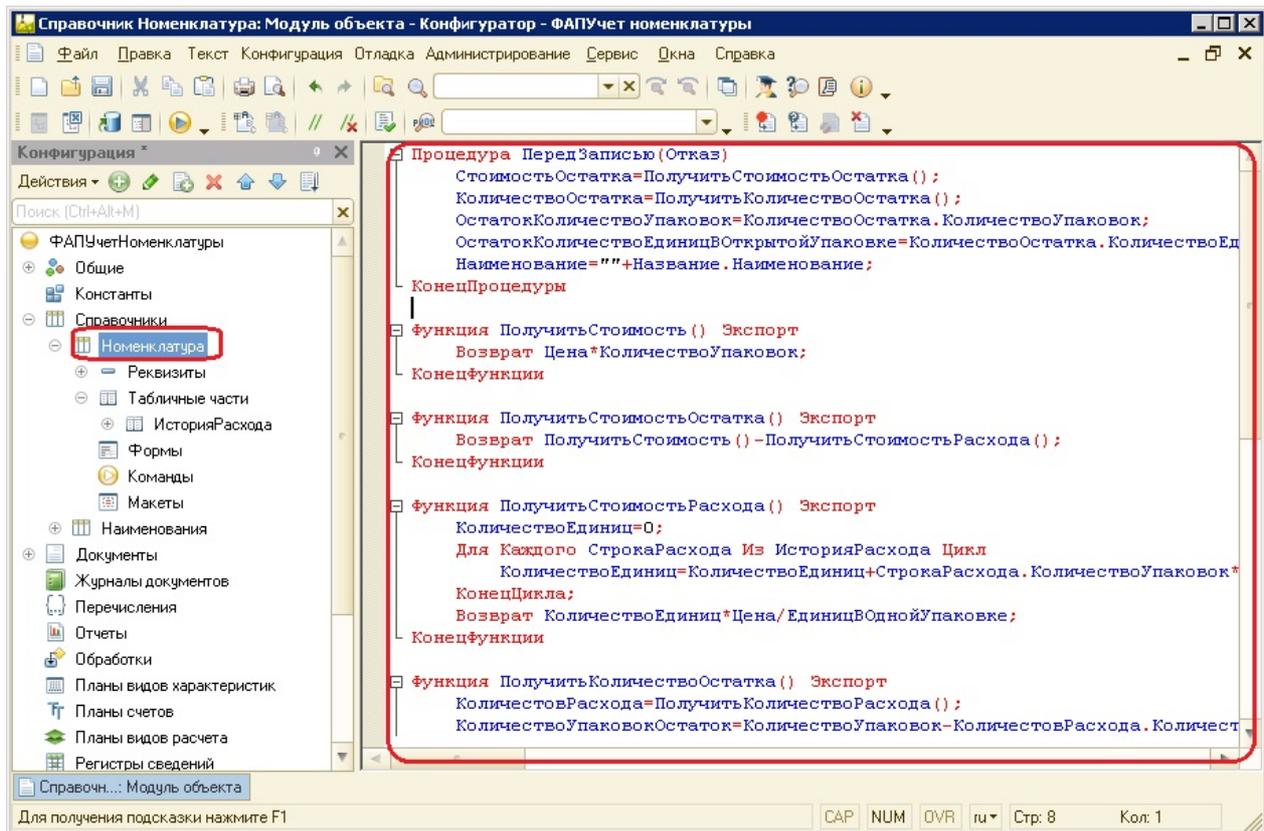


Рис. 22. Модуль объекта справочника Номенклатура

Процедура ПередЗаписью(Отказ)

СтоимостьОстатка=...;

ОстатокКоличествоУпаковок=...;

ОстатокКоличествоЕдиницВОткрытойУпаковке=...;

КонецПроцедуры

Для этого нажимаем правой кнопкой мыши на название справочника и в появившемся меню выбираем Открыть модуль объекта. В открывшемся текстовом окне вводим соответствующий код (рис. 22).

Полностью код выглядит следующим образом:

Процедура ПередЗаписью(Отказ)

СтоимостьОстатка=ПолучитьСтоимостьОстатка();

КоличествоОстатка=ПолучитьКоличествоОстатка());

ОстатокКоличествоУпаковок=КоличествоОстатка.КоличествоУпаковок;

ОстатокКоличествоЕдиницВОткрытойУпаковке=

КоличествоОстатка.КоличествоЕдиницВОткрытойУпаковке;

Наименование="'" +Название.Наименование;

КонецПроцедуры

Функция ПолучитьСтоимость() Экспорт

Возврат Цена*КоличествоУпаковок;

КонецФункции

Функция ПолучитьСтоимостьОстатка() Экспорт

Возврат ПолучитьСтоимость()-ПолучитьСтоимостьРасхода();

КонецФункции

Функция ПолучитьСтоимостьРасхода() Экспорт

КоличествоЕдиниц=0;

Для Каждого СтрокаРасхода Из ИсторияРасхода Цикл

КоличествоЕдиниц=КоличествоЕдиниц+

СтрокаРасхода.КоличествоУпаковок*ЕдиницВОднойУпаковке+

СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке;

КонецЦикла;

Возврат КоличествоЕдиниц*Цена/ЕдиницВОднойУпаковке;

КонецФункции

Функция ПолучитьКоличествоОстатка() Экспорт

КоличествоРасхода=ПолучитьКоличествоРасхода();

КоличествоУпаковокОстаток=КоличествоУпаковок-

КоличествоРасхода.КоличествоУпаковок;

Если КоличествоРасхода.КоличествоЕдиницВОткрытойУпаковке=0 Тогда

КоличествоЕдиницВОткрытойУпаковке=0;

Иначе

КоличествоЕдиницВОткрытойУпаковке=ЕдиницВОднойУпаковке-

КоличествоРасхода.КоличествоЕдиницВОткрытойУпаковке;

КонецЕсли;

Если КоличествоРасхода.КоличествоЕдиницВОткрытойУпаковке>0 Тогда

КоличествоУпаковокОстаток=КоличествоУпаковокОстаток-1;

КонецЕсли;

Возврат Новый Структура("КоличествоУпаковок,
КоличествоЕдиницВОткрытойУпаковке",

КоличествоУпаковокОстаток, КоличествоЕдиницВОткрытойУпаковке);

КонецФункции

Функция ПолучитьКоличествоРасхода() Экспорт

КоличествоЕдиниц=0;

Для Каждого СтрокаРасхода Из ИсторияРасхода Цикл

КоличествоЕдиниц=КоличествоЕдиниц+

СтрокаРасхода.КоличествоУпаковок*ЕдиницВОднойУпаковке+

СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке;

КонецЦикла;

лКоличествоУпаковок=Цел(КоличествоЕдиниц/ЕдиницВОднойУпаковке);

Возврат Новый Структура("КоличествоУпаковок,

КоличествоЕдиницВОткрытойУпаковке", лКоличествоУпаковок,

КоличествоЕдиниц-лКоличествоУпаковок*ЕдиницВОднойУпаковке);

КонецФункции

Функция ПолучитьКоличествоРасходаНаДату(НачальнаяДата, КонечнаяДата)

Экспорт

КоличествоЕдиниц=0;

Для Каждого СтрокаРасхода Из ИсторияРасхода Цикл

Если СтрокаРасхода.ДатаРасхода<НачальнаяДата Или

СтрокаРасхода.ДатаРасхода>КонечнаяДата Тогда

Продолжить;

КонецЕсли;

КоличествоЕдиниц=КоличествоЕдиниц+

СтрокаРасхода.КоличествоУпаковок*ЕдиницВОднойУпаковке+

СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке;

КонецЦикла;

лКоличествоУпаковок=Цел(КоличествоЕдиниц/ЕдиницВОднойУпаковке);

Возврат Новый Структура("КоличествоУпаковок,

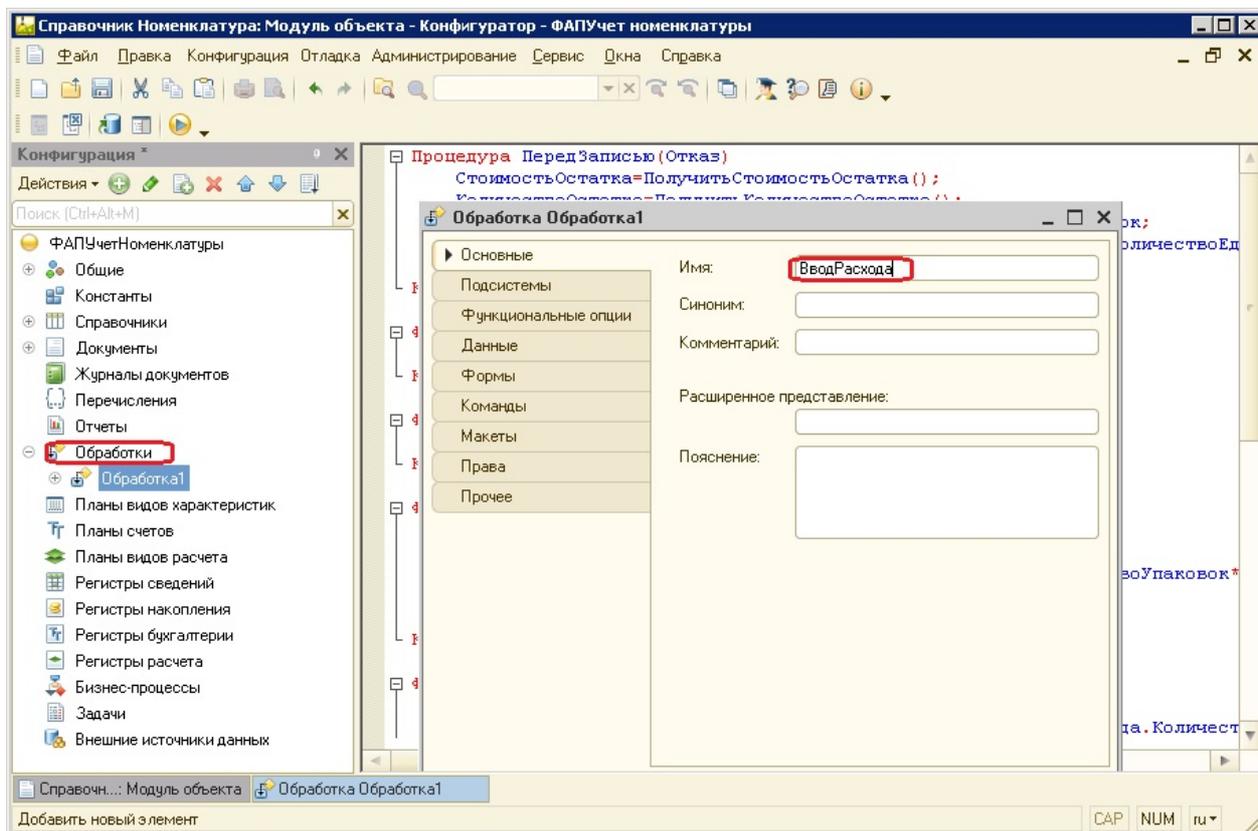


Рис. 23. Добавление обработки ВводРасхода

КоличествоЕдиницВОткрытойУпаковке”, лКоличествоУпаковок,
КоличествоЕдиниц-лКоличествоУпаковок*ЕдиницВОднойУпаковке);

КонецФункции

Добавим в конфигурацию обработку ВводРасхода, которая будет представлять удобный интерфейс для фиксирования данных о расходе. Для этого нажимаем правой кнопкой мыши на раздел Обработки в дереве конфигурации, выбираем Добавить и указываем название обработки (рис. 23).

Добавим в обработку форму. Для этого нажимаем правой кнопкой мыши на раздел Формы обработки ВводРасхода, выбираем добавить и в открывшемся окне нажимаем кнопку Готово (рис. 24).

Функционал ввода расхода должен предоставлять следующие возможности:

- просмотр информации о наличии номенклатуры, которая может быть израсходована;
- ввод расхода сразу по нескольким позициям номенклатуры;
- проверку корректности расхода (не должен превышать текущий остаток).

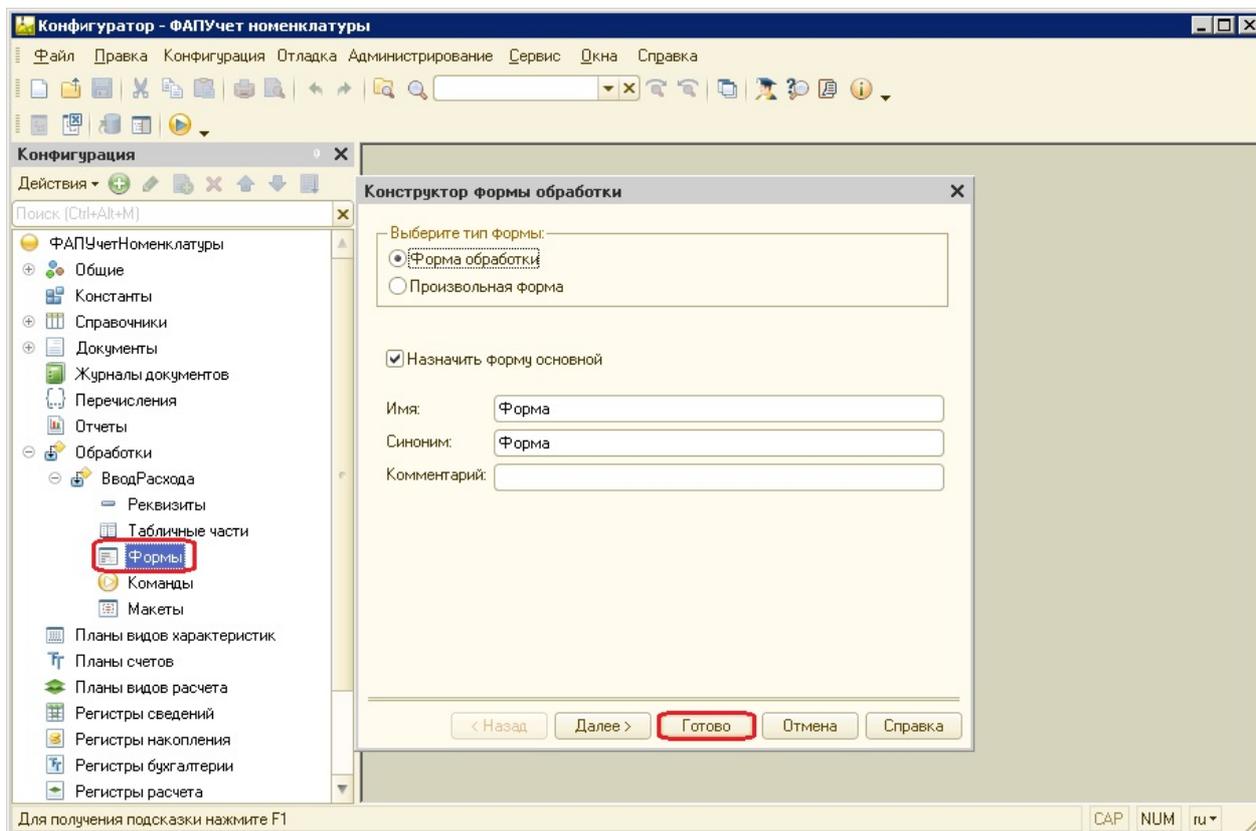


Рис. 24. Добавление формы в обработку ВводРасхода

В связи с этим предлагается создать форму из двух функциональных областей: в левой части будет отображаться информация о наличии номенклатуры, а в правой — о текущем расходе. Добавим реквизит формы Остаток, в котором будут храниться данные о оставшейся номенклатуре. Для этого нажимаем на кнопку Добавить реквизит в области реквизитов формы. В свойствах реквизита указываем имя и тип ТаблицаЗначений (рис. 25).

Добавим колонки в таблице. Для этого выделяем нужный реквизит и нажимаем на кнопку Добавить колонку реквизита (рис. 26). Добавляем следующие колонки:

- *Название* — назначение и тип совпадают с реквизитом Название у справочника Номенклатура;
- *Ссылка* — ссылка на элемент Номенклатуры;
- *НомерНакладной* — совпадает с реквизитом НомерНакладной Номенклатуры;
- *ДатаПрихода* — совпадает с реквизитом ДатаПрихода Номенклатуры;

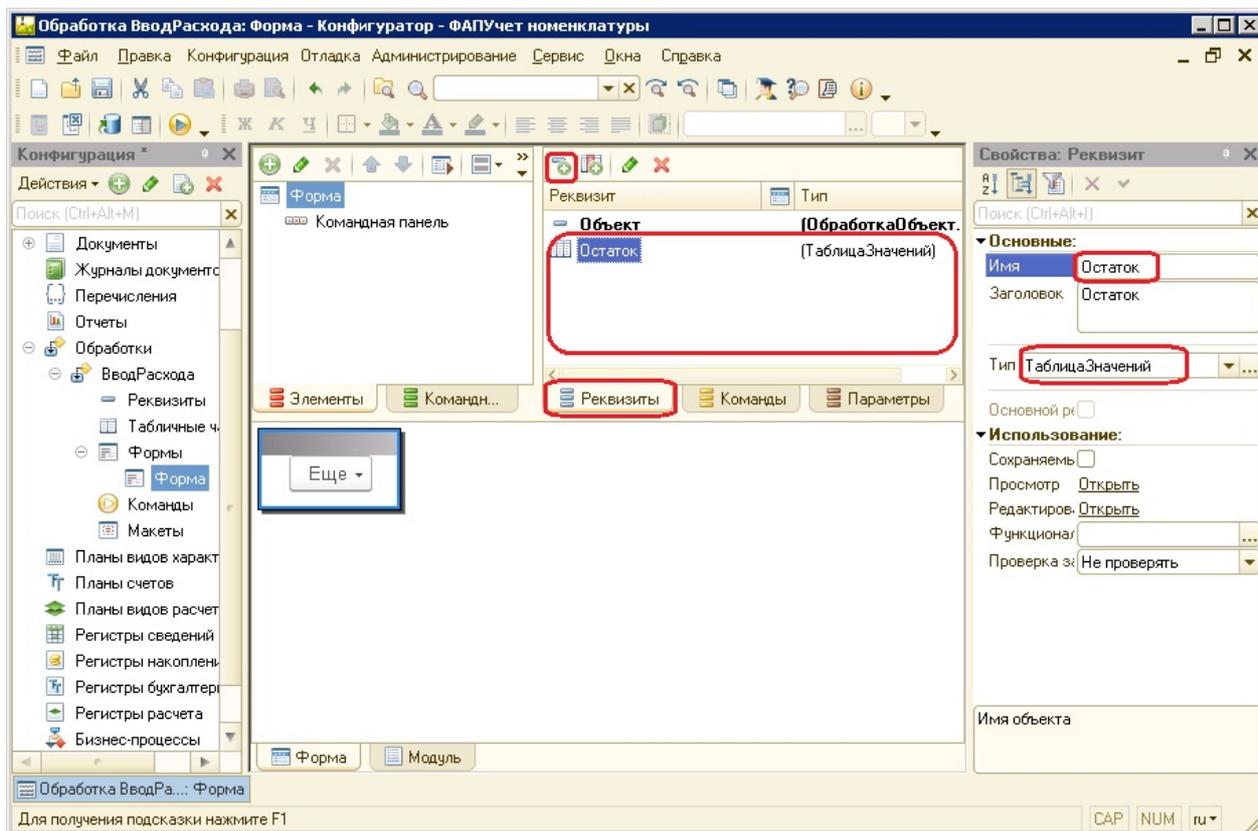


Рис. 25. Добавление реквизита Остаток на форме

- *КоличествоУпаковок* — совпадает с реквизитом ОстатокКоличествоУпаковок Номенклатуры;
- *Цена* — совпадает с реквизитом Цена Номенклатуры;
- *ЕдиницВОднойУпаковке* — совпадает с реквизитом ЕдиницВОднойУпаковке Номенклатуры;
- *КоличествоЕдиницВОткрытойУпаковке* — совпадает с реквизитом ОстатокКоличествоЕдиницВОткрытойУпаковке Номенклатуры;
- *фДобавленоВРасход* — пометка о том, что текущая позиция Номенклатуры уже включена в расход.

Аналогично добавляем реквизит Расход. Добавляем следующие колонки реквизита Расход (рис. 27):

- *Название* — совпадает с колонкой Название таблицы Остаток;
- *Ссылка* — совпадает с колонкой Ссылка таблицы Остаток;
- *НомерНакладной* — совпадает с колонкой НомерНакладной таблицы Остаток;

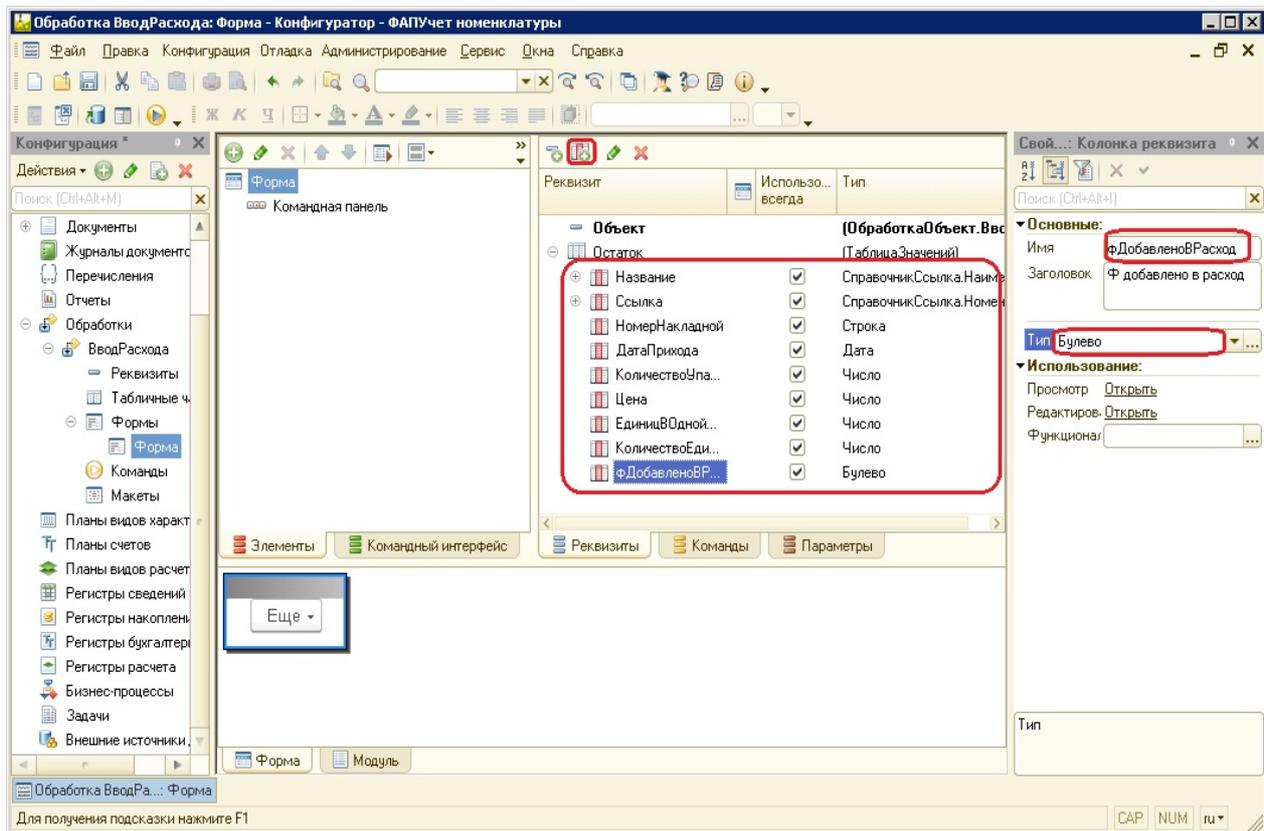


Рис. 26. Добавление колонок реквизита Остаток на форме

- *КоличествоУпаковок* — тип совпадает с типом колонки *КоличествоУпаковок* таблицы *Остаток*, но хранит количество израсходованных полных упаковок;
- *КоличествоЕдиницВОткрытойУпаковке* — тип совпадает с типом колонки *КоличествоЕдиницВОткрытойУпаковке* таблицы *Остаток*, но хранит количество израсходованных единиц в открытой упаковке;
- *ДатаРасхода* — хранит информацию о дате расхода текущей позиции.

Далее необходимо сделать так, чтобы данные визуально отображались на форме. Для этого перенесем реквизиты непосредственно на форму. Для этого по очереди перетаскиваем оба реквизита (рис. 28).

Установим горизонтальную группировку элементов формы, чтобы остаток отображался слева, а расход — справа (рис. 29).

Далее нам требуется, чтобы при открытии формы таблица *Остаток* уже была заполнена соответствующими значениями. Для этого, указываем в свойствах формы событие *ПриСозданииНаСервере*. Это можно сделать нажатием на значок лупы справа от соответствующего поля свойства (рис. 30).

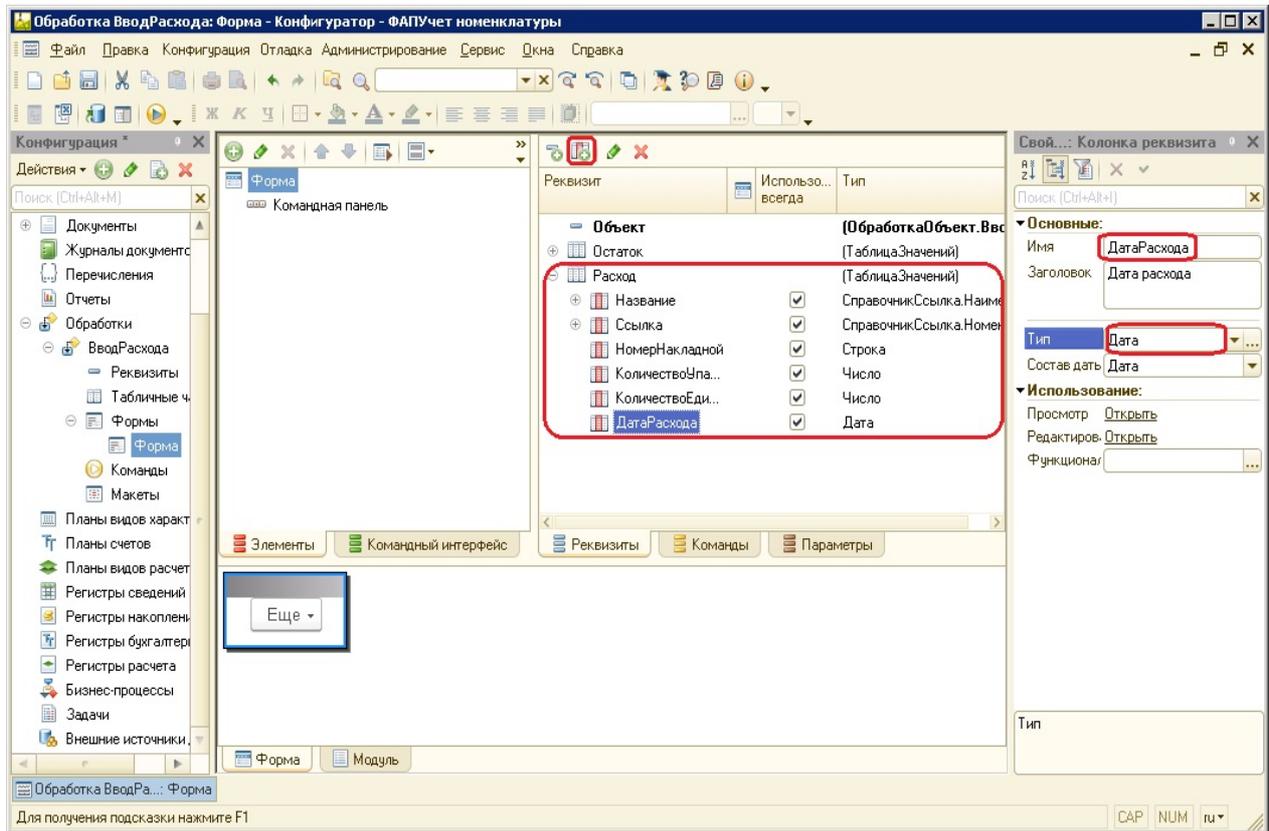


Рис. 27. Добавление реквизита Расход и колонок Расхода

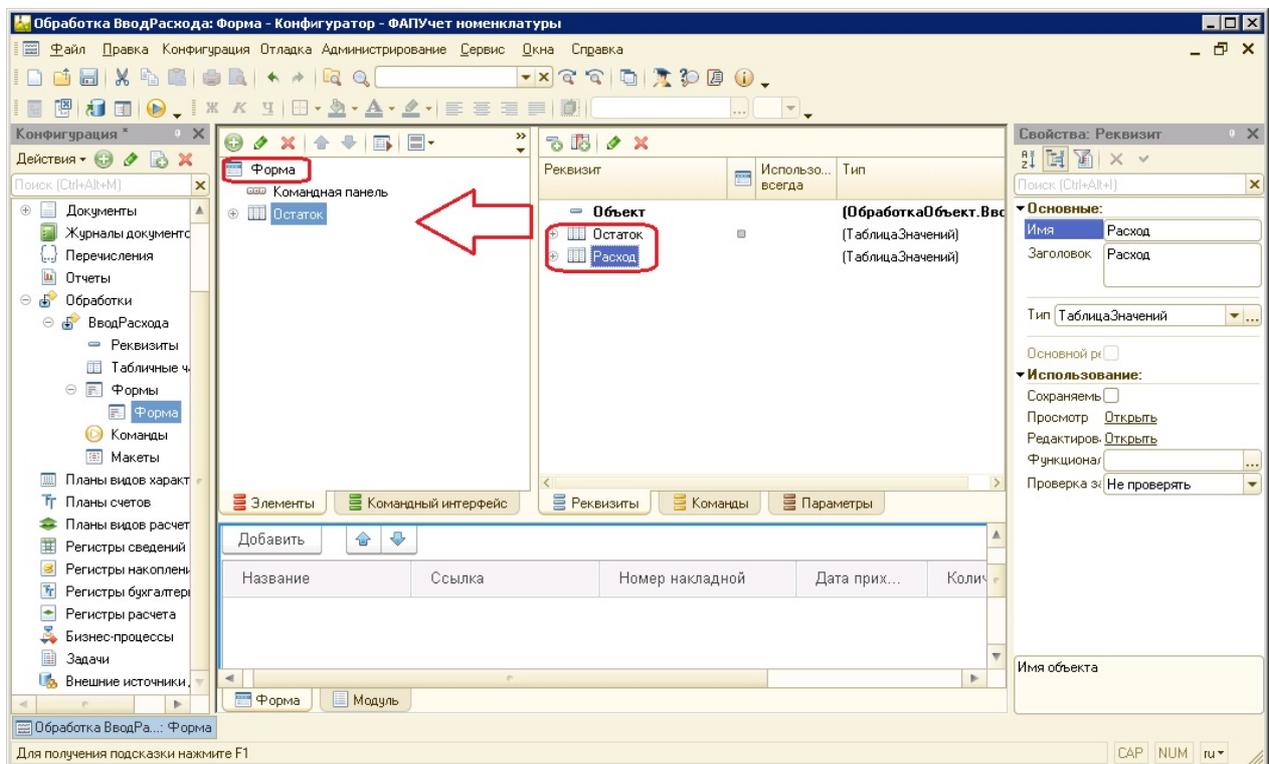


Рис. 28. Перенос реквизитов непосредственно на форму

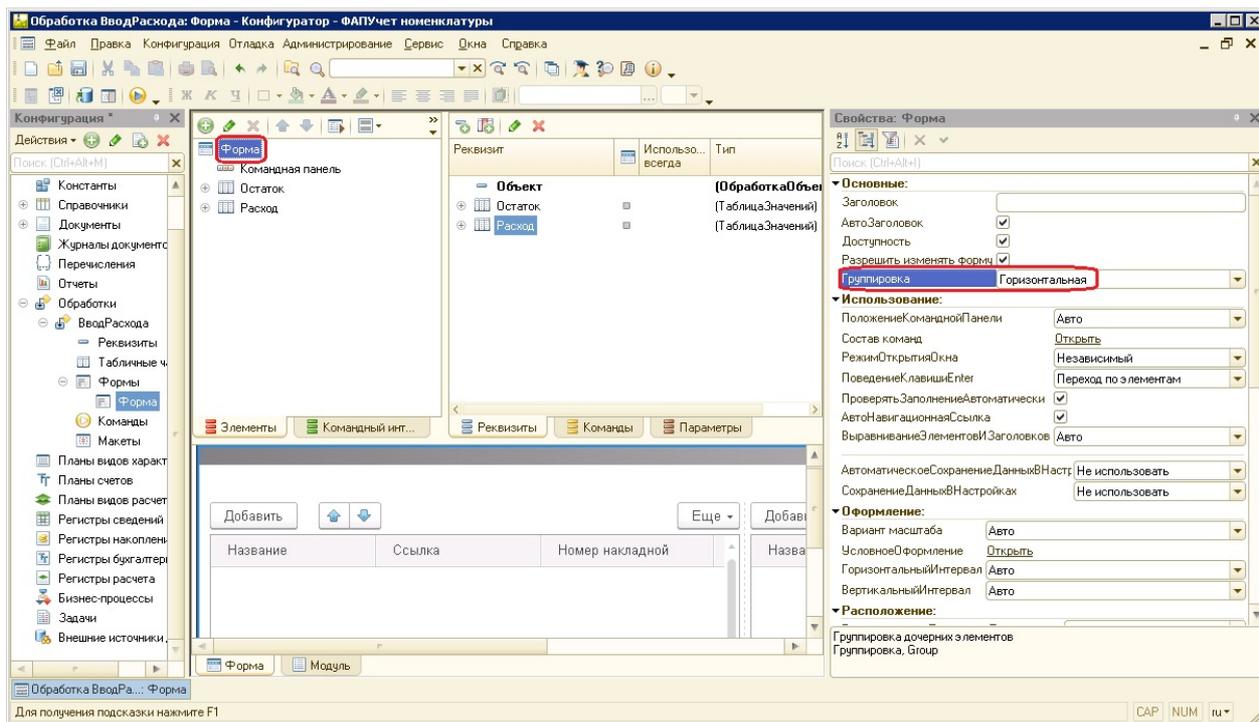


Рис. 29. Изменение способа группировки элементов на форме

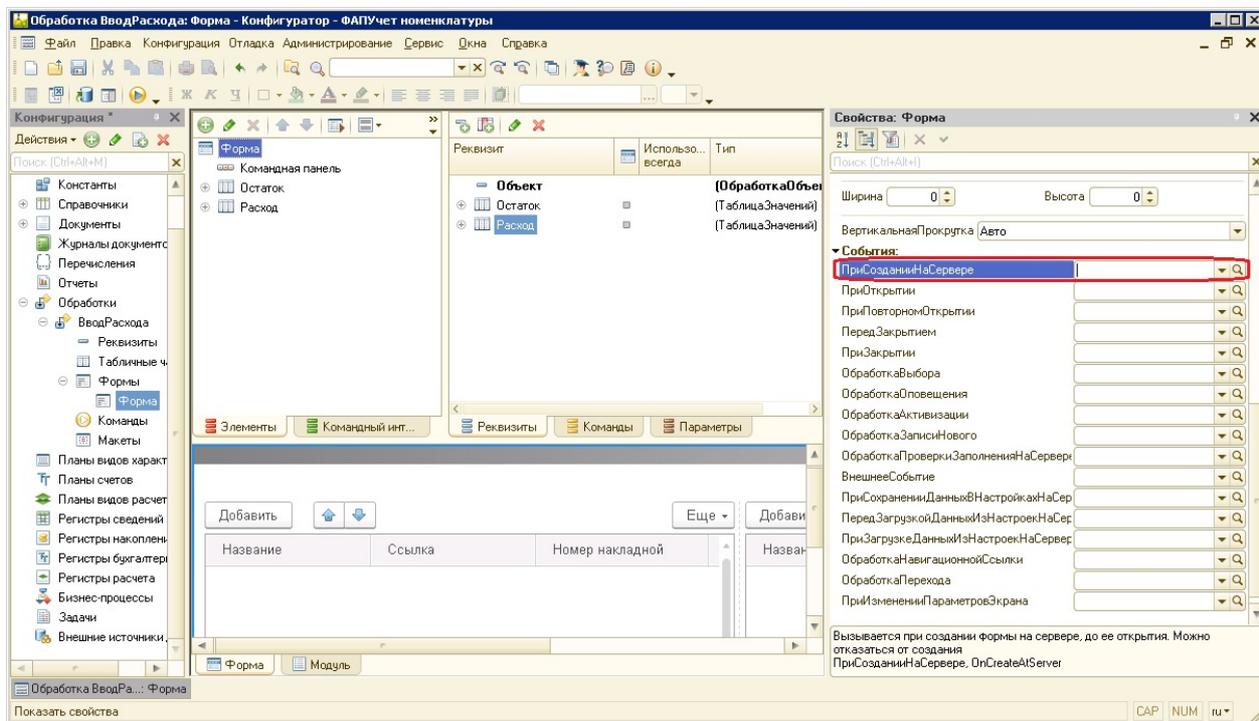


Рис. 30. Создание процедуры ПриСозданииНаСервере

Процедура ПриСозданииНаСервере будет сформирована автоматически. Остается только вписать туда нужный код (рис. 31):

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Запрос=Новый Запрос;

Запрос.Текст="ВЫБРАТЬ

| Номенклатура.Ссылка,

| Номенклатура.Название,

| Номенклатура.НомерНакладной,

| Номенклатура.ДатаПрихода,

| Номенклатура.ОстатокКоличествоУпаковок КАК КоличествоУпаковок,

| Номенклатура.Цена,

| Номенклатура.ЕдиницВОднойУпаковке,

| Номенклатура.ОстатокКоличествоЕдиницВОткрытойУпаковке

| КАК КоличествоЕдиницВОткрытойУпаковке,

| ЛОЖЬ КАК фДобавленоВРасход

|ИЗ

| Справочник.Номенклатура КАК Номенклатура

|ГДЕ

| Номенклатура.СтоимостьОстатка > 0";

Остаток.Загрузить(Запрос.Выполнить().Выгрузить());

КонецПроцедуры

Здесь выполняется запрос к справочнику Номенклатура, и полученные данные загружаются в таблицу Остаток. Фрагмент «ЛОЖЬ КАК фДобавленоВРасход» означает, что текущие данные пока не учтены в расходе. Условие «Номенклатура.СтоимостьОстатка > 0» говорит о том, что будут выбраны только те записи, у которых еще есть остаток, т.е. расход возможен.

Можно запустить обработку и убедиться, что данные действительно считываются из справочника Номенклатура и отображаются на форме (рис. 32).

Добавим команду (ДобавитьВРасходТекущую), которая будет переносить выделенную строку таблицы Остаток в таблицу Расход. Для этого переходим на вкладку Команды, нажимаем кнопку Добавить и заполняем необходимые свойства (рис. 33). Код команды выглядит следующим образом:

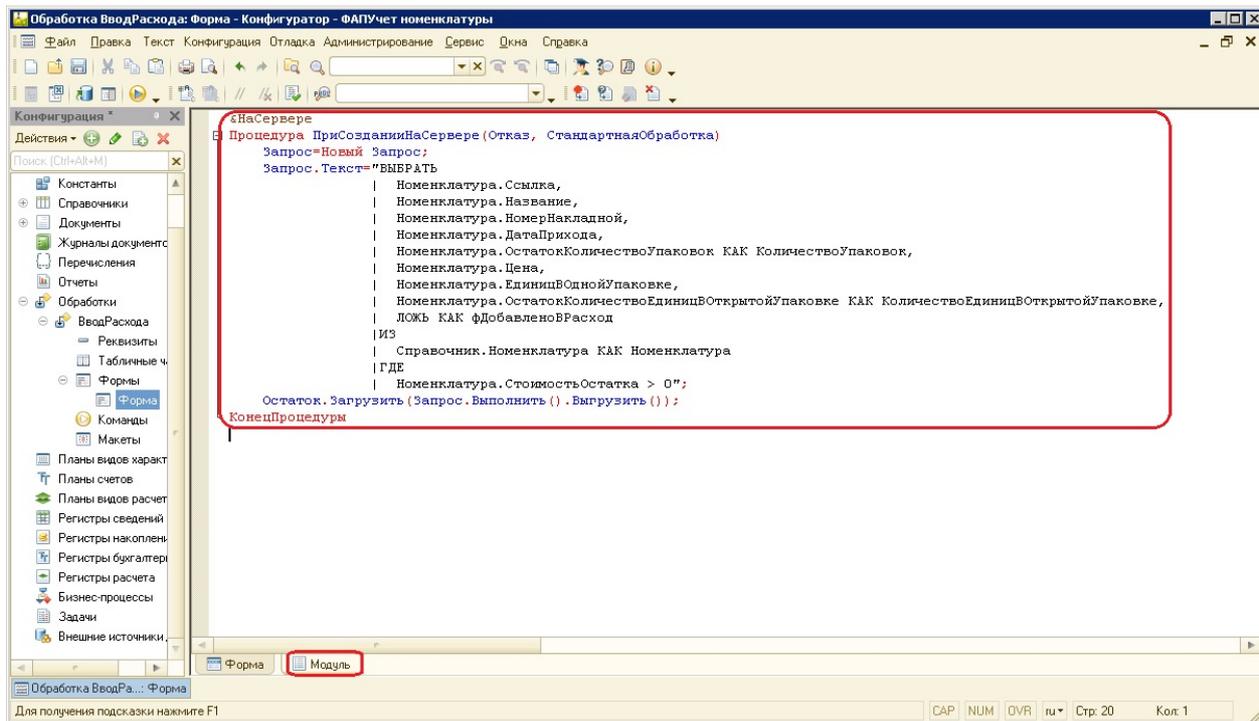


Рис. 31. Создание процедуры ПриСозданииНаСервере

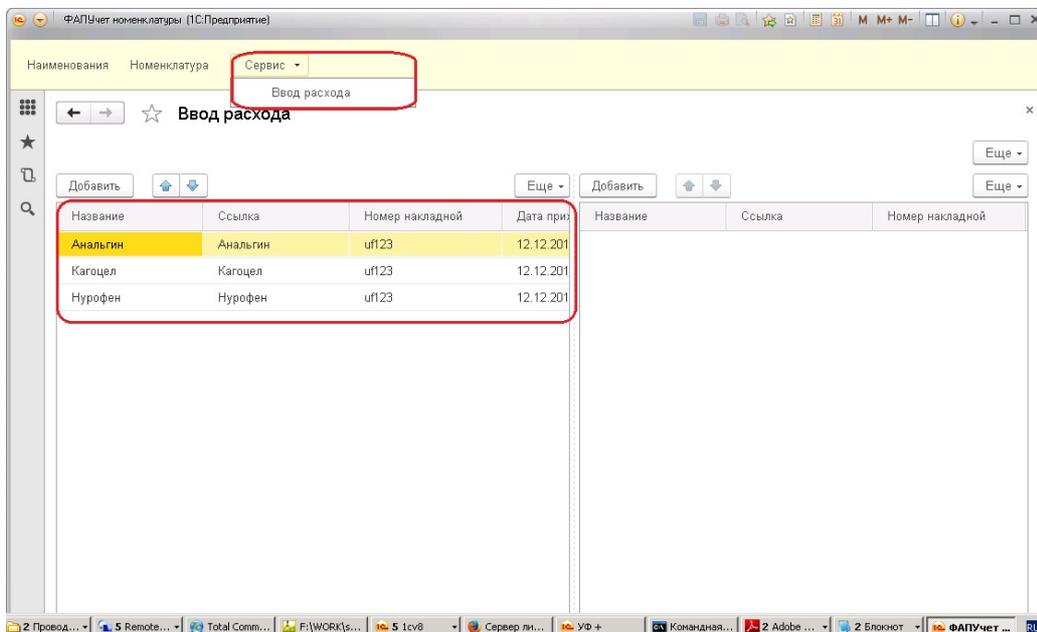


Рис. 32. Первый запуск обработки

&НаКлиенте

Процедура ДобавитьВРасходТекущую(Команда)

Если Элементы.Остаток.ТекущиеДанные=Неопределено Тогда

Возврат;

КонецЕсли;

ДобавитьВРасходТекущуюНаСервере(

Остаток.Индекс(Элементы.Остаток.ТекущиеДанные));

КонецПроцедуры

&НаСервере

Процедура ДобавитьВРасходТекущуюНаСервере(ИндексЭлемента)

текЭлемент=Остаток.Получить(ИндексЭлемента);

Если текЭлемент.фДобавленоВРасход Тогда

Сообщить("Уже добавлено: "+текЭлемент.Название);

Возврат;

КонецЕсли;

ДобавитьСтрокуРасхода(Остаток.Получить(ИндексЭлемента).Ссылка);

текЭлемент.фДобавленоВРасход=Истина;

КонецПроцедуры

&НаСервере

Функция ДобавитьСтрокуРасхода(СсылкаНаНоменклатуру)

НоваяСтрока=Расход.Добавить();

НоваяСтрока.Ссылка=СсылкаНаНоменклатуру;

НоваяСтрока.Название=СсылкаНаНоменклатуру.Название;

НоваяСтрока.НомерНакладной=СсылкаНаНоменклатуру.НомерНакладной;

НоваяСтрока.ДатаРасхода=ТекущаяДата();

НоваяСтрока.КоличествоУпаковок=

СсылкаНаНоменклатуру.ОстатокКоличествоУпаковок;

НоваяСтрока.КоличествоЕдиницВОткрытойУпаковке=

СсылкаНаНоменклатуру.ОстатокКоличествоЕдиницВОткрытойУпаковке;

КонецФункции

Переносим команду на форму (рис. 34).

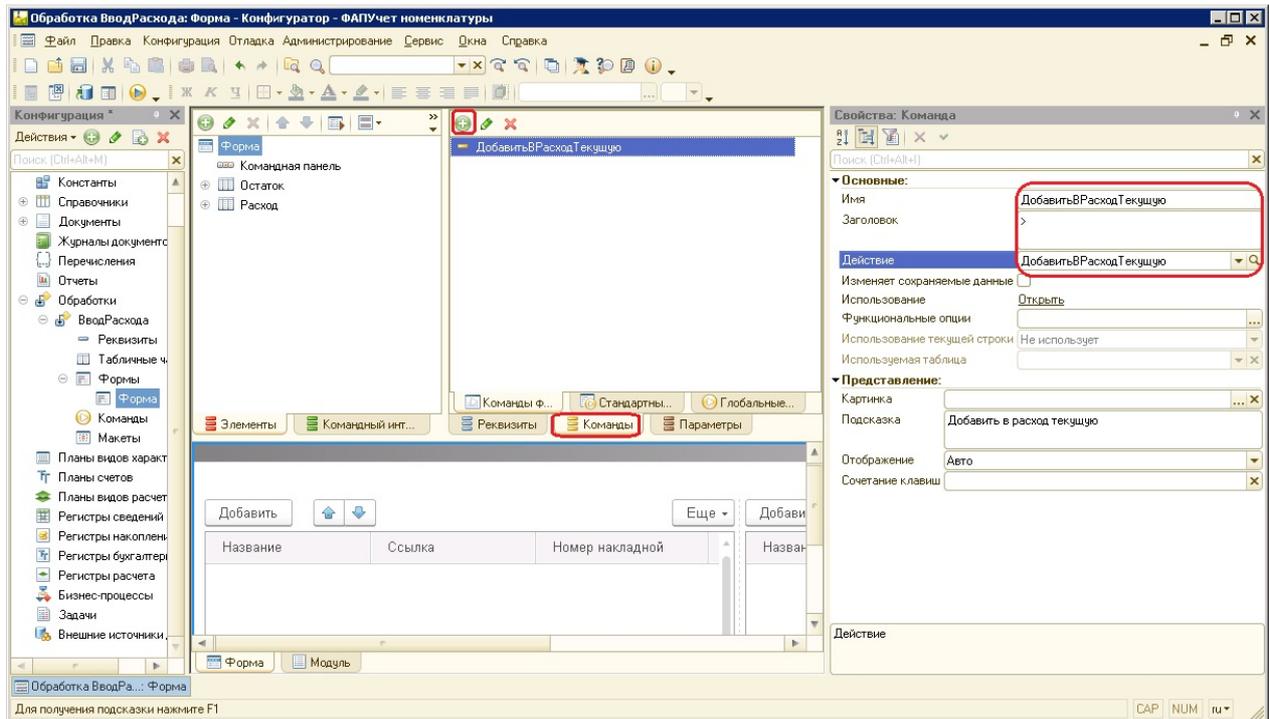


Рис. 33. Добавление команды

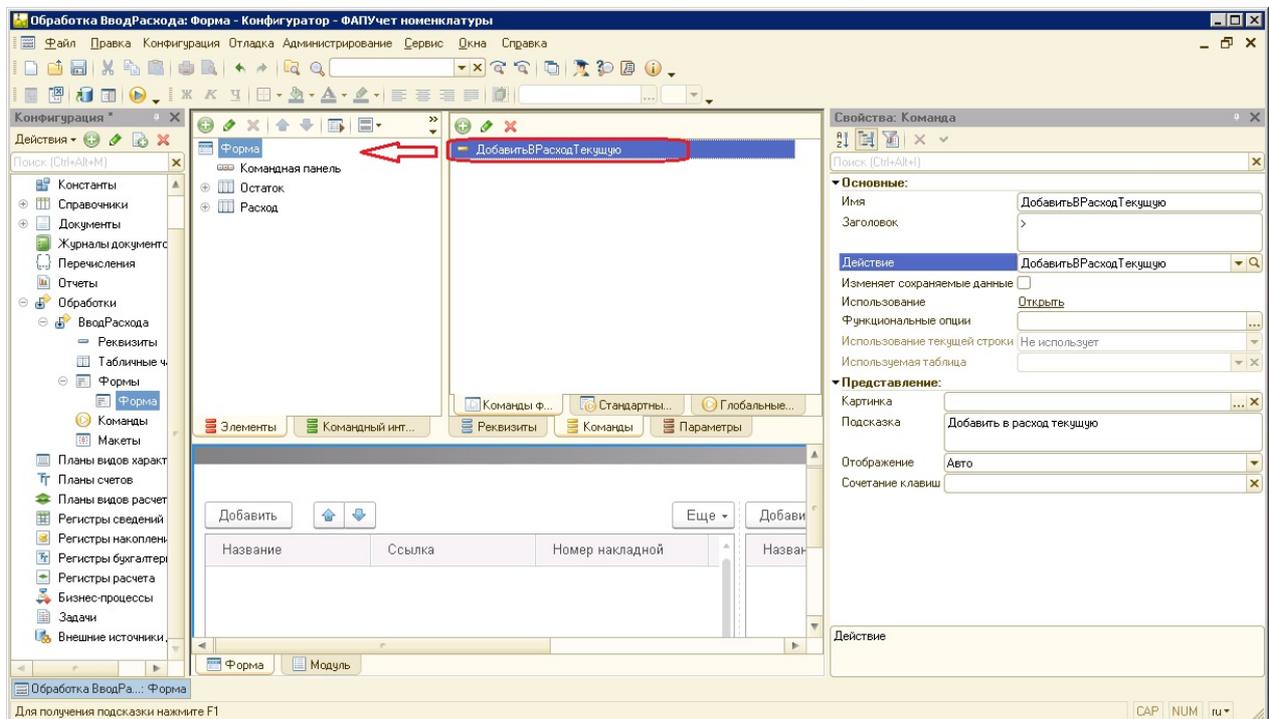


Рис. 34. Перенос команды на форму

Для удобства потребуются еще команды удаления текущей строки расхода (УдалитьИзРасходаТекущую), перенос сразу всех остатков в расход (ДобавитьВРасходВсе) и удаление всех строк расхода (УдалитьИзРасходаВсе). Код команд выглядит следующим образом:

&НаКлиенте

Процедура УдалитьИзРасходаТекущую(Команда)

Если Элементы.Расход.ТекущиеДанные=Неопределено Тогда

 Возврат;

КонецЕсли;

УдалитьИзРасходаТекущуюНаСервере(

 Расход.Индекс(Элементы.Расход.ТекущиеДанные));

КонецПроцедуры

&НаСервере

Процедура УдалитьИзРасходаТекущуюНаСервере(ИндексЭлемента)

 СтрокаВОстатка=

 НайтиВОстатках(Расход.Получить(ИндексЭлемента).Ссылка);

 Если Не СтрокаВОстатка=Неопределено Тогда

 СтрокаВОстатка.фДобавленоВРасход=Ложь;

 КонецЕсли;

 Расход.Удалить(ИндексЭлемента);

КонецПроцедуры

&НаСервере

Функция НайтиВОстатках(СсылкаНаНоменклатуру)

 СтрокиОстатков=Остаток.НайтиСтроки(

 Новый Структура("Ссылка", СсылкаНаНоменклатуру));

 Если СтрокиОстатков.Количество()=0 Тогда

 Возврат Неопределено;

 КонецЕсли;

 Возврат СтрокиОстатков[0];

КонецФункции

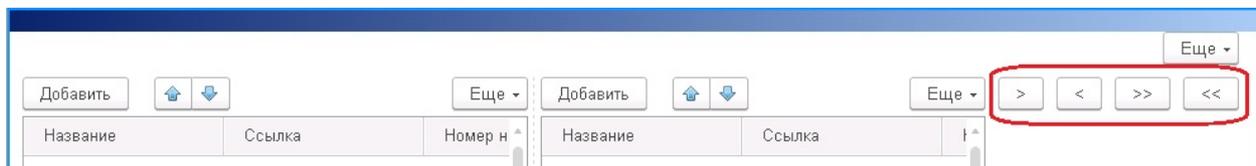


Рис. 35. Добавленные команды

&НаКлиенте

Процедура **ДобавитьВРасходВсе(Команда)**

ДобавитьВРасходВсеНаСервере();

КонецПроцедуры

&НаСервере

Процедура **ДобавитьВРасходВсеНаСервере()**

Для $\mathbf{\tau=0}$ По Остаток.Количество()-1 Цикл

ДобавитьВРасходТекущуюНаСервере($\mathbf{\tau}$);

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура **УдалитьИзРасходаВсе(Команда)**

УдалитьИзРасходаВсеНаСервере();

КонецПроцедуры

&НаСервере

Процедура **УдалитьИзРасходаВсеНаСервере()**

Пока Расход.Количество() $\mathbf{>0}$ Цикл

УдалитьИзРасходаТекущуюНаСервере($\mathbf{0}$);

КонецЦикла;

КонецПроцедуры

Команды добавляются с правой стороны формы (рис. 35).

Переместим их в более удобное для использования место. Добавим Группу на форме между двумя таблицами. Для этого выделяем форму и нажимаем кнопку Добавить. В открывшемся окне выбираем Группа - Обычная группа

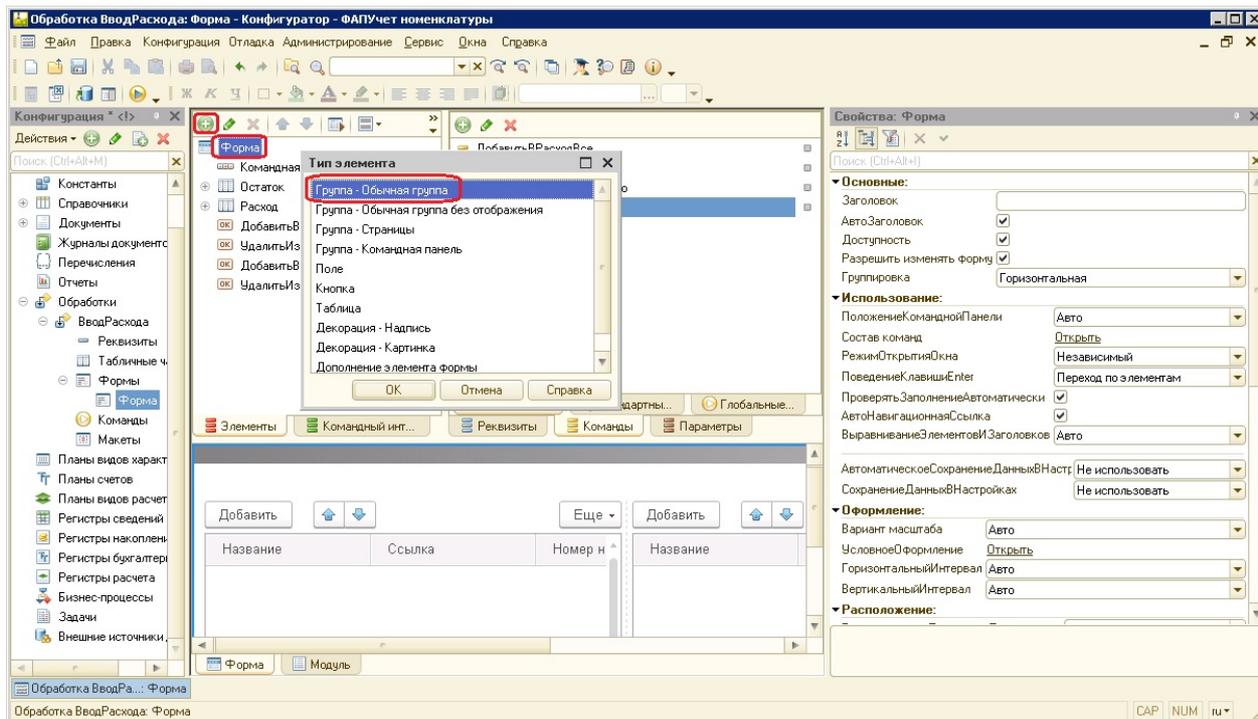


Рис. 36. Добавление Группы

и нажимаем ОК (рис. 36).

Перетаскиваем группу между таблицами и добавленные команды в эту группу (рис. 37).

Добавим основную команду (ЗаписатьРасход), которая будет записывать данные о расходе в справочник Номенклатура. И добавим эту команду в основную командную панель формы (рис. 38). Код команды будет выглядеть следующим образом:

&НаКлиенте

Процедура ЗаписатьРасход(Команда)

Если Не ПроверитьКорректностьРасхода() Тогда Возврат; КонецЕсли;

ЗаписатьРасходНаСервере();

Сообщить("Данные о расходе успешно введены");

КонецПроцедуры

&НаСервере

Процедура ЗаписатьРасходНаСервере()

НачатьТранзакцию();

Для Каждого СтрокаРасхода Из Расход Цикл

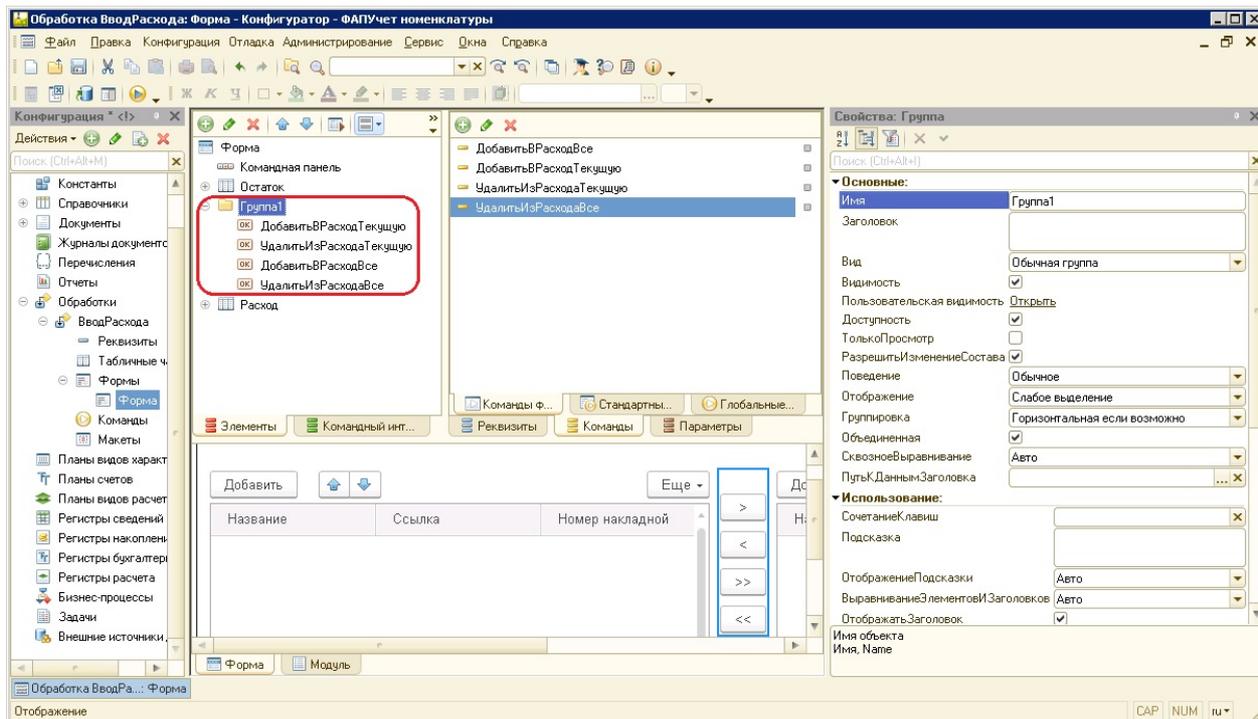


Рис. 37. Группировка команд

ОбъектНоменклатура=СтрокаРасхода.Ссылка.ПолучитьОбъект());

НовыйРасход=ОбъектНоменклатура.ИсторияРасхода.Добавить());

НовыйРасход.ДатаРасхода=СтрокаРасхода.ДатаРасхода;

НовыйРасход.КоличествоУпаковок=

СтрокаРасхода.КоличествоУпаковок;

НовыйРасход.КоличествоЕдиницВОткрытойУпаковке=

СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке;

ОбъектНоменклатура.Записать());

КонецЦикла;

ЗафиксироватьТранзакцию());

КонецПроцедуры

&НаСервере

Функция ПроверитьКорректностьРасхода()

Для Каждого СтрокаРасхода Из Расход Цикл

Если СтрокаРасхода.КоличествоУпаковок>

СтрокаРасхода.Ссылка.ОстатокКоличествоУпаковок Или

(СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке>

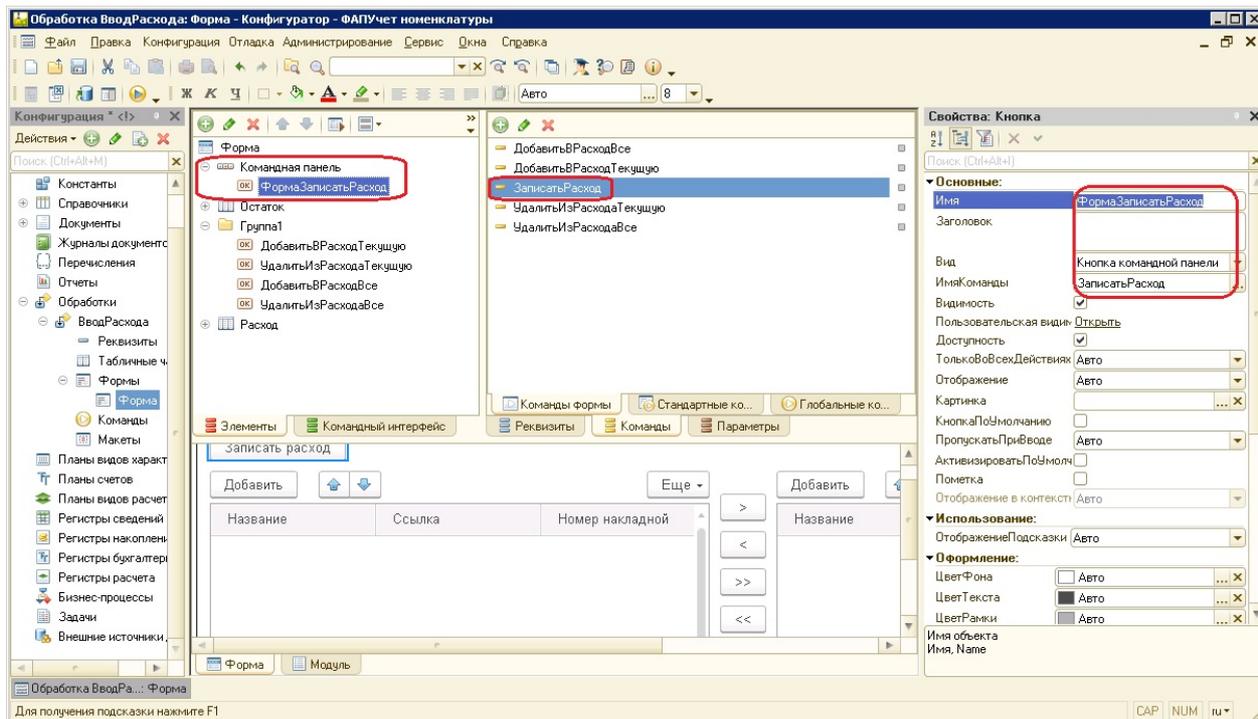


Рис. 38. Добавление команды ЗаписатьРасход

СтрокаРасхода.Ссылка.ОстатокКоличествоЕдиницВОткрытойУпаковке
И СтрокаРасхода.КоличествоУпаковок>

СтрокаРасхода.Ссылка.ОстатокКоличествоУпаковок-1) Или

СтрокаРасхода.КоличествоЕдиницВОткрытойУпаковке>

СтрокаРасхода.Ссылка.ЕдиницВОднойУпаковке Тогда

Сообщить("Расход не может превышать количество остатка",

СтатусСообщения.Важное);

Возрат Ложь;

КонецЕсли;

КонецЦикла;

Возрат Истина;

КонецФункции

Отметим, что перед записью выполняется вызов функции ПроверитьКорректностьРасхода(), которая проверяет корректность введенных данных. Таким образом, форма готова к использованию. Можно запускать в режиме 1С:Предприятия и протестировать работу (рис. 39).

После записи расхода в табличных частях Номенклатуры появляются соответствующие записи о расходе (рис. 40).

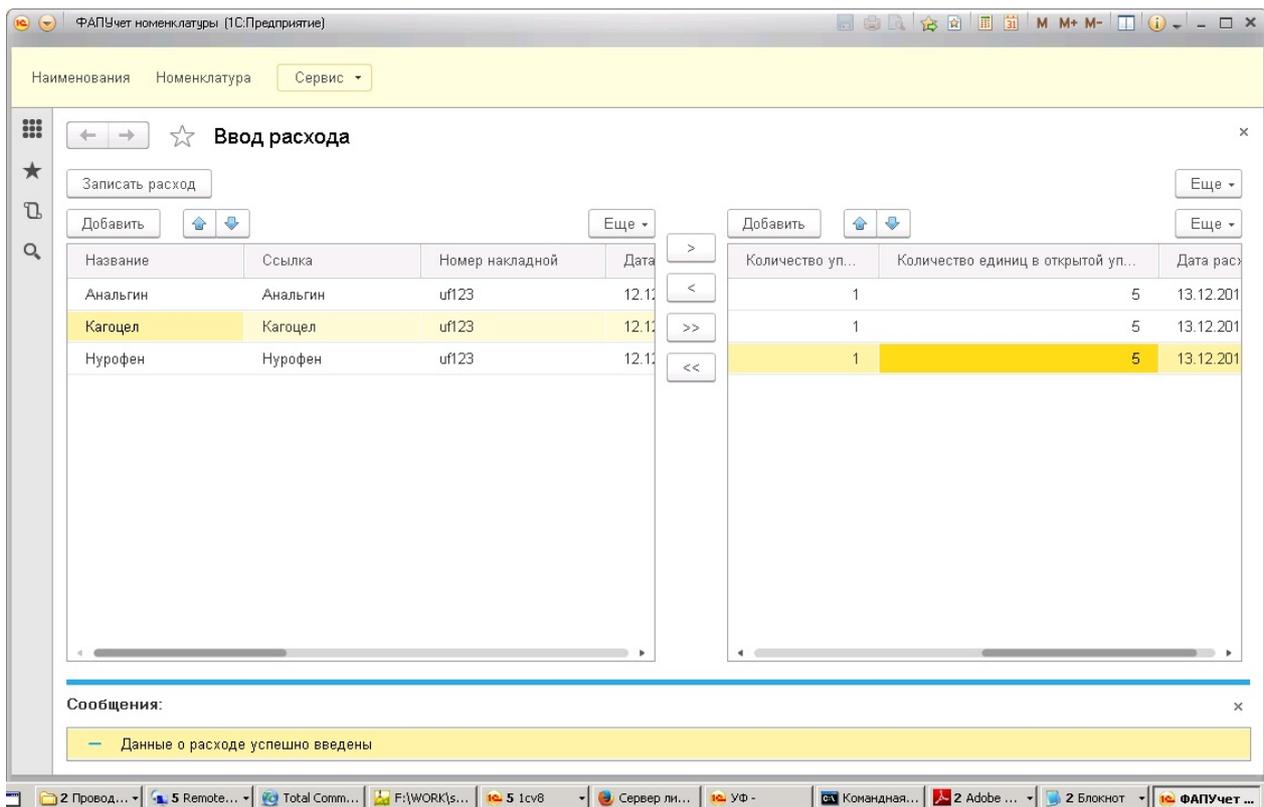


Рис. 39. Работа обработки

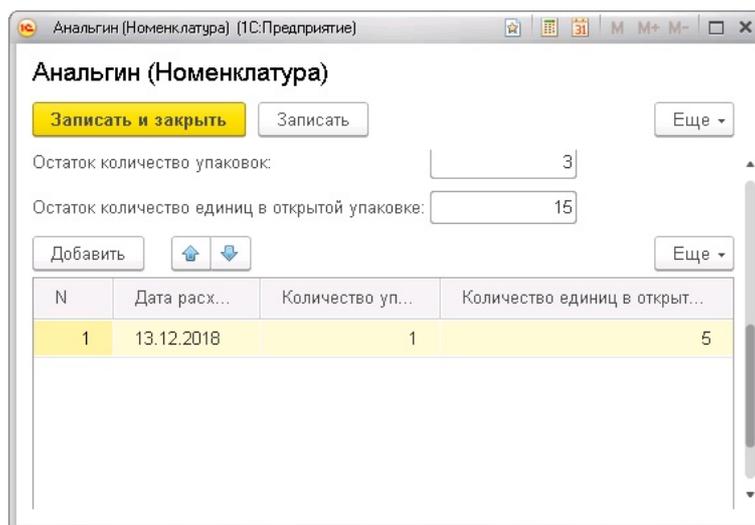


Рис. 40. Расход в табличной части Номенклатуры

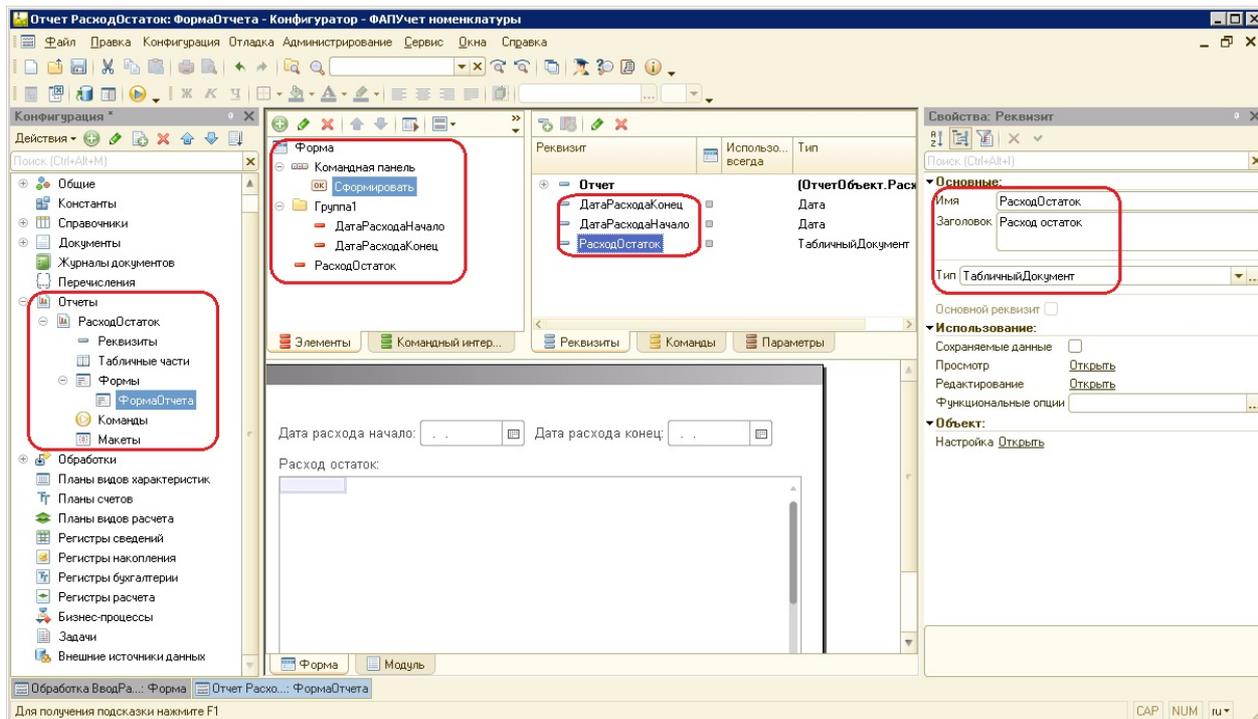


Рис. 41. Добавление отчета

Перейдем к созданию отчета (рис. 41). Добавим в дерево конфигурации отчет с формой. На форме добавим реквизиты:

- *ДатаРасходаНачало* — начало периода регистрации расхода;
- *ДатаРасходаКонец* — конец периода регистрации расхода;
- *РасходОстаток* — табличный документ, который будет содержать данные итогового отчета.

Перенесем реквизиты на форму в удобном виде. Добавим команду Сформировать, при нажатии на которую будет формироваться отчет.

Отчет будет состоять из шапки, содержащей названия колонок, строк с данными и строки с итогами. В свою очередь строки данных будут сгруппированы по названиям Номенклатуры. Оформление всех разделов отчета будем формировать с использованием табличного макета. Для этого в отчет добавляем макет с типом Табличный документ (рис. 42). Таблица заполняется с учетом размеров тех или иных данных и предпочтений пользователя.

Таблица состоит из 3 частей: шапка, данные и итоги. Шапка будет выведена только один раз в итоговый отчет. Данные будут выводиться столько раз, сколько Номенклатуры имеется в справочнике. Итоги будут выводиться для

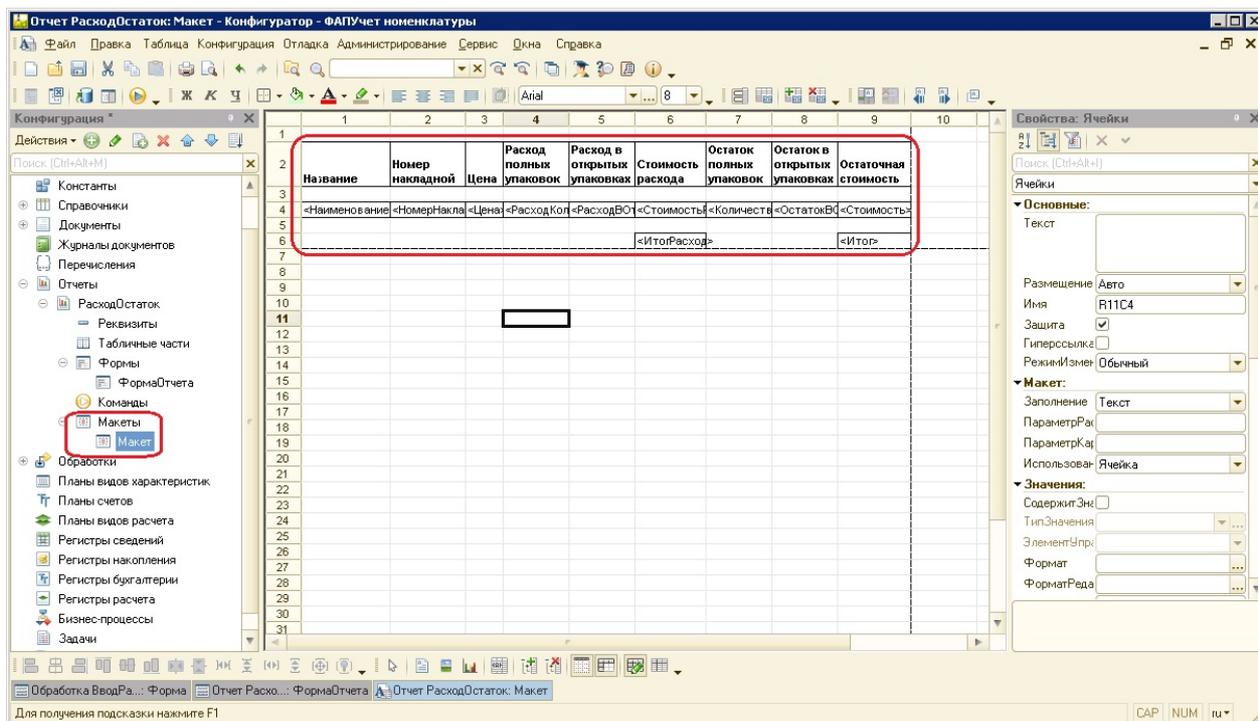


Рис. 42. Добавление макета

каждой группы Номенклатуры и один раз в качестве общего итога. Шапка из представляет собой таблицу с постоянными текстовыми данными, поэтому свойство **Заполнение** каждой ячейки должно быть указано **Текст** (рис. 43).

У всех остальных ячеек, значения которых могут изменяться, свойство **Заполнение** должно быть указано **Параметр**, и должно быть указано имя этого параметра (рис. 44).

Шапка, строки данных и строка итога должны представлять собой неделимые конструкции, поэтому они группируются при помощи указания **Имени**. Для этого необходимо выделить нужную строку макета, в меню выбрать **Таблица → Имена → Назначить имя** и в открывшемся окне указать имя (рис. 45).

Выделенным областям назначаем следующие имена: **Шапка**, **ОбластьДанные**, **ОбластьИтог** (рис. 46).

Макет готов к использованию. В тексте команды **Сформировать** указываем следующий код:

```
&НаКлиенте
Процедура Сформировать(Команда)
СформироватьОчет();
```

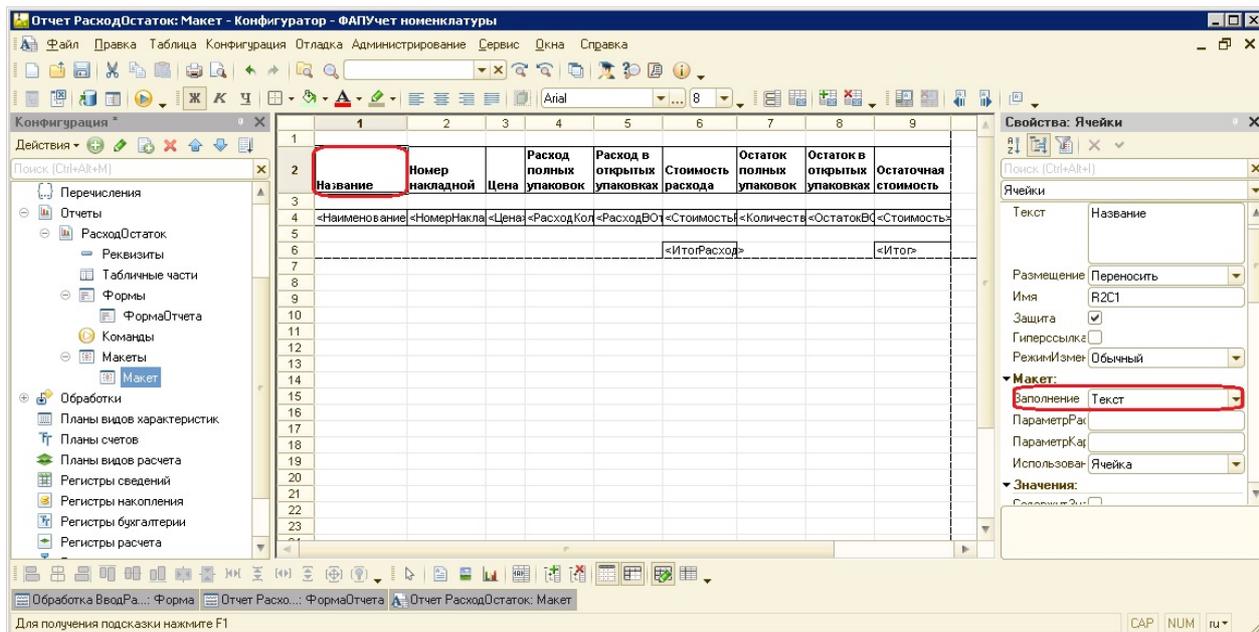


Рис. 43. Заполнение шапки макета

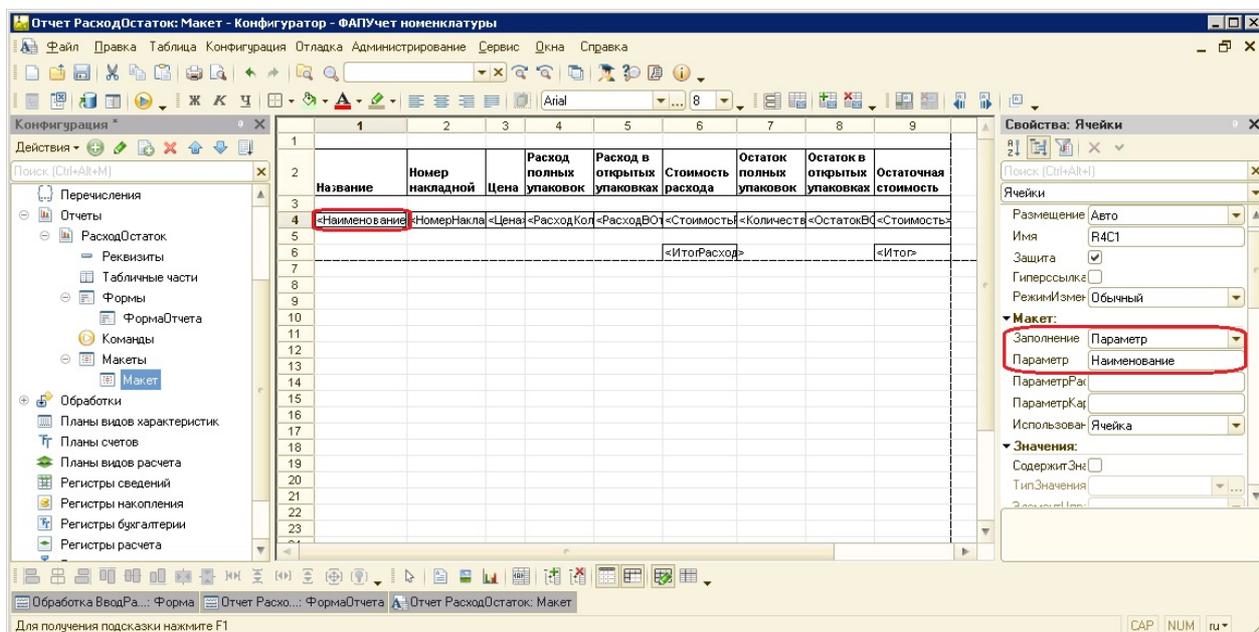


Рис. 44. Заполнение изменяющихся ячеек макета

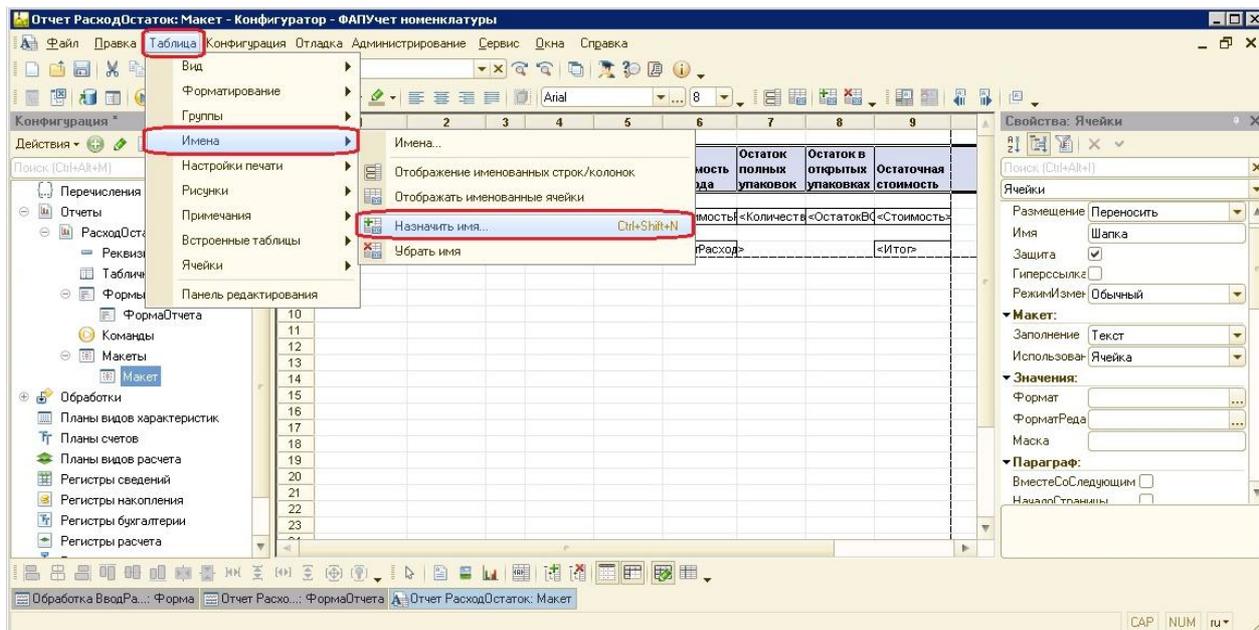


Рис. 45. Назначение имени выделенному фрагменту макета

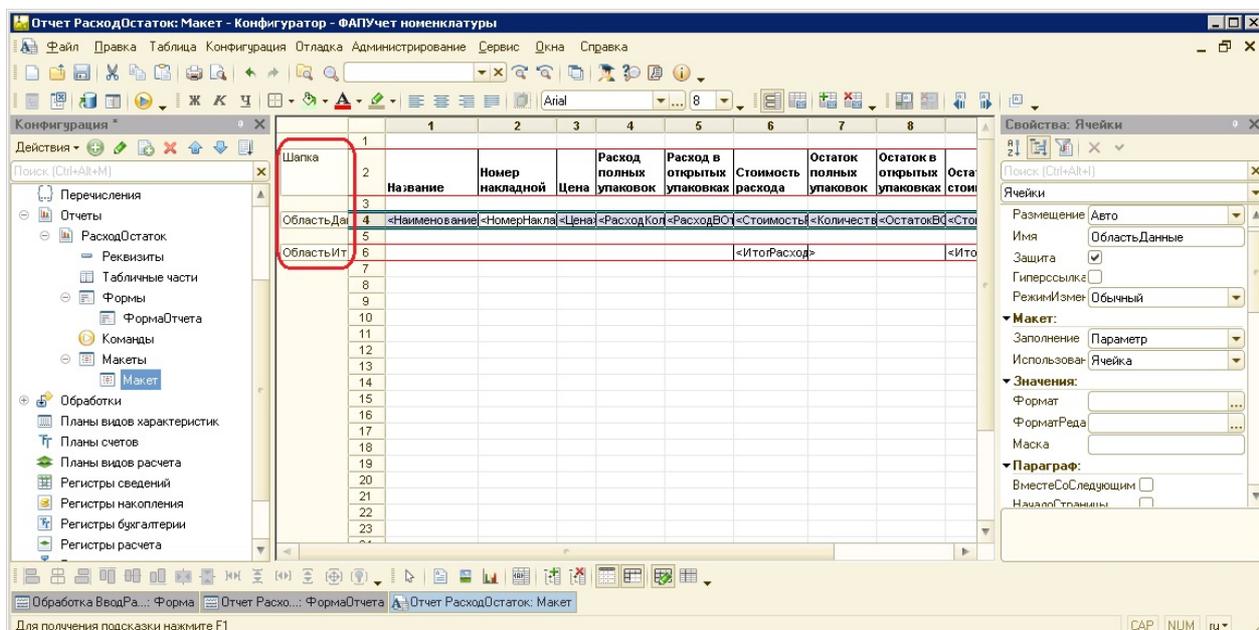


Рис. 46. Назначение имен выделенным областям

КонецПроцедуры

&НаСервере

Функция СформироватьОчет()

```
РасходОстаток=Новый ТабличныйДокумент;  
Макет=Отчеты.РасходОстаток.ПолучитьМакет("Макет");  
РасходОстаток.Вывести(Макет.ПолучитьОбласть("Шапка"));  
ОбластьДанные=Макет.ПолучитьОбласть("ОбластьДанные");  
ОбластьИтог=Макет.ПолучитьОбласть("ОбластьИтог");  
ОбщийИтог=0;  
Запрос=Новый Запрос;  
Запрос.Текст="ВЫБРАТЬ  
| НоменклатураИсторияРасхода.Ссылка КАК Ссылка,  
| СУММА(НоменклатураИсторияРасхода.КоличествоУпаковок)  
|   КАК СУММАКоличествоУпаковок,  
| СУММА(НоменклатураИсторияРасхода.  
|КоличествоЕдиницВОткрытойУпаковке)  
|   КАК СУММАКоличествоЕдиницВОткрытойУпаковке  
|ПОМЕСТИТЬ ВременнаяТаблица  
|ИЗ  
| Справочник.Номенклатура.ИсторияРасхода  
|   КАК НоменклатураИсторияРасхода  
|ГДЕ  
| НоменклатураИсторияРасхода.ДатаРасхода  
|   МЕЖДУ &ДатаРасходаНачало И &ДатаРасходаКонец  
|  
|СГРУППИРОВАТЬ ПО  
| НоменклатураИсторияРасхода.Ссылка  
|;  
|  
|/////////////////////////////////////  
|ВЫБРАТЬ  
| Номенклатура.Ссылка,
```

| Номенклатура.Название КАК Название,
| Номенклатура.НомерНакладной,
| Номенклатура.КоличествоУпаковок,
| Номенклатура.ЕдиницВОднойУпаковке,
| Номенклатура.СтоимостьОстатка,
| Номенклатура.ОстатокКоличествоУпаковок,
| Номенклатура.ОстатокКоличествоЕдиницВОткрытойУпаковке,
| Номенклатура.Цена,
| ВременнаяТаблица.СУММАКоличествоУпаковок,
| ВременнаяТаблица.СУММАКоличествоЕдиницВОткрытойУпаковке,
| Номенклатура.ИсторияРасхода.(
| ДатаРасхода,
| КоличествоУпаковок,
| КоличествоЕдиницВОткрытойУпаковке
|)

| ИЗ

| Справочник.Номенклатура КАК Номенклатура
| ЛЕВОЕ СОЕДИНЕНИЕ ВременнаяТаблица КАК ВременнаяТаблица
| ПО Номенклатура.Ссылка = ВременнаяТаблица.Ссылка

| ГДЕ

| (Номенклатура.СтоимостьОстатка > 0
| ИЛИ ВременнаяТаблица.СУММАКоличествоУпаковок > 0
| ИЛИ ВременнаяТаблица.
| СУММАКоличествоЕдиницВОткрытойУпаковке > 0)

|

| УПОРЯДОЧИТЬ ПО

| Номенклатура.Название.ПолноеНаименование”;

Запрос.УстановитьПараметр(”ДатаРасходаНачало”, ДатаРасходаНачало);

Запрос.УстановитьПараметр(”ДатаРасходаКонец”, ДатаРасходаКонец);

Выборка=Запрос.Выполнить().Выбрать();

текНазвание=Неопределено;

Итог=0;

ИтогРасход=0;

ОбщийИтогРасход=0;

Пока Выборка.Следующий() Цикл

Если Не текНазвание=Неопределено

И Не текНазвание=Выборка.Название Тогда

ОбластьИтог.Параметры.Итог=Итог;

ОбластьИтог.Параметры.ИтогРасход=ИтогРасход;

РасходОстаток.Вывести(ОбластьИтог);

Итог=0;

ИтогРасход=0;

КонецЕсли;

текНазвание=Выборка.Название;

ОбластьДанные.Параметры.Наименование=Выборка.Название;

КоличествоРасходаНаДату=

Выборка.Ссылка.ПолучитьОбъект().

ПолучитьКоличествоРасходаНаДату(

ДатаРасходаНачало, ДатаРасходаКонец);

ОбластьДанные.Параметры.РасходКоличествоУпаковок=

КоличествоРасходаНаДату.КоличествоУпаковок;

ОбластьДанные.Параметры.РасходВОткрытыхУпаковках="" +

КоличествоРасходаНаДату.КоличествоЕдиницВОткрытойУпаковке

+"/"+Выборка.ЕдиницВОднойУпаковке;

ОбластьДанные.Параметры.КоличествоУпаковок=

Выборка.ОстатокКоличествоУпаковок;

ОбластьДанные.Параметры.ОстатокВОткрытыхУпаковках=""

+Выборка.ОстатокКоличествоЕдиницВОткрытойУпаковке+"/"

+Выборка.ЕдиницВОднойУпаковке;

ОбластьДанные.Параметры.Цена=Выборка.Цена;

ОбластьДанные.Параметры.НомерНакладной=

Выборка.НомерНакладной;

ОбластьДанные.Параметры.Стоимость=Выборка.СтоимостьОстатка;

Итог=Итог+Выборка.СтоимостьОстатка;

текРасход=

КоличествоРасходаНаДату.КоличествоУпаковок*Выборка.Цена
+КоличествоРасходаНаДату.КоличествоЕдиницВОткрытойУпаковке
*Выборка.Цена/Выборка.ЕдиницВОднойУпаковке;

ОбластьДанные.Параметры.СтоимостьРасхода=текРасход;

ИтогРасход=ИтогРасход+текРасход;

РасходОстаток.Вывести(ОбластьДанные);

ОбщийИтог=ОбщийИтог+Выборка.СтоимостьОстатка;

ОбщийИтогРасход=ОбщийИтогРасход+текРасход;

КонецЦикла;

ОбластьИтог.Параметры.Итог=Итог;

ОбластьИтог.Параметры.ИтогРасход=ИтогРасход;

РасходОстаток.Вывести(ОбластьИтог);

ОбластьИтог.Параметры.Итог=ОбщийИтог;

ОбластьИтог.Параметры.ИтогРасход=ОбщийИтогРасход;

РасходОстаток.Вывести(ОбластьИтог);

КонецФункции

Отчет готов к использованию. Запускаем базу в режиме 1С:Предприятие, открываем отчет, заполняем параметры отчета и нажимаем кнопку Сформировать (рис. 47).

2.4 Тестирование системы

Тестирование разработанной системы необходимо выполнять таким образом, чтобы предусмотреть все возможные варианты развития событий.

Все обнаруженные ошибки должны быть исправлены, после чего процесс тестирования необходимо начать сначала. Повторять это до тех пор, пока на всех тестовых данных программа не выдаст правильный результат.

Пример.

Протестируем разработанный отчет. Для этого пробуем построить отчет при разных тестовых данных:

1) Указываем разные варианты периода:

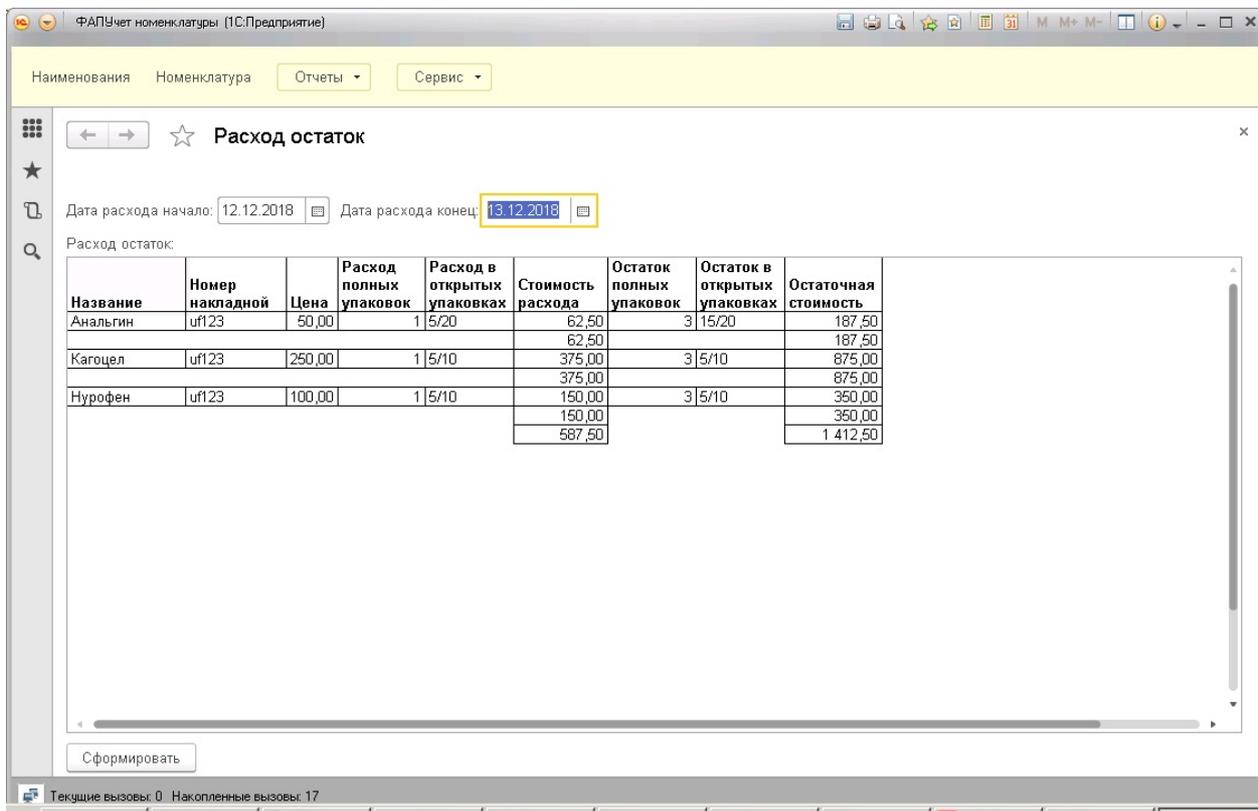


Рис. 47. Проверка работы отчета

- а) Начало периода < конец периода.
 - б) Начало периода = конец периода.
 - в) Начало периода > конец периода (если есть возможность).
- 2) Справочник Номенклатура содержит:
 - а) 0 записей.
 - б) 1 запись на каждую запись в справочнике Названия (связь только один-к-одному).
 - в) 2 или более записей на каждую запись в справочнике Названия (связь многие-к-одному).
 - 3) Табличная часть ИсторияРасхода содержит:
 - а) 0 записей.
 - б) 1 запись.
 - в) 2 или более записей.

Необходимо протестировать отчет с использованием разных сочетаний структуры входных данных. Например, 1a2a3a, 1a2б3a, 1a2б3б и т.д.

2.5 Внедрение системы

Если информационная система разрабатывается по заказу какой-то организации, то она в конечном итоге должна быть внедрена, и должен быть составлен соответствующий акт. В акте о внедрении указывается название разработанной системы, дата внедрения и краткая характеристика. Акт подписывается представителями фирмы, где внедрена разработка, и ставится печать этой организации.

Акт о внедрении не является обязательной частью курсовой работы, но является ее достоинством и документально подтверждает достижение высоких практических результатов.

3 Составление отчета по работе

По результатам проделанной работы должен быть составлен отчет. Примерный план отчета:

1. Титульный лист.
2. Оглавление.
3. Введение.
4. Постановка задачи.
5. Обзорная часть.
6. Описание метода решения.
7. Заключение.
8. Список литературы.
9. Приложения.

3.1 Титульный лист

На титульном листе указывается учебное заведение, в котором выполнена работа; вид работы (в данном случае это «Курсовая работа»); название темы; кем выполнена работа (указывается номер группы, курс, фамилия, имя и отчество); кто является научным руководителем; город и год составления отчета.

Пример:

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Удмуртский государственный университет»
Институт математики, информационных технологий и физики
Кафедра теоретических основ информатики

КУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ

Автоматизированные информационные системы

**НА ТЕМУ «Создание автоматизированной информационной
системы для ФАП в программе 1С:Предприятие»**

Выполнил(а) студент(ка):

Иванов Иван Иванович

(ФИО студента(ки))

Направления подготовки/специальности

Прикладная информатика

группы **ОАБ-09.03.03.д-21**

(наименование группы)

Научный руководитель:

Петров П.П., доцент кафедры ТОИ, к.т.н.

(ФИО, должность, ученое звание, ученая степень)

Итоговая оценка по курсовой работе _____

(оценка, подпись руководителя)

Ижевск 2019 г.

3.2 Оглавление

В оглавлении указываются названия всех разделов с номерами страниц.

| | |
|---|----|
| Введение | 3 |
| 1 Постановка задачи | 4 |
| 1.1 Требования к функционалу | 4 |
| 1.2 Требования к интерфейсу | 5 |
| 1.3 Требования к программному обеспечению | 7 |
| 1.4 Требования к техническому обеспечению | 7 |
| 1.5 Требования к информационному обеспечению | 7 |
| 1.6 Требования к организационному обеспечению | 7 |
| 2 Обзорная часть | 8 |
| 2.1 Описание схемы работы ФАП. | 8 |
| 2.2 Обзор существующих систем | 10 |
| 2.3 Обзор системы «1С:Предприятие» | 12 |
| 3 Описание метода решения | 16 |
| 3.1 Описание структуры данных | 16 |
| 3.2 Разработка функционала ввода расхода | 18 |
| 3.3 Разработка отчета | 21 |
| 3.4 Тестирование системы | 24 |
| Заключение | 25 |
| Список литературы | 26 |
| Приложения | 27 |

3.3 Введение

Введение — одна из главных частей отчета по курсовой работе. Оно должно содержать краткое изложение текста всей работы. Несмотря на то, что текст работы начинается с введения, рекомендуется его составить только после того, как будут готовы все остальные разделы. Это связано с тем, что некоторые сведения можно получить только после того, как будет готова вся работа. Введение должно содержать следующую информацию:

1. **Актуальность работы.** Под актуальностью понимается важность, значимость, востребованность рассматриваемого вопроса в данный момент вре-

мени.

Пример.

Тема данной работы на сегодняшний день весьма **актуальна** потому что, система «1С:Предприятие» в последнее время набирает популярность, и многие организации переходят на использование продуктов фирмы 1С. Нередко возникает необходимость решения проблемы совместимости при переходе с других систем, поэтому профессия 1С –программиста довольно востребована. . .

2. Цель работы. Формулируется исходя из основной проблемы.

Пример.

Цель работы: Автоматизация процесса составления отчетов по расходам и остаткам медикаментов в фельдшерско-акушерском пункте.

3. Задачи. Перечисляются те задачи, при решении которых планируется достижение поставленной цели.

Пример.

Для достижения поставленной цели необходимо решить следующие **задачи:**

- 1) Анализ предметной области.
- 2) Формулировка требований.
- 3) Разработка системы.
- 4) Тестирование системы.
- 5) Внедрение системы.»

5. Объект исследования. Объект — это та сфера (область), на уровне которой происходит изучение темы работы.

Пример.

Объект исследования: Фельдшерско-акушерский пункт.

5) Предмет исследования. Предмет — это свойство объекта или вопрос связанный с этим объектом.

Пример.

Предмет исследования: Автоматизация отчетности фельдшерско-акушерского пункта.

6) Теоретическая и/или практическая значимость. Перечисляются результаты работы, которые принесут реальную пользу обществу. Если говорить

о теоретической значимости, то результаты должны быть полезны в будущих исследованиях как самого автора, так и других.

Пример.

Разрабатываемая система имеет неоспоримую **практическую значимость** для фельдшерско-акушерских пунктов. Составление отчета занимает значительную часть рабочего времени сотрудников ФАП и требует серьезной концентрации внимания. Разрабатываемая система позволит сократить трудозатраты на порядок. . .

7) **Описание структуры работы.** Перечисляются основные разделы работы и краткое их содержание. Также приводятся основные данные по объему работы.

Пример.

Работа состоит из введения, трех глав и заключения.

В введении приводится краткая характеристика работы. В первой главе описывается задача. Во второй главе выполняется анализ предметной области. В третьей главе рассказывается про метод решения задачи. В заключении приводятся основные результаты проделанной работы.

Объем работы 30 страниц. Работа содержит 6 рисунков, 2 таблицы, 3 диаграммы, 1 приложение. Список литературы содержит 10 источников.

3.4 Постановка задачи

В постановке задачи приводится полностью формализованное описание поставленной задачи. Если есть техническое задание, то включается вся ее основная часть, касающаяся требований к разрабатываемой системе. Если объем технического задания слишком большой, то допускается включать только основные моменты в сокращенном виде со ссылкой на полный текст. Полный текст допускается включать в виде приложений в конце отчета.

Пример.

1.1 Требования к функционалу.

Входные данные:

- Накладная с указанным номером и датой. Накладная содержит список медикаментов с указанием названия, цены и количества упаковок. Список

может содержать как лекарственные препараты, так и другие медицинские средства (шприцы, бинты и т.д.)

- Данные о расходе. В конце каждого рабочего дня сотрудник ФАП фиксирует данные об израсходованных медикаментах: из какой партии (номер накладной) израсходованы медикаменты, сколько полных упаковок израсходовано и сколько единиц в открытых упаковках израсходовано.

Должен быть реализован следующий функционал:

- просмотр информации об имеющейся номенклатуре;
- ввод данных о новом приходе;
- ввод данных о расходе;
- составление отчета об израсходованных медикаментах за указанный период и остатках на конец отчетного периода.

Подробнее по каждому пункту: . . .

Выходные данные:

Отчет об израсходованных медикаментах за указанный период и остатках на конец отчетного периода. Отчет должен быть в виде таблицы. Колонки таблицы: Название, Номер накладной, Цена, Расход полных упаковок, Расход в открытых упаковках, Стоимость расхода, Остаток полных упаковок, Остаток в открытых упаковках, Остаточная стоимость. Строки таблицы: Номенклатура, сгруппированная по стандартному наименованию. . . .

1.2 Требования к интерфейсу. . . .

1.3 Требования к программному обеспечению. . . .

1.4 Требования к техническому обеспечению. . . .

1.5 Требования к информационному обеспечению. . . .

1.6 Требования к организационному обеспечению. . . .

3.5 Обзорная часть

В обзорной части должна быть полностью раскрыта проблематика работы. Даже человек, чьими первостепенными интересами не является рассматриваемая тема, прочитав ее, должен войти в курс дела. Прежде чем приступить к этому разделу, требуется изучить необходимую литературу, часть из которой пойдет в раздел список литературы.

Рекомендуется следующий план при оформлении раздела:

1. Суть объекта и предмета исследования. Необходимо подробно описать объект и его свойства. Обратить внимание на те моменты, которые необходимы для правильной оценки важности выполнения работы на данную тему.

2. Степень изученности рассматриваемой проблемы. Если речь идет о какой-либо разработке, то необходимо проанализировать существующие решения по этой теме (либо по близкой теме). Выявить их недостатки, которые оставляют рассматриваемую проблему нерешенной

3. Сравнительный анализ различных точек зрения на эту проблему. Необходимо рассмотреть, как пытаются решить проблему другие. Если работа более практическая, то необходимо выполнить сравнение технологий (методов, программ и т.д.), использующихся при решении задач на эту тему.

4. Выводы. Необходимо кратко выписать промежуточные результаты работы, полученные в текущем разделе.

Советы:

- При сборе материала для курсовой сразу оформлять список литературы. Это позволит в дальнейшем сэкономить время.
- Не использовать прямое цитирование. Это увеличит оригинальность работы и научит правильно выражать свои мысли.
- Включать в раздел только то, что связано с темой. Это позволит сэкономить место.

Пример.

2.1 Описание схемы работы ФАП.

Чаще всего отчетную работу в ФАП ведет сам фельдшер, и это занимает значительную часть его времени. Согласно выполненным исследованиям [1] в отчетные дни это занимает от 30% до 70% рабочего времени в зависимости от различных факторов (население, сезона). Это приводит к тому, что фельдшер не успевает в положенное рабочее время выполнить свои непосредственные обязанности по оказанию медицинской помощи населению (что является более важным, чем составление какой-либо отчетности).

Согласно исследованиям [2]. . .

2.2 Обзор существующих систем

Если рассматривать существующие системы, то наиболее близким по функциональности является система под названием «Управление аптекой» [3]. К основным функциональным возможностям этой системы относятся [4]:

- Оформление прихода товаров;
- Оформление реализации товаров и услуг контрагенту;
- Оформление перемещения товаров между аптеками;
- Оформление возвратов товаров от покупателей;
- Оформление документов инвентаризации товаров;
- Оформление приходных и расходных кассовых ордеров;
- Оформление чеков продажи.

...

Специализированных решений для ФАП в настоящее время пока не существует. Если использовать существующие системы, такие как «Управление аптекой» или аналогичные, то приходится сталкиваться со следующими проблемами:

- Предоставляют возможность учета номенклатуры только с расходными полными упаковками;
- Платные;
- Сложные в освоении ввиду наличия большого количества функционала, ненужного для ФАП .

...

В связи с невозможностью использования существующих систем, требуется разработка собственного решения, специализированного именно для ФАП. Для ее реализации могут быть использованы следующие программные средства ... [5], ... [6], ..., система «1С:Предприятие»[7]. Выполним сравнительный анализ для выбора наиболее эффективного средства.

...

С учетом того, что «1С:Предприятие» предоставляет возможность дальнейшей модификации используемой программы, и код этой программы может быть написан с использованием русскоязычного синтаксиса, наиболее подходящим средством является «1С:Предприятие».

(Если в работе выделяются еще и учебные цели, то рекомендуется выполнить обзор языка программирования.)

2.3 Обзор системы «1С:Предприятие»

Язык запросов 1С.

Для того, чтобы обратиться к данным при помощи запроса, сначала во встроенном языке создается объект Запрос:

```
ЗапросКДанным=Новый Запрос;
```

У объекта Запрос есть свойство Текст, в которое нужно поместить текст запроса, написанный на языке запросов. В тексте запроса описывается, какие данные, из каких таблиц нужно получить и как эти данные представить:

```
ЗапросКДанным.Текст=
```

```
”Выбрать
```

```
| Наименование
```

```
|Из
```

```
| Справочник.Товары”;
```

Далее запрос выполняется с помощью метода Выполнить() объекта Запрос:

```
РезультатЗапросаКДанным=ЗапросКДанным.Выполнить();
```

Именно в этот момент и происходит чтение данных из базы данных. Прочитанные данные возвращаются в виде объекта РезультатЗапроса, содержащего выбранные данные из базы данных. Чтобы обработать данные, содержащиеся в объекте РезультатЗапроса, из результата запроса получается выборка с помощью метода Выбрать(), который возвращает новый объект ВыборкаИзРезультатаЗапроса, то есть коллекцию данных, предназначенную для последовательного обхода ее элементов:

```
Выборка=РезультатЗапросаКДанным.Выбрать();
```

Далее выборка обходится с помощью цикла Пока Выборка.Следующий() Цикл, а в теле цикла производятся какие-то действия над данными, полученными с помощью запроса:

```
Пока Выборка.Следующий() Цикл
```

```
Сообщить(Выборка.Наименование);
```

```
КонецЦикла;
```

В результате, если соединить вместе текст, мы получим процедуру встроенного языка, в которой создается и выполняется запрос, выводящий наиме-

нование всех товаров из справочника Товары в окно сообщений:

ЗапросКДанным.Текст=

”Выбрать

| Наименование

|Из

| Справочник.Товары”;

РезультатЗапросаКданным=ЗапросКДанным.Выполнить();

Выборка=РезультатЗапросаКданным.Выбрать();

Пока Выборка.Следующий() Цикл

Сообщить(Выборка.Наименование);

КонецЦикла;

Язык запросов «1С:Предприятия» основан на стандартном SQL, но при этом содержит большое количество расширений, ориентированных на финансово-экономические задачи, и значительно облегчает разработку бизнес-приложений. Язык запросов — мощный инструмент, предоставляющий разнообразные возможности получения данных.

Текст запроса состоит из нескольких частей (секций):

- описание запроса,
- объединение запросов,
- упорядочивание результатов,
- автоупорядочивание,
- описание итогов.

Обязательной частью запроса является только первая — описание запроса. Все остальные присутствуют в запросе по необходимости.

...

Главным преимуществом «1С:Предприятия» является то, что разрешается писать код программы с использованием русского синтаксиса. Кроме того, текст запроса описывается также с использованием ключевых слов на русском языке.



Рис. 48. ER-диаграмма

3.6 Описание метода решения

В описании метода решения приводятся пошагово ход решения всех задач и основные практические результаты.

Если работа теоретическая, то все используемые понятия должны быть строго определены, все выводы — обоснованы, а теоремы — доказаны. Если работа имеет более практическое значение, то должно быть строго обосновано использование тех или иных технологий при решении поставленных задач. Необходимо акцентировать внимание на те моменты, которые имеют перво-степенное значение в работе.

Пример.

3.1 Описание структуры данных

Согласно построенной ER-диаграмме (рис. 48) данные будут храниться в двух справочниках: Номенклатура и Наименования. Первый будет содержать табличную часть ИсторияРасхода.

Реквизиты справочника Номенклатура:

- *НомерНакладной* с типом строка, который будет хранить информацию о номере накладной поступления;
- *ДатаПрихода* с типом дата, который будет хранить информацию о дате поступления;
- *КоличествоУпаковок* с типом число (целое), который будет хранить информацию о количестве упаковок поступившей номенклатуры;
- *Цена* с типом число (с точностью 2 знака после запятой), который будет хранить информация о цене за одну упаковку;

- *ЕдиницВОднойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц хранится в одной упаковке (таблеток, ампул и т.д.).
- *Название* с типом СправочникСсылка.Наименования, который позволит группировать номенклатуру по наименованиям.
- *СтоимостьОстатка* с типом число (с точностью 2 знака после запятой), который будет хранить информацию о стоимости оставшейся части медикаментов;
- *ОстатокКоличествоУпаковок* с типом число (целое), который будет хранить информацию о том, сколько полных упаковок осталось;
- *ОстатокКоличествоЕдиницВОткрытойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц осталось в открытых упаковках.

Реквизиты справочника Наименования:

- *ПолноеНаименование* с типом строка, который будет хранить полное наименование;

Реквизиты табличной части ИсторияРасхода:

- *ДатаРасхода* с типом дата, который будет хранить информация о дате расхода;
- *КоличествоУпаковок* с типом число (целое), который будет хранить информацию о том, сколько полных упаковок израсходовано;
- *КоличествоЕдиницВОткрытойУпаковке* с типом число (целое), который будет хранить информацию о том, сколько единиц использовано в открытых упаковках.

...

3.2 Разработка функционала ввода расхода

Функционал ввода расхода реализуется в обработке ВводРасхода, которая будет предоставлять удобный интерфейс для фиксирования данных о расходе.

Функционал ввода расхода должен предоставлять следующие возможности:

- просмотр информации о наличии номенклатуры, которая может быть израсходована;

- ввод расхода сразу по нескольким позициям номенклатуры;
- проверку корректности расхода (не должен превышать остаток).

Форма обработки будет состоять из двух функциональных областей: в левой — отображение информации о наличии номенклатуры, а в правой — о текущем расходе.

...

3.3 Разработка отчета

Отчет состоит из 3 реквизитов:

- *ДатаРасходаНачало* — начало периода регистрации расхода;
- *ДатаРасходаКонец* — конец периода регистрации расхода;
- *РасходОстаток* — табличный документ, который будет содержать данные итогового отчета.

Основной запрос к данным выглядит следующим образом:

...

3.4 Тестирование системы

Тестирование разработанной системы выполнялось на следующих тестовых данных:

...

По результатам тестирования ошибок не выявлено, на всех тестовых данных обработка и отчет отработали корректно и выдали правильные результаты.

3.7 Заключение

В заключении приводятся основные результаты работы. Необходимо сделать выводы по решению каждой поставленной задачи и о достижении итоговой цели. Если какая-либо из задач не решена полностью, то перечислить причины.

Пример.

В ходе выполнения курсовой работы были решены все поставленные задачи.

1) Выполнен подробный анализ предметной области. На основе этого выявлены основные требования к разрабатываемой системе. Согласно выполненному обзору готовых систем, которые можно было бы использовать, на данный момент не существует. . .

2) Сформулированы все требования к системе. Составлено техническое задание. . .

3) Информационная система полностью разработана и соответствует техническому заданию. . .

4) Разработанная система протестирована. Ошибок не выявлено. . .

5) Система внедрена на местном ФАП. Это подтверждается соответствующим актом.

Итоговая цель, а именно, автоматизация процесса составления отчетов по расходам и остаткам медикаментов в ФАП, успешно достигнута. Согласно отзывам сотрудников ФАП, экономия времени на составление отчетов составляет от 50% до 90%.

3.8 Список литературы

В списке литературы выписываются все источники, которые были использованы в работе. Оформляется по соответствующему стандарту. На каждый источник в тексте отчета должна быть хотя бы одна ссылка.

3.9 Приложения

В приложении приводятся все материалы (тексты программ, схемы, скриншоты, инструкции и т.д.), которые не вошли в основной текст отчета, но необходимы для пояснения хода решения задачи.

4 Защита

Защита курсовой — это выступление студента по теме своей работы перед экзаменационной комиссией. Проводится с целью определения уровня знаний студента и углубленности в тему исследования. для того, чтобы защитить свою курсовую работу и получить оценку, необходимо быть допущенным к этой защите.

4.1 Допуск к защите

К защите допускаются студенты, которые не позднее чем за 48 часов до назначенной даты предоставят на кафедру следующие документы:

1. текст отчета в электронном виде;
2. текст отчета в распечатанном виде;
3. файл презентации;
4. отзыв от научного руководителя в электронном либо бумажном виде.

4.2 Презентация

Презентации принимаются в формате pdf либо pptx. В презентацию включаются:

1. Титульная страница. Указывается тема работы, номера группы, ФИО студента, должность и ФИО научного руководителя.
2. Введение. Кратко приводятся актуальность работы, объект и предмет исследования, цель и задачи.
3. Обзорная часть. Приводятся основные результаты обзорной части курсовой работы.
4. Практическая часть. Приводятся основные результаты практической части курсовой работы.
5. Заключение. Приводятся итоговые результаты работы.

Рекомендации при составлении презентации:

1. Делать не более 10 слайдов. Так как время на выступление ограничено, то на большее количество слайдов просто может не хватить времени.
2. Для каждого слайда указывать заголовок.
3. Текст не должен сливаться с фоном, так как на защите будет использован проектор, и такой текст будет сложно прочитать. Рекомендуется использовать черные буквы на белом фоне.
4. Минимизировать количество текста. Лучше вместо текста использовать картинки, диаграммы, схемы.

5. Не делать больших схем на одном слайде. Необходимо учесть, что чем больше схема, тем меньше шрифт текста и сложнее его прочитать.
6. Размер шрифта рекомендуется делать 18-24.

4.3 Речь

Речь строится согласно построенной презентации и с учетом регламентированного времени на выступление:

1. Представиться. Назвать ФИО, номер группы, название темы, ФИО и должность руководителя.
2. Кратко рассказать об актуальности работы, объекте и предмете исследования, цели и задачах.
3. Рассказать про результаты обзорной части работы.
4. Рассказать про практические результаты.
5. Сделать общее заключение по работе.
6. Сообщить о завершении доклада и готовности ответить на вопросы комиссии.

Рекомендации при выступлении:

1. Не делать больших пауз. Время выступления сильно ограничено, нужно стараться каждую секунду использовать по назначению.
2. Не говорить дословно то, что написано на слайдах. Если рассказывать другими словами, то в большинстве случаев это позволяет сделать информацию на слайдах более понятным. Кроме того, у комиссии составляется впечатление о том, что студент хорошо разбирается в своей теме. Текст презентации итак видят слушатели, поэтому лучше при выступлении приводить дополнительную информацию, которая не уместилась на слайдах.

Запрещается при выступлении:

1. Читать текст с бумажки или с экрана.
2. Рассказывать стоя спиной к слушателям.
3. Рассказывать дольше, чем позволяет регламент.

4.4 Порядок защиты

1. В конце каждого семестра назначается комиссия из педагогического состава кафедры.
2. Не позднее чем за 7 суток до назначенного времени защиты объявляется дата и место проведения защиты.
3. Студент предоставляет все необходимые материалы на кафедру. Электронные материалы отправляются по электронной почте. Адрес почты можно узнать у руководителя либо у методиста на кафедре.
4. Если студент предоставил все материалы, то по электронной почте ему сообщается номер в списке выступающих.
5. В назначенное время студент приходит с зачетной книжкой в назначенную аудиторию.
6. При наступлении своей очереди студент выступает перед комиссией. На выступление дается не более 5 минут.
7. После выступления комиссия задает вопросы и студент на них отвечает.
8. После выступления всех студентов объявляется перерыв, и в аудитории остаются только члены комиссии и научные руководители для выставления оценок.
9. После перерыва студенты приглашаются обратно в аудиторию, и объявляются оценки.

4.5 Итоговая оценка

Итоговая оценка выставляется на основании следующих показателей: оценка от руководителя, оценка выступления, оценка ответов на вопросы.

4.6 Повторная защита

В случае получения неудовлетворительной итоговой оценки студент может защититься повторно. Число повторных защит может быть не более 3. Последовательность действий всегда одинакова: студент исправляет недочеты в работе и предоставляет на кафедру исправленные материалы, после чего назначается новая защита.

Список литературы

1. Пространство имен System.Windows.Forms [Электронный ресурс] / Microsoft — Электрон. дан. — Режим доступа: [https://msdn.microsoft.com/ru-ru/library/system.windows.forms\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.windows.forms(v=vs.110).aspx), свободный. — Загл. с экрана. Дата обращения: 12.10.2016
2. Радченко М.Г., Хрусталева Е.Ю. 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы. Электронная книга в формате pdf; ISBN 978-5-9677-2043-7. - М.: ООО «1С-Публишинг», 2013. - 964 с.
3. Хрусталева Е. Ю. Язык запросов «1С:Предприятия 8». ISBN 978-5-9677-1987-5, - М.: ООО «1С-Публишинг», 2013, - 358 с.
4. Ажеронок В.А., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Разработка управляемого интерфейса. ISBN 978-5-9677-1148-0, - М.: ООО «1С-Публишинг», 2010, - 723 с.
5. Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. Реализация прикладных задач в системе «1С:Предприятие 8.2». ISBN 978-5-9677-1387-3, - М.: ООО «1С-Публишинг», 2010, - 714 с.
6. Гончаров Д.И., Хрусталева Е.Ю. Решение специальных прикладных задач в «1С:Предприятии 8.2». ISBN 978-5-9677-1611-9, - М.: ООО «1С-Публишинг», 2012, - 300 с.
7. Канивец Д., Безбородов А. Как создать удобное и понятное приложение в «1С:Предприятии 8.2». - М.: ООО «1С-Публишинг», 2010, - 180 с.

Содержание

| | |
|---|-----------|
| Введение | 3 |
| 1 Выбор темы | 4 |
| 2 Решение задачи | 6 |
| 2.1 Анализ предметной области | 6 |
| 2.2 Формулировка требований | 8 |
| 2.3 Разработка системы | 9 |
| 2.4 Тестирование системы | 48 |
| 2.5 Внедрение системы | 50 |
| 3 Составление отчета по работе | 50 |
| 3.1 Титульный лист | 50 |
| 3.2 Оглавление | 52 |
| 3.3 Введение | 52 |
| 3.4 Постановка задачи | 54 |
| 3.5 Обзорная часть | 55 |
| 3.6 Описание метода решения | 60 |
| 3.7 Заключение | 62 |
| 3.8 Список литературы | 63 |
| 3.9 Приложения | 63 |
| 4 Защита | 63 |
| 4.1 Допуск к защите | 64 |
| 4.2 Презентация | 64 |
| 4.3 Речь | 65 |
| 4.4 Порядок защиты | 66 |
| 4.5 Итоговая оценка | 66 |
| 4.6 Повторная защита | 66 |

Учебное издание

Сапаров Алексей Юрьевич

**Методические указания по выполнению
курсовой работы для студентов
направления подготовки «Прикладная информатика»**

Методическое пособие

Отпечатано с оригинал-макета заказчика

Подписано в печать 01.10.19. Формат 60x84¹/₁₆.

Тираж 50 экз. Заказ № 2255.

Типография

Издательского центра «Удмуртский университет»
426034, Ижевск, ул. Университетская, 1, корп. 2.

Тел. 68-57-18