

Требования к выполнению лабораторных работ

Тематика лабораторных работ: автоматизация хозяйственной деятельности предприятия некоего профиля.

Выполнение каждой лабораторной работы заключается в реализации всех пунктов сценария, приведенного в разделе «Задания к работе». Задания лабораторных работ выполняются в соответствии с предметной областью, определяемой вариантом студента или выбранной последним по согласованию с преподавателем.

Студент обязан перед выполнением каждой лабораторной работы самостоятельно ознакомиться с теоретическим материалом и по её результатам предоставить отчет.

Требования к отчётам о выполнении лабораторных работ

Все отчёты о выполнении лабораторных работ оформляются в печатном виде на листах формата А4.

Каждый отчёт должен содержать:

1. Заголовок лабораторной работы.
2. Цель работы.
3. Задание к лабораторной работе.
4. Ход выполнения работы.
5. Экранные формы разработанных решений.
6. Текст программы, если он необходим при решении поставленной задачи.
7. Выводы по сделанной работе.

Лабораторная работа №1

ЗНАКОМСТВО С СИСТЕМОЙ 1С:ПРЕДПРИЯТИЕ

Цель работы: получение практических навыков по установке и начальной работе с системой 1С:Предприятие; знакомство с интерфейсом программы и назначением основных объектов.

Основные понятия

Общие сведения о системе 1С:Предприятие

Система 1С:Предприятие является универсальной системой автоматизации экономической и организационной деятельности предприятия. Поскольку такая деятельность может быть довольно разнообразной, система 1С:Предприятие имеет возможность «приспосабливаться» к особенностям конкретной области деятельности, в которой она используется. Для обозначения такой способности используется термин конфигурируемость, то есть возможность настройки системы на особенности конкретного предприятия и класса решаемых задач.

Это достигается тем, что 1С:Предприятие – это не просто программа, существующая в виде набора неизменяемых файлов, а совокупность различных программных инструментов, с которыми работают разработчики и пользователи. Логически всю систему можно разделить на две большие части, которые тесно взаимодействуют друг с другом: *конфигурацию* и *платформу*, которая управляет работой конфигурации. Платформа – это среда разработки приложений (конфигураций) высокого уровня. Она же обеспечивает и работу приложений (выполнение программного кода). В платформе определен набор базовых объектов со своими свойствами и методами – это константы, документы, справочники, регистры и другие. Всего набора объектов достаточно для решения задач автоматизации учета, связанного с бизнесом (и многих сопутствующих задач). Для работы со множеством объектов был создан специальный язык разработки – встроенный язык 1С:Предприятие 8. Таким образом, из имеющегося в платформе набора объектов можно создавать конкретные приложения (конфигурации). Очень важно, что конфигурация – это решение, открытое для редактирования.

Существует одна платформа (1С:Предприятие 8) и множество конфигураций. Для функционирования какого-либо прикладного

решения всегда необходима платформа и какая-либо (одна) конфигурация.

Начиная с версии 8.2, в системе 1С:Предприятие существует несколько клиентских приложений: *толстый клиент*, *тонкий клиент*, *веб-клиент*. «Толстым» клиент называется потому, что может исполнять практически всю функциональность, предоставляемую встроенным языком системы 1С. По этой же причине он требует значительного количества аппаратных ресурсов на компьютере пользователя и может обращаться к базе данных (в файловом режиме работы) или к кластеру серверов 1С:Предприятия 8 (в клиент-серверном варианте) только посредством файлового доступа или по локальной сети. В операционной системе Windows исполняемый файл этого приложения – *1cv8.exe*. «Тонким» клиент называется потому, что умеет исполнять ограниченный набор функциональности встроенного языка. Вся работа с базой данных, объектными данными, исполнение запросов – выполняется на стороне сервера. Тонкий клиент только получает готовые данные, подготовленные для отображения. В операционной системе Windows исполняемый файл этого приложения – *1cv8c.exe*. Веб-клиент, в отличие от толстого и тонкого клиентов, исполняется не в среде операционной системы компьютера, а в среде интернет-браузера (*Windows Internet Explorer*, *Mozilla Firefox*, *Google Chrome* или *Safari*): любому пользователю достаточно ввести адрес веб-сервера, на котором опубликована информационная база. Работе через веб-клиент сопутствует ряд ограничений: нет возможности работать с файловой системой, XML и электронной почтой. Невозможность использования веб-клиента может быть связана с требованиями информационной безопасности.

Развитие платформы 8.3 связано с охватом мобильных устройств. Для работы на мобильных устройствах может использоваться специальная технология – *мобильная платформа* 1С:Предприятия 8. Она позволяет создавать самостоятельные приложения, работающие на мобильных устройствах под управлением операционных систем Android или iOS. Такими устройствами, как правило, являются смартфоны и планшетные ПК. Приложение, созданное с помощью этой технологии, хранит прикладные данные прямо на мобильном устройстве и позволяет их изменять. Как правило, мобильные приложения не предназначены для автономной работы, а используются в качестве удалённых рабочих мест какого-то «основного» прикладного решения.

Гибкость платформы позволяет применять 1С:Предприятие 8 в самых разнообразных областях:

- автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.
- поддержка оперативного управления предприятием;
- автоматизация организационной и хозяйственной деятельности;
- ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность;
- широкие возможности для управленческого учета и построения аналитической отчетности, поддержка многовалютного учета;
- решение задач планирования, бюджетирования и финансового анализа;
- расчет зарплаты и управление персоналом;
- другие области применения.

Поскольку задачи автоматизации могут быть самыми разными, фирма «1С» и ее партнеры выпускают прикладные решения, каждое из которых предназначено для автоматизации одной определенной области человеческой деятельности.

В качестве примеров существующих прикладных решений можно привести решение *«Управление торговлей»*, предназначенное для автоматизации задач оперативного и управленческого учета, *«Управление персоналом»*, предназначенное для реализации кадровой политики компании или *«Управление производственным предприятием»*, охватывающее основные контуры учета и управления на производственном предприятии, *«Бухгалтерия предприятия»*, обеспечивает решение всех основных задач бухгалтерского учета.

Прикладное решение является, по возможности, универсальным, чтобы удовлетворить потребности самых разных предприятий, работающих в одной области деятельности. Такая универсальность неизбежно приведет к тому, что на конкретном предприятии будут использоваться далеко не все возможности прикладного решения, а каких-то возможностей в нем будет не доставать.

Вот тут и выходит на передний план *конфигурируемость* системы, поскольку платформа, помимо управления работы конфигурацией, содержит средства, позволяющие вносить изменения в используемую конфигурацию. Более того, платформа позволяет создать свою собственную конфигурацию «с нуля», если по каким-либо причинам использование типовой конфигурации представляется нецелесообразным.

Для того чтобы было возможно вносить изменения в программные модули системы, 1С:Предприятие имеет различные режимы работы: *1С:Предприятие* и *Конфигуратор*. Режим 1С:Предприятие является основным и служит для работы пользователей системы. В этом

режиме пользователи вносят данные, обрабатывают их и получают выходные результаты. Режим конфигулятора используется разработчиками и администраторами баз данных. Именно этот режим и предоставляет инструменты, необходимые для модификации существующей или создания новой конфигурации.

Установка системы 1С:Предприятие

Перед установкой системы полезно ознакомиться с системными требованиями 1С:Предприятия 8 (<http://www.v8.1c.ru/requirements/>).

Установка 1С:Предприятие может выполняться первично, когда производится полная установка нужных компонент или повторно, когда есть необходимость изменения, переустановки или удаления некоторых или всех компонент системы.

При наличии дополнительной многопользовательской лицензии установку 1С:Предприятия 8 необходимо проводить на всех рабочих местах, поэтому рекомендуется произвести установку по образцу или административную установку, а затем выполнить установку на рабочих местах.

Интерфейс программы установки выбирается автоматически по значению региональных установок операционной системы пользователя. Для указания иного интерфейса следует использовать ключ командной строки /L. После запуска программа установки анализирует установленные компоненты и производит выбор режима установки.

Начальная установка

На компьютере пользователя запускается файл Setup.exe из дистрибутива установки 1С:Предприятие 8 и в диалоговом режиме выбираются компоненты для установки. На последнем шаге программа установки производит копирование файлов и создание программной группы и ярлыков. После окончания данных действий программа установки предлагает произвести установку драйвера защиты от несанкционированного использования HASP Device Driver.

Установка драйвера требуется, если к USB-порту данного компьютера будет присоединяться аппаратный ключ защиты (локальный или серверный). При использовании сетевой лицензии или программной защиты драйвер устанавливать не имеет смысла.

Обновление 1С:Предприятие

Новые версии 1С:Предприятие выпускается в виде полного дистрибутива и в виде дистрибутива обновления (поставляется на диске ИТС). Любой из них может быть использован для выполнения обновления установленной программы.

Программа обновления выполняет обновление только тех компонент, которые были установлены. Для изменения, восстановления или исправления установленного экземпляра 1С:Предприятие необходимо использовать Панель управления/Установка и удаление программ (Control Panel/Add or Remove Programs).

Установка по образцу

При необходимости установки системы 1С:Предприятие 8 на несколько однотипных компьютеров (например, для подготовки компьютерного класса) можно выполнить установку на один компьютер, а затем провести установки на остальные компьютеры по образцу первой.

Для этого необходимо выполнить запуск программы установки *setup.exe* из командной строки с указанием ключей.

При первой установке необходимо задать ключ */r*. В результате установки в системном каталоге Windows будет создан файл *setup.iss*, содержащий ответы пользователя на вопросы программы установки (для изменения каталога файла *setup.iss* можно использовать ключ */f1* - например, */f1"c:\temp\setup.iss"*).

Этот файл можно использовать при последующих установках, задавая ключ */s*, - файл *setup.iss* должен находиться в одном каталоге с *setup.exe* или же его местонахождение должно быть задано ключом */f1*.

При установке по образцу программа установки создаст файл *setup.log*, содержащий записи о ходе установки - этот файл будет создан в том же каталоге, что и *setup.iss*, однако можно явно задать каталог для *setup.log*, используя ключ */f2* - например, */f2"c:\temp\setup.log"*.

Административная установка

Данный режим предназначен для установки системы 1С:Предприятие на компьютеры, объединённые в локальную сеть.

При использовании административной установки администратор локальной сети получает возможность устанавливать 1С:Предприятие с единого источника в сети, а затем обновлять версию системы только в месте административной установки. Дальнейшее обновление версий на локальных рабочих местах производят сами пользователи при очередном запуске системы.

Поскольку административная установка рассчитана на подготовленных пользователей, возможность такой установки не вынесена в пользовательский интерфейс программы установки *setup.exe*. Для выполнения административной установки необходимо запустить программу установки *setup.exe* с ключом */a*. После запуска на экран

выводится диалог, в котором необходимо указать каталог административной установки. В указанном каталоге будет создан набор файлов, необходимый для установки по сети на локальных рабочих местах.

Для установки системы 1С:Предприятие на локальных рабочих местах необходимо запустить программу *setup.exe* из каталога административной установки.

При выходе очередной версии 1С:Предприятия выполняется процедура обновления административной установки. При очередном запуске 1С:Предприятия с локального рабочего места пользователю будет предложено произвести обновление локальной установки системы. При подтверждении необходимости обновления процедура обновления производится в автоматическом режиме.

Если по каким-либо причинам требуется отказаться от проверки наличия обновления, то достаточно удалить файл *Admupd.cfg*.

Установка конфигурации

Конфигурации для системы 1С:Предприятие устанавливаются из шаблонов. Установка шаблонов выполняется специальной программой установки, которая создается при создании комплекта поставки из Конфигуратора. После запуска шаблона программа установки предлагает выбрать каталог на жестком диске, куда установится шаблон конфигурации.

В том случае, если в указанном вами каталоге для установки шаблона конфигурации уже существуют подкаталоги с установленной ранее конфигурацией, программа сообщает о найденной конфигурации и повторно запрашивает о необходимости установки в данный каталог. В случае подтверждения будет произведена перезапись файлов конфигурации.

Отличительные особенности установки 1С:Предприятие в варианте клиент-сервер

1С:Предприятие позволяет работать с информационными базами в варианте «клиент-сервер» под которым понимается архитектура с тремя программными уровнями:

- клиентское приложение 1С:Предприятие;
- сервер 1С:Предприятие;
- сервер баз данных.

Клиентское приложение 1С:Предприятия - это программа, с которой работает конечный пользователь. Для того, чтобы 1С:Предприятие получило возможность работать с информационными базами в варианте «клиент-сервер», обычная установка, позволяющая

работать с файловым вариантом информационной базы, должна быть дополнена специализированными компонентами доступа к серверу 1С:Предприятие. Это достигается во время процедуры установки выбором флажка «Компоненты доступа к серверу 1С:Предприятия 8». При этом устанавливаемое 1С: Предприятие получает возможность работы с информационными базами, как в файловом варианте, так и в варианте «клиент-сервер».

Установка сервера 1С:Предприятия выполняется программой установки 1С:Предприятия, которая копирует на компьютер необходимые файлы и настраивает сервер 1С:Предприятия. Приложение сервера 1С:Предприятия выполняется от лица специального пользователя, например «USER1CV8SERVER», который также может быть создан в процессе установки программы.

Сервер 1С:Предприятия - это специализированное серверное приложение, через которое осуществляется доступ к информационной базе в варианте «клиент-сервер». Он образует промежуточный программный слой между клиентским приложением и сервером баз данных. Клиентские приложения не имеют непосредственного доступа к серверу баз данных. При этом, помимо простой передачи данных от клиентского приложения серверу баз данных, сервер 1С:Предприятие выполняет и ряд других задач. В частности, в его среде может быть организовано выполнение достаточно сложных обработок, написанных на встроенном языке системы 1С:Предприятие.

Сервер 1С:Предприятие функционирует на компьютере, работающем под управлением Microsoft Windows XP/Vista/Server 2003/Server2008/Server2012/7/8/8.1 и Linux (архитектура x86 и x86-64).

Хранение данных информационных баз в варианте «клиент-сервер» обеспечивается сервером баз данных, в качестве которого может использоваться MS SQL Server/PostgreSQL/IBM DB2/Oracle Database. При этом каждая информационная база целиком сохраняется в отдельной базе данных MS SQL Server/PostgreSQL/IBM DB2/Oracle Database.

Клиентские приложения 1С:Предприятие, сервер 1С:Предприятие и сервер баз данных могут быть распределены по компьютерам достаточно произвольным образом. Также они вполне могут работать и на одном компьютере. Сервер 1С:Предприятие и сервер баз данных при работе создают примерно одинаковую нагрузку на компьютеры, на которых они исполняются.

Сервер 1С:Предприятие защищен от несанкционированного использования с помощью ключа аппаратной защиты HASP4.

Совместное использование различных версий системы 1С:Предприятия

Если при установке была обнаружена более ранняя версия 1С:Предприятия и предполагается использовать данную версию совместно с прежней, то для установки новой версии (меньше 8.2) следует перед установкой скопировать каталог существующей установки (по умолчанию «C:\Program Files\1Cv8»). Для сделанной копии необходимо создать новые ярлыки или внести исправления в ярлыки, созданные ранее при установке 1С:Предприятия 8. Если существующая установка производилась с административной установки, то в копии установки необходимо удалить файл admupd.cfg для отключения механизма автоматического обновления.

Начиная с версии 1С:Предприятие 8.2 вводится гибкий механизм использования множества различных версий системы на одном компьютере. В этом случае каждая новая версия системы автоматически устанавливается в новую папку с именем номера версии, что не приводит к затиранию предыдущих версий, установленных на компьютере.

Совет. При установке 2-х и более различных конфигураций программы 1С:Предприятие целесообразно создать в папке Program Files (а еще лучше не на том диске, где установлена ОС Windows и программы из пакета MS Office) папку для программы и отдельные папки для каждой из конфигураций (баз данных).

Понятие «информационная база» и создание её резервной копии

К информационным базам относится наиболее ответственная информация, включающая: конфигурацию, все данные о хозяйственной деятельности предприятия, а также административную информацию. Все данные, относящиеся к информационной базе, объединяются в базу данных. Потеря или искажение каких-то данных информационной базы может привести к потере работоспособности системы, построенной на базе 1С:Предприятия.

Данные, которые определяют логику функционирования системы на базе 1С:Предприятие, относятся к информационной базе. Хранение информационной базы осуществляется в базе данных в виде набора таблиц, для чего 1С:Предприятие может использовать одну из двух систем управления базами данных (СУБД):

– Встроенную в 1С:Предприятие (файловый вариант информационной базы). В этом случае все данные информационной базы хранятся в файле, например с именем 1Cv8.1CD. Этот файл имеет двоичный

формат и по сути является базой данных для встроенной в 1С:Предприятие СУБД.

– Внешний сервер баз данных (клиент-серверный вариант информационной базы). Все данные информационной базы хранятся на соответствующем сервере баз данных, например MS SQL Server.

Периодически (в зависимости от интенсивности изменения базы) рекомендуется создавать архивную копию рабочей базы данных на автономных носителях информации для возможности восстановления рабочей базы из архива при её случайном разрушении.

Резервное копирование и восстановление базы данных, хранящей информационную базу, обязательно должно выполнялось только целиком. С этой целью рекомендуется использование средств резервного копирования баз данных, встроенных в сервер БД. Резервное сохранение файлового варианта информационной базы может быть выполнено копированием файла информационной базы, например 1Cv8.1CD.

В конфигураторе есть специальная функция: Администрирование > Выгрузить информационную базу. С ее помощью можно выгрузить в указанный файл (файл выгрузки) все данные, относящиеся к информационной базе, и больше никакие. Обратная ей функция, Загрузить информационную базу, позволяет в текущую информационную базу вместо существующих загрузить все данные из файла выгрузки. Эти функции также можно использовать для резервного копирования данных информационной базы, как в файловом, так и в клиент-серверном варианте.

Создание пустой информационной базы без конфигурации

Сначала на диске создается пустой каталог с именем базы. Потом запускается 1С:Предприятие 8, в результате чего отображается окно запуска системы 1С:Предприятие с двумя режимами – 1С:Предприятие и Конфигуратор.

Добавление новой информационной базы осуществляется путём нажатия кнопки Добавить... окна запуска 1С:Предприятие. Для создания пустой информационной базы во вновь открывшемся окне выбирается пункт Создание новой информационной базы, а в следующем окне – Создание информационной базы без конфигурации....

Начиная с версии 8.0, добавлена возможность создания информационных баз при помощи шаблонов, которые могут поставляться фирмой 1С, ее партнерами и другими организациями. Это упрощает процесс создания множества информационных баз.

Для создаваемой информационной базы задаётся имя, тип расположения, каталог (если выбран тип расположения На данном компьютере) и язык.

Файловая структура базы

В каталоге базы находятся (могут находиться) следующие файлы, представленные в табл.1.

Начиная с версии 8.0, список и настройки параметров информационных баз расположены не в реестре Windows, как в версии 1С 7.7, а в каталоге профиля пользователя "Application Data\1С\1Сv8". Теперь при установке/переустановке операционной системы не нужно искать и сохранять ветки реестра, а нужно просто скопировать файл *ib.lst*. Там же находятся и настройки пользователя – **.pfl*.

Таблица 1. Структура ИБ системы 1С:Предприятие

файл	Описание
1Cv8.1CD	Файл информационной базы (структура и данные)
1Cv8.pfl	Параметры и настройки по умолчанию
*.st	Файлы шаблонов
1Cv8.log	Лог-файл журнала регистрации
.****.epf	Файлы внешних обработок и отчетов
*.dt	Архивная копия базы со всеми данными
*.cf	файл поставки конфигурации
*.cfu	файл обновления

Понятие «конфигурация»

Функционирование системы делится на два процесса — *конфигурирование* (описание модели предметной области средствами системы) и *исполнение* (обработку данных предметной области). Результатом конфигурирования является конфигурация, которая представляет собой модель предметной области.

Конфигурацией в системе 1С:Предприятие называется совокупность взаимосвязанных составных частей:

- структуры учетных данных, форм их ввода, выбора, печати;
- состав механизмов учета итоговых данных и движений учетных данных;
- состав различных отчетов и обработок;
- набора пользовательских интерфейсов;
- набора ролей (прав доступа);
- набора общих процедур и функций (глобальный модуль и общие модули), макетов табличных документов, и др.;
- вспомогательных объектов: картинки, шаблоны, стили и т.д.

Объект конфигурации

Под **объектом конфигурации** в системе 1С:Предприятие понимается формальное описание группы понятий (предметной области, средств взаимодействия пользователя с системой) со сходными характеристиками и одинаковым предназначением.

Каждый объект конфигурации обладает уникальным набором свойств. Этот набор описан на уровне системы и не может быть изменен в процессе настройки конфигурации задачи. Набор свойств объекта конфигурации определяется, в основном, его назначением в системе 1С: Предприятие.

Главным свойством любого объекта конфигурации является имя — краткое наименование объекта конфигурации. При создании нового объекта конфигурации ему автоматически присваивается условное имя, состоящее из слова, определяемого по виду объекта, и цифры (например, при создании реквизита создается реквизит с именем Реквизит1, при создании документа — Документ1, и т.д.). Это имя можно изменить в процессе редактирования свойств объекта конфигурации, при этом система отслеживает уникальность имен. Имя объекта конфигурации не может быть пустым.

Основы работы в конфигураторе

После запуска программы в режиме конфигуратора открывается окно конфигуратора, с помощью которого настраивается или создается требуемая конфигурация.

Первая команда, с которой начинается работа с любой конфигурацией — Конфигурация > Открыть конфигурацию. На экране откроется дерево конфигурации (дерево метаданных).

Формально дерево конфигурации представляет собой объединённые в виды объекты конфигурации. В таблице 2 приведена краткая характеристика основных видов объектов конфигурации системы 1С: Предприятие.

Таблица 2. Основные объекты конфигурации

Наименование	Краткая характеристика
Общие	Группа вспомогательных объектов конфигурации, с помощью которых производится настройка пользовательского интерфейса; определяются права доступа к объектам конфигурации; разрабатываются механизмы взаимодействия пользователей с данными; настраиваются механизмы обмена данными с внешними системами.
Константы	Постоянные (условно-постоянные) величины. Константы хранят информацию, которая не

	изменяется или изменяется достаточно редко: название организации, ее почтовый адрес и так далее.
Справочники	Списки однородных элементов данных. Используются для хранения нормативно-справочной информации.
Документы	Служат для ввода информации о совершенных хозяйственных операциях.
Журналы документов	Служат для отображения списков документов разного вида.
Перечисления	Списки значений, задаваемые на этапе конфигурирования.
Отчеты	Средство получения выходной информации. Источником данных для построения отчетов служат документы, справочники и регистры, также используется информация, хранящаяся в константах.
Обработки	Объекты конфигурации этого вида используются для выполнения различных действий над информационной базой.

В окне конфигурации мы можем создавать, редактировать и удалять нужные нам объекты. При этом система контролирует ссылочную целостность структуры метаданных – если на данный объект есть ссылки из других объектов, то мы не сможем его удалить. Для проверки ссылок следует использовать Поиск ссылок на объект или Поиск ссылок в объекте из меню кнопки Действия дерева метаданных. Также существует Глобальный поиск из меню Правка. С помощью него можно находить заданную строку не только во всех текстах программных модулей, но и элементах интерфейса, макетах (печатных формах), внешних файлах и т.д.

Информацию по конструкциям встроенного языка программирования можно получить, используя Синтаксис-помощник из меню Справка.

Редактирование существующих свойств объекта производится в Палитре свойств. Поскольку разные объекты конфигурации имеют разные свойства, содержимое этого окна будет меняться в зависимости от того, какой объект является активным. При некоторых действиях разработчика (например, создание табличной части), палитра свойств открывается автоматически. Но существует возможность открытия палитры свойств, для объекта конфигурации самостоятельно, воспользовавшись пунктом Свойства контекстного меню или сочетанием клавиш [Alt+Enter].

Сохранение конфигурации

После изменений в конфигурации ее следует сохранять. Для сохранения выбираем меню Конфигурация > Сохранить конфигурацию либо стандартная комбинация клавиш [Ctrl+S].

Начиная с версии 8.0, изменение конфигурации двухступенчатое: есть понятие Конфигурация и Конфигурация базы данных. Теперь не нужно прекращать работу пользователей в запущенных конфигурациях для того, чтобы сохранить текущие изменения. Просто в нужные моменты сохраняем конфигурацию, при этом конфигурация базы данных не затрагивается. И лишь когда всё реализовано и необходимо осуществить проверку реализации – выполняется перенос изменений из конфигурации в конфигурацию базы данных (Конфигурация > Обновить конфигурацию базы данных или клавиша [F7]). В этом случае нужен монопольный доступ к базе.

Режим работы пользователя

Начиная с версии 8.2 появляется возможность выбора режима работы пользователя: *обычный* и *управляемое приложение*. Обычный режим обеспечивает совместимость информационных систем, разработанных в более ранних версиях. Режим *Управляемое приложение* позволяет работать с *управляемым интерфейсом* и *управляемыми формами*. Последний режим ориентирован на комфортную и эффективную работу с системой, основа которой определяется возможностями web-технологий.

Константы

В системе 1С:Предприятие *константы* предназначены для хранения постоянной или условно-постоянной информации. Такая информация либо совсем не изменяется в процессе деятельности предприятия, либо изменяется достаточно редко, но, как правило, часто используется в работе. Например, в константах может храниться наименование предприятия, его ИНН, организационно-правовая форма и другая подробная информация.

В системе может быть описано неограниченное количество констант.

Основная причина использования констант заключается в том, что в них один раз заносится какая-либо информация, которая затем может использоваться при формировании документов, в расчетах, при построении отчетных форм. Значение константы время от времени может редактироваться.

Создание нового объекта конфигурации (например, создание константы) производится следующим образом:

1. устанавливается курсор на тот тип объекта, который необходимо создать. В нашем случае на Константы;

2. в командной панели окна конфигурации нажимается кнопка Действия > Добавить, либо, вызвав контекстное меню, выбирается пункт Добавить.

В качестве имени константы может служить её название, написанное без пробелов, начинающееся с буквы. Например: НазваниеОрганизации. После ввода имени константы автоматически прописывается синоним, в котором расставляются пробелы и заменяются заглавные буквы.

Константа является типизированным объектом конфигурации. Например, тип константы НазваниеОрганизации будет Строка.

В 1С:Предприятии 8 тип Строка доработан: помимо фиксированной длины (дополнение пробелами) и неограниченной длины, появилась возможность задать переменную длину. Длина реквизита такого типа может быть меньше заданной. При сравнении таких строк не нужно обрезать лишние пробелы.

Перечисления

Перечисление является прикладным объектом конфигурации и предназначено для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации.

Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять их, удалять или добавлять новые. Пользователь видит значения перечислений только в списках выбора на формах. Следовательно, на перечисления могут опираться алгоритмы работы программы.

Справочники

При заполнении бланка какого-либо документа часто требуется указывать информацию, выбирая значение из заранее заданного списка. Для работы с постоянной и условно постоянной информацией с некоторым множеством значений в системе используются объекты типа Справочник.

Справочники используются в тех случаях, когда необходимо исключить неоднозначный ввод информации. Например, для того, чтобы покупатель, продавец, кладовщик, директор понимали, о каком товаре идет речь, каждый должен называть его одинаково.

Система 1С:Предприятие позволяет вести практически неограниченное количество необходимых справочников. Каждый справочник представляет собой список однородных объектов: сотрудников, организаций, товаров и т.д. Каждый такой объект будем называть элементом справочника.

Следует иметь в виду, что *в Конфигураторе создается структура справочника*, а собственно его содержимое — элементы справочника — вводятся при работе с программой пользователем.

Как правило, справочники имеют стандартные реквизиты Код и Наименование, при этом код может иметь тип число или строка.

Система 1С:Предприятие предоставляет широкие возможности по работе с кодами элементов справочника: автоматическое присвоение кодов, автоматический контроль уникальности кода и другие.

Справочник в системе 1С: Предприятие может быть иерархическим. В этом случае все данные справочника будут разделяться на два вида: «просто» элементы справочника и группы справочника. Группы — это логическое объединение элементов справочника. Примером иерархического справочника может служить справочник товаров, где группами являются группы товаров («Сантехника», «Бытовая химия» и т.д.), а элементами — конкретные товары («Смеситель», «Зеркало», «Стиральный порошок»).

Помимо кода и наименования, для справочника можно создать набор реквизитов, позволяющих хранить дополнительную информацию об элементе справочника. Например, справочник Контрагенты может содержать такие сведения как полное наименование контрагента, его ИНН, фамилии директора и главного бухгалтера, и другую информацию.

Если объект предметной области, которой соответствует справочник, имеет не только «простые» свойства, например, полное наименование или ИНН, но и составные (списочные) свойства, справочнику может быть создан набор табличных частей.

Создание справочника

Рассмотрим основную концепцию создания справочника на примере создания справочника контрагентов. Под контрагентами будем подразумевать клиентов, с которыми работает предприятие. То есть клиентов, у которых покупается товар и которым продается.

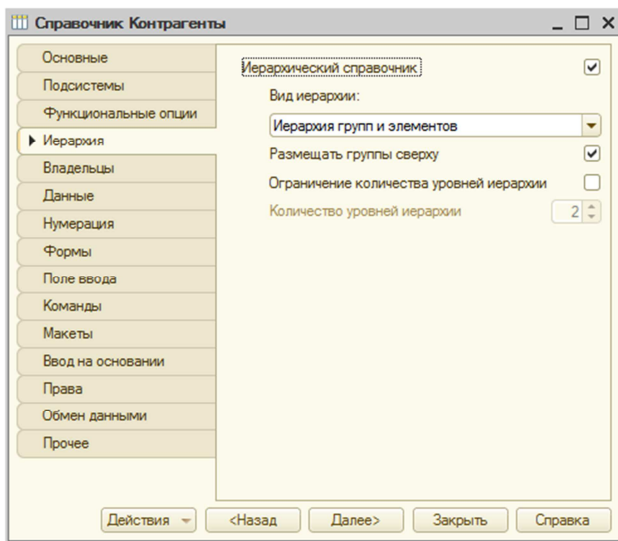
Создадим справочник, в котором будут храниться наименования клиентов. После того, как вы нажмете кнопку *Добавить*, система откроет окно редактирования объекта конфигурации (мастер объекта). Зададим Имя, Синоним и Комментарий справочника.

Свойство Синоним предназначено для хранения альтернативного наименования объекта конфигурации, которое будет использовано в элементах интерфейса нашей программы – то есть будет показано пользователю.

На вкладке Основные мастера объекта можно указать различные представления в пользовательском интерфейсе для этого объекта (Представление объекта, Расширенное представление объекта, Представление списка, Расширенное представление списка). Если оставить поля пустыми, во всех формах пользовательского интерфейса будет использован Синоним.

По кнопке Далее > или перемещаясь по вкладкам, настроим параметры иерархии объекта конфигурации. Примем, что у нас происходит разделение контрагентов на две группы: поставщики и покупатели.

Если разделение на группы не происходит, то создается простой список. В нашем случае справочник контрагентов получается многоуровневым. Для этого ставим галочку Иерархический справочник. После этого становятся доступны другие реквизиты.



В отличие от версии 7.7, в версии 8 есть два варианта иерархии: Иерархия групп и элементов и Иерархия элементов. Первый вариант аналогичен 7.7 – есть элементы справочника, есть группы, причем набор реквизитов групп, обычно, отличается от элементов.

Во втором случае справочник состоит из однотипных элементов. Любой элемент может быть группой для набора других элементов. То есть все элементы справочника – это однотипные сущности, но с разным уровнем детализации, например, справочник подразделений. Каждое подразделение может входить в состав более крупного подразделения.

Для справочника с клиентами подходит первый вариант.

Если снять галочку с пункта Размещать группы сверху, то группы будут сортироваться в порядке, установленном для элементов, например по алфавиту. Оставим как есть.

В версии 8 снято ограничение на максимальную вложенность элементов справочников. И рекомендуется количество вложенностей без крайней надобности не ограничивать. При необходимости ставим галочку Ограничение количества уровней иерархии и указываем максимально возможное количество вложений.

На вкладке Владельцы задается список владельцев данного справочника и вариант использования подчинения. Например, можно настроить, что элементы справочника Договоры будет принадлежать какому-то конкретному контрагенту и только ему. Использование механизма подчинения исключает ситуацию, когда договора одного контрагента присваивают другому.

В версии 8 у справочника может быть несколько владельцев, то есть справочник может быть подчинен сразу нескольким объектам, но каждый элемент подчиненного справочника имеет только одного владельца.

В примере мы не будем подчинять справочник другим объектам.


На закладке **Данные** определяется весь состав реквизитов справочника и, если нужно, состав табличных частей.

Длина кода по-умолчанию нас устроит. (9 символов дают 10^9 -1 элементов при числовом представлении кода). Длину наименования изменим на 150. Тип кода может быть числовой и строковый.

Основное представление определяет, в виде чего (код или наименование) будет отображаться ссылка на элемент справочника в формах выбора.

Реквизиты – это поля данных, содержащие дополнительные сведения о конкретном объекте. Заведем основные данные о контрагентах:

Имя	Тип	Параметры
НаименованиеПолное	Строка	длина = 250, переменная
НаименованиеСокращенное	Строка	длина = 250, переменная
ИНН	Строка	длина = 12, переменная
КПП	Строка	длина = 9, переменная
ДатаРегистрации	Дата	
ЮридическийАдрес	Строка	неограниченная длина
Телефон	Строка	
WebСайт	Строка	

Для создания нового реквизита вызываем контекстное меню заголовка ветки **Реквизиты** и выбираем **Добавить** (либо используем кнопку  на панели инструментов).

Свойство **Использование** реквизита содержит в выпадающем списке три варианта:



- для элемента – реквизит может принадлежать только элементу справочника;
- для группы – реквизит может принадлежать только группе справочника;
- для группы и элемента – реквизит может принадлежать как элементу справочника, так и группе.

Конечно, это имеет смысл только для иерархического справочника.

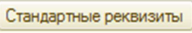
Свойство **Индексировать**: позволяет установить индексирование справочника по значениям данного реквизита для ускорения отбора по нему. Не стоит им злоупотреблять – может сказаться на производительности в целом. Поле станет неактивным, если тип данных – строка неограниченной длины.

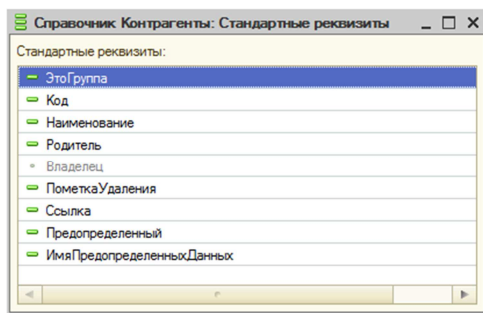
Последняя категория – Тип данных. Не стоит часто использовать строку неограниченной длины, она занимает лишние ресурсы.

В 1С:Предприятии 8 появилась возможность задавать составной тип данных для реквизитов справочника (и реквизитов других объектов) и возможность для справочника создать одну или несколько табличных частей. В них рекомендуется хранить связанную с элементом справочника информацию, которая не имеет собственной объектной сущности. Например, мы сделаем табличную часть для контактных лиц контрагентов.

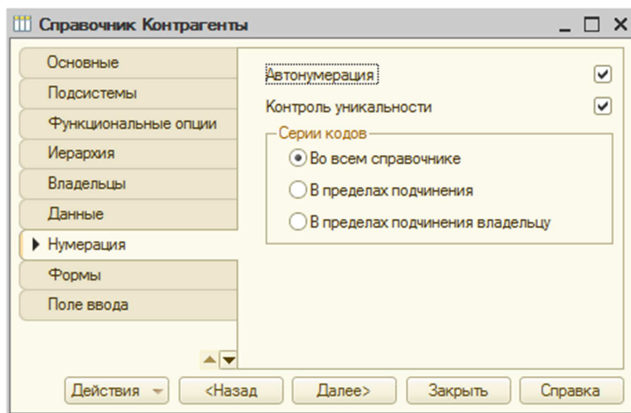
Нажимаем кнопку  и называем созданную табличную часть КонтактныеЛица. Нажимаем кнопку  и создаем реквизиты табличной части. Добавим следующие реквизиты:

Имя	Тип	Параметры
ФИО	Строка	длина = 50, переменная
Должность	Строка	длина = 30, переменная
РабочийТелефон	Строка	длина = 20, переменная
МобТелефон	Строка	длина = 20, переменная
АдресЭлПочты	Строка	Длина = 50, переменная

Также через вкладку Данные доступны стандартные реквизиты справочника (по кнопке , заложенные на уровне платформы. По мере задействования механизмов справочника (иерархия, владелец) эти реквизиты становятся активными. Для каждого из них можно вызвать Палитру свойств [Alt+Enter].



Нажимаем кнопку Далее > и переходим к настройке параметров нумерации справочников. Второй и третий варианты серий кодов имеют смысл только для подчиненных справочников: для них можно выбрать, чтобы каждая группа элементов, подчиненная одному владельцу, имела свою нумерацию. Иначе нумерация будет сквозная (вариант Во всем справочнике). Все остальное оставляем без изменения и переходим на вкладку Поле ввода.

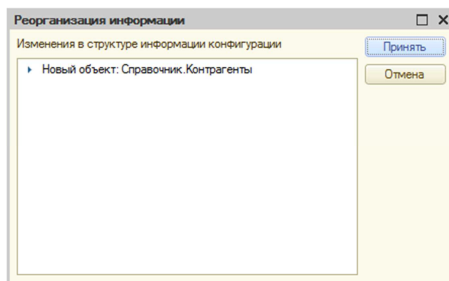


Вкладка Поле ввода позволяет задействовать механизм Ввода по строке для пользовательского режима (по части слова, введенной в поле ввода на форме, пользователю будут показаны подходящие варианты подстановки для быстрого выбора нужного элемента).

Знакомство со свойствами справочников продолжим на этапе выполнения практического задания к работе.

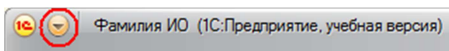
После того как были сделаны изменения в конфигурации (после создания справочников и других объектов конфигурации), из конфигуратора выполним пункт меню Отладка > Начать отладку [F5].

На появившейся вопрос Редактируемая конфигурация отличается от конфигурации базы данных. Обновить конфигурацию базы данных? ответим «Да» и в следующем окне нажмем Принять:

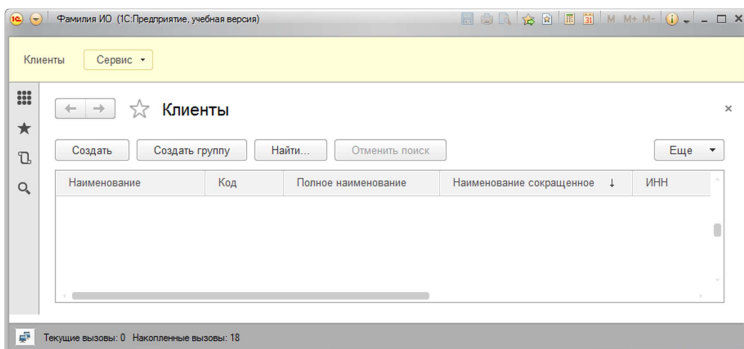


Перед нами откроется окно системы в режиме 1С:Предприятие.

Для просмотра результатов нашей работы можно воспользоваться навигационными ссылками в панелях, либо пунктом Все функции главного меню, появляющегося при нажатии на кнопку в левом верхнем углу окна приложения:



Выбираем справочник Клиенты. Система откроет одну из основных форм справочника – форму списка, сгенерированную автоматически, поскольку при создании справочника на вкладке «Формы» мы оставили значения по умолчанию.



Подробно элементы пользовательского интерфейса и принципы его построения будут рассмотрены в следующей работе.

В процессе отладки допускается редактирование текущей конфигурации и сохранение изменений. Но для отладки новых изменений потребуется завершить работу в режиме 1С:Предприятие, сохранить изменения, обновить конфигурацию базы данных и повторно запустить Отладчик.

Задания к работе

1. Установите систему 1С:Предприятие 8. Учебная версия в папку по умолчанию технологической платформы («сетевая папка установки»\platform\setup.exe).

2. Ознакомьтесь с элементами окна Запуск 1С:Предприятия (учебной версии).

3. Установите конфигурацию Бухгалтерия предприятия (учебная) («сетевая папка установки»\accounting\setup.exe).

4. Создайте новую информационную базу Бухгалтерия предприятия учебная (демо) на основе одноименного шаблона.

5. Создайте новую чистую информационную базу без конфигурации, наименование информационной базы – 1CD_ФамилияИО_ИТ-41 (база будет использоваться в дальнейшем в течение всего семестра для конфигурирования по индивидуальным заданиям).

6. Для корневого элемента дерева метаданных (по умолчанию – Конфигурация) через палитру свойств заполните свойства Имя, Синоним, Краткая информация, Подробная информация, Логотип, Заставка, Авторские права, Адрес информации о поставщике, Адрес информации о конфигурации, Поставщик, Версия. Изменения отразятся в разделе О Программе (команда Справка > О программе...).

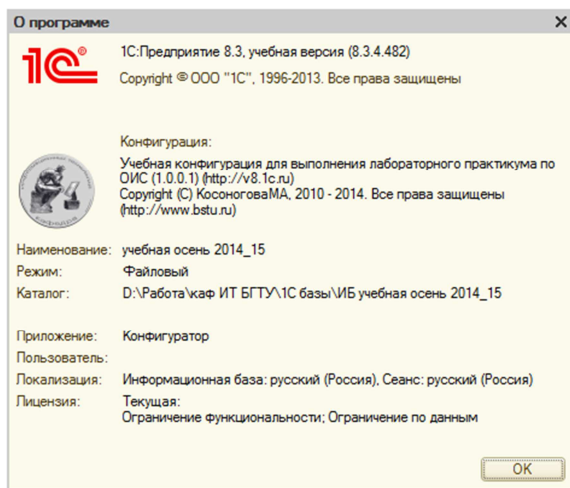


Рис. 1 – Информация о программе

7. Добавьте общий реквизит Комментарий (тип Строка, Неограниченная длина, включен Многострочный режим).

8. Создайте константы с общими сведениями об организации:

Имя	Тип	Параметры
НаименованиеПолное	Строка	длина = 250, переменная
НаименованиеСокращенное	Строка	длина = 250, переменная
ИНН	Строка	длина = 12, переменная
КПП	Строка	длина = 9, переменная
ДатаРегистрации	Дата	
ЮридическийАдрес	Строка	неограниченная длина
Телефон	Строка	
WebСайт	Строка	
ПрофильПредприятия	Строка	неограниченная длина

9. Создайте форму констант, разместите на ней все созданные константы. Для каждой константы заполните свойство Основная форма.

10. В режиме 1С:Предприятие заполните созданные константы по вариантам заданий. Как повлияло свойство Основная форма на способ редактирования констант?

11. Создайте перечисление СтавкиНДС.

Совет. Значения перечисления можно найти в конфигурации Бухгалтерия предприятия учебная (демо). Для этого в режиме Конфигуратор запустите Глобальный поиск по текстам (Правка > Глобальный поиск...).

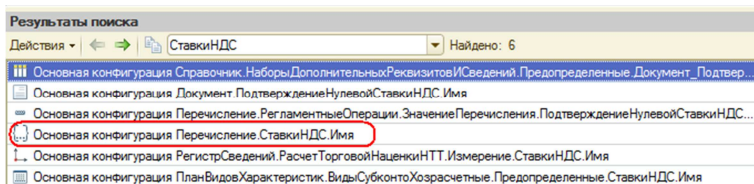


Рис. 2 – Результаты поиска по текстам

12. Создайте перечисления ТипыСкладов, ВидыДоговоров, ВидыЗанятости. Значения перечислений можно найти также в конфигурации Бухгалтерия предприятия учебная (демо).

13. Создайте перечисление ПолФизическогоЛица.

14. Создайте справочник Контрагенты по примеру, рассмотренному в описании лабораторной работы №1.

15. Создайте справочник Договоры. Владелец – справочник Контрагенты (свойство Использование подчинения = Элементам). Реквизиты справочника:

Имя	Тип	Параметры
Дата	Дата	
Номер	Строка	длина = 20, переменная

СрокДействия	Дата	
ВидДоговора	ПеречислениеСсылка. ВидыДоговоров	
Комментарий	Используйте общий реквизит	

Реквизиты Дата, Номер, ВидДоговора сделайте обязательными для заполнения.

16. Создайте справочник ЕдиницыИзмерения. Нужно предусмотреть набор предопределенных основных единиц (на ваше усмотрение).

Реквизиты справочника:

Имя	Тип	Параметры
НаименованиеПолное	Строка	длина = 30, переменная

17. Создайте справочник Номенклатура. Справочник иерархический (иерархия групп и элементов), ограничение количества уровней иерархии снято. Реквизиты справочника:

Имя	Тип	Параметры
НаименованиеПолное	Строка	длина = 1000, перемен.
Артикул	Строка	длина = 25, переменная
ЕдиницаИзмерения	СправочникСсылка. ЕдиницыИзмерения	
СтавкаНДС	ПеречислениеСсылка. СтавкиНДС	значениеЗаполнения = НДС18
Услуга	Булево	
Комментарий	Используйте общий реквизит	

Какие реквизиты целесообразно включить в построение индекса (свойство Индексировать)?

Задействуйте в механизме Ввод по строке созданный реквизит НаименованиеПолное (Способ поиска строки = Любая часть). Какие требования предъявляются к реквизитам, участвующим в Вводе по строке?

18. Создайте справочник Подразделения. Для справочника установите иерархию элементов, без ограничения количества уровней.

Реквизиты справочника:

Имя	Тип	Параметры
НаименованиеПолное	Строка	длина = 250, переменная

19. Создайте справочник ФизическиеЛица. Для стандартного реквизита Наименование установите следующие свойства: Длина наименования = 50, Синоним = ФИО. Добавьте реквизиты:

Имя	Тип	Параметры
ДатаРождения	Дата	
Пол	ПеречислениеСсылка. ПолФизическогоЛица	
ИНН	Строка	длина = 12
СтраховойНомерПФР	Строка	длина = 14

МестоРождения	Строка	длина = 240
Адрес	Строка	длина = 240

Для реквизита СтраховойНомерПФР необходимо предусмотреть ввод в формате 'ЧЧЧ-ЧЧЧ-ЧЧЧ ЧЧ', где Ч – число.

Совет. Решить поставленную задачу можно с помощью свойства Маска реквизита справочника. О том, какие символы может содержать Маска – читайте Справку (по запросу «ПолеВвода.Маска»).

20. Создайте справочник Должности. Длину кода для элементов справочника установите в 0. Реквизиты справочника:

Имя	Тип	Параметры
НаименованиеКраткое	Строка	длина = 25, переменная

21. Создайте справочник Сотрудники с реквизитами:

Имя	Тип	Параметры
ФизическоеЛицо	СправочникСсылка. ФизическиеЛица	
Подразделение	СправочникСсылка. Подразделения	
ВидЗанятости	ПеречислениеСсылка. ВидыЗанятости	
Должность	СправочникСсылка. Должности	
ДатаПриема	Дата	
ДатаУвольнения	Дата	

22. Создайте справочник Склады с реквизитами:

Имя	Тип	Параметры
ТипСклада	ПеречислениеСсылка. ТипыСкладов	
Ответственный	СправочникСсылка. Сотрудники	

Включите использование общего реквизита Комментарий для данного справочника.

23. В режиме 1С:Предприятие заполните справочники согласно вариантам задания правдоподобными сведениями. Количество элементов справочников не менее 5. Обратите внимание, как работают механизмы иерархии, подчинения и ввода по строке. Поэкспериментируйте с предопределенными элементами справочников.

24. Пометьте на удаление несколько элементов справочника Должности. Удалите их, используя стандартный механизм платформы (Все функции... > Стандартные > Удаление помеченных объектов). Если возникают конфликты (см. рисунок ниже), следует найти ссылки на помеченный объект, которые препятствуют его удалению. Для этих

целей освоите еще один стандартный механизм – Поиск ссылок на объекты.

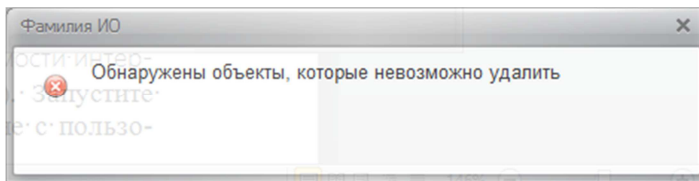
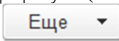


Рис. 3 – Конфликт при удалении помеченных объектов

25. Из режима Конфигуратор поменяйте Режим совместимости интерфейса (свойство корневого элемента дерева метаданных). Запустите приложение, обратите внимание, как выглядит приложение с пользовательским интерфейсом Такси, Версия 8.2 и др.

26. Запустив приложение с интерфейсом Такси, освоите команды Настроить список... и Изменить форму... (командная панель формы списка любого справочника > кнопка ).

27. Произведите архивирование (выгрузку) рабочей информационной базы (команда Администрирование > Выгрузить информационную базу...).

Дополнительные задания.

28. Модифицируйте структуры данных согласно представлениям формирования нормализованных БД и логики функционирования предприятия. В отчёте указать изменённые/добавленные данные и причину изменения.

Варианты выполнения лабораторной работы (тип предприятия)

№ вар	Вариант задания (профиль предприятия)
1	Автосервис (производство запчастей, ремонт и модернизация автомобилей, продажа запчастей)
2	Жилищно-эксплуатационная организация (производство деталей, ремонт, продажа оборудования для ЖКХ)
3	Издательство (производство печатной продукции, её ремонт (переплёт), продажа и торговля своей продукции)
4	Лесохозяйственное предприятие (рубка и посадка леса, первичная переработка древесины, продажа сырого леса и брёвен, торговля)
5	Общественное питание (производство блюд, ресторан, торговля продуктами питания)
6	Производственно-торговое предприятие (производство железобетонных изделий для строительства многоэтажных зданий)
7	Производственно-торговое предприятие (производство стеновых материалов для индивидуального жилищного строительства)
8	Производственно-торговое предприятие (производство кровельных материалов для строительства)
9	Производственно-торговое предприятие (производство материалов деревообработки для строительства)
10	Производственно-торговое предприятие (производство отделочных материалов для строительства)
11	Производственно-торговое предприятие (производство металлических и пластиковых окон и дверей)
12	Машиностроительный завод (производство и торговля спортивным инвентарём)
13	Машиностроительный завод (производство и торговля газового оборудования)
14	Машиностроительный завод (производство и торговля сантехнического оборудования)
15	Машиностроительный завод (производство и торговля садово-огородного инвентаря)
16	Машиностроительный завод (производство и торговля сварочного оборудования)
17	Предприятие пищевой промышленности (производство и торговля мясопродуктами)
18	Предприятие пищевой промышленности (производство и торговля молочными продуктами)
19	Предприятие пищевой промышленности (производство и торговля кондитерскими изделиями)
20	Производственно-торговое предприятие (производство и торговля парфюмерно-косметическими изделиями)

Лабораторная работа №2

РАЗРАБОТКА КОНФИГУРАЦИИ: ОПРЕДЕЛЕНИЕ ПРАВ ДОСТУПА К ФУНКЦИОНАЛЬНОСТИ СИСТЕМЫ, НАСТРОЙКА ИНТЕРФЕЙСНЫХ МЕХАНИЗМОВ, СРЕДСТВА ИНТЕГРАЦИИ

Цель работы: освоение основных приемов работы с системой 1С:Предприятие, правил работы с формами, знакомство с назначением ролей и прав доступа, освоение методов проектирования интерфейса пользователя, изучение средств интеграции и взаимодействия с другими программными системами.

Теоретические сведения

В предыдущей лабораторной работе были созданы константы, определяющие основные данные организации, а также ряд перечислений и справочников. Разработаем интерфейсную часть для ввода данных в реализованные структуры.

Концепция управляемого приложения и управляемого интерфейса

Начиная с версии 8.2 появляются два режима запуска 1С:Предприятие: Управляемое приложение и Обычное приложение. Режим запуска задается через свойство конфигурации Основной режим запуска. Другое свойство, Режим совместимости, устанавливает возможность использования механизмов указанной в свойстве версии платформы (а также предшествующих ей версий).

Замечание. Тонкий клиент и веб-клиент не могут быть запущены в режиме обычного приложения.

Замечание. Запрещена комбинация значений свойств конфигурации Основной режим запуска = Управляемое приложение и Режим совместимости = Версия 8.1.

Замечание. Включение режима совместимости с версией 8.2.13 и ниже несовместимо с наличием в конфигурации общих реквизитов.

Принципиальные отличия между управляемым и обычным приложением состоят в следующем:

- для управляемого приложения предусмотрено либо физическое разделение на сервер и клиент, либо программная эмуляция;
- в управляемом приложении для просмотра, ввода и обработки данных используются управляемые формы, в обычном приложении — обычные формы. Управляемые и обычные формы имеют разные принципы построения и функционирования;

– концепция построения пользовательского интерфейса для управляемого и обычного приложения координально отличаются.

Интерфейс обычного режима предполагает стандартную работу с системой, как в предыдущих версиях (ниже 8.2).

Концепция пользовательского интерфейса в системе 1С:Предприятие неоднократно претерпевала изменения.

Пользовательский интерфейс – средство доступа пользователя к функциональности приложения, средство, позволяющее перемещаться между формами и выполнять различные действия.

Обычный интерфейс. В версиях платформы ниже 8.2 для создания пользовательских интерфейсов имеется специальный объект конфигурации Интерфейс. С его помощью все действия и объекты, к которым пользователь должен иметь доступ, можно сгруппировать в выпадающие списки, часть команд – продублировать в командных панелях. Если конфигурация включает несколько сотен объектов, для разработчика отразить вручную попиксельно каждый из них в интерфейсах весьма трудоемко. Разработка обычного интерфейса ведется следующим образом: интерфейс «рисует» отдельно, права на доступ к объектам настраиваются отдельно. Поэтому вероятно возникновение ситуации, при которой объект обозначен в интерфейсе, но права на доступ к нему отсутствуют. Пользователь, в свою очередь, не имеет гибкого механизма изменения такого интерфейса (только возможность переключаться между интерфейсами).

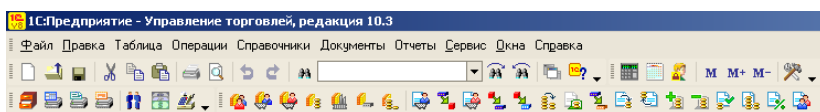


Рис. 12 – Обычный интерфейс (платформа 8.1)

С появлением платформы 8.2 механизм формирования пользовательского интерфейса становится более гибким. Для обозначения нового интерфейса введен термин *командный интерфейс, управляемый интерфейсом*, который нет необходимости прорисовывать в деталях.

Командный интерфейс – это основное средство навигации пользователя по функциональности конфигурации. В системе 1С:Предприятие он строится на основе объектов Подсистемы. Разработчик должен создать в конфигурации иерархию подсистем, отражающую для пользователя структуру функциональности прикладного решения. Все прикладные объекты конфигурации (константы, справочники, документы, отчеты и т. д.) привязываются разработчиком конфигурации к подсистемам. Иерархия подсистем должна отражать структуру функциональности именно так, как она будет представляться пользователю.

Уже на основе этой информации (структуры подсистем и привязки объектов к подсистемам) платформа автоматически строит командный интерфейс для пользователя. Пользователю отображается структура прикладного решения (иерархия подсистем) и предоставляются стандартные команды доступа к функциональности прикладных объектов (вызов списков справочников, документов, открытие отчетов, обработка и т. д.). Однако разработчик, разумеется, может отредактировать предлагаемое системой построение командного интерфейса (изменить порядок, видимость команд). Для этого предназначен целый ряд редакторов командного интерфейса, которые позволяют настраивать различные части интерфейса программы. Стандартной возможностью, которая предоставляет система пользователю, является настройка внешнего вида командного интерфейса (отсюда термины *управляемое приложение*, *управляемый интерфейс*, *управляемые формы*).

Сами команды, включаемые в командный интерфейс (открытие списков, ввод новых объектов, открытие отчетов и т. д.), предоставляются системой автоматически. Но разработчик может создать свои команды, которые будут включаться в командный интерфейс.

Важно! Платформа с версии 8.2 автоматически учитывает при построении интерфейса права пользователя, заданные с помощью объекта конфигурации Роли. В соответствии с этим, из командного интерфейса исключаются команды для доступа к объектам, которые пользователю недоступны. Поэтому говорят, что командный интерфейс «порождается» из прав пользователей на доступ к объектам конфигурации.

В версии платформы 8.3 интерфейсные механизмы вновь претерпевают изменения, развиваясь в русле концепции командного и управляемого интерфейса. Основные преимущества нового интерфейса Такси:

- улучшение возможностей навигации по приложению;
- настраиваемое пользователем рабочее пространство (по части взаимного расположения панелей и их состава команд);
- улучшение юзабилити отдельных элементов интерфейса;
- современный дизайн.

С версии платформы 8.3 появилась возможность настраивать Режим совместимости интерфейса (одноименное свойство конфигурации). Свойство определяет, какой интерфейс по умолчанию использует клиентское приложение и возможность/невозможность изменения этого режима средствами встроенного языка. Например, при режиме совместимости Такси. Разрешить Версия 8.2 клиентское приложение по умолчанию использует интерфейс Такси, но возможно изменение режима средствами встроенного языка. Для режима совместимости Версия 8.2

клиентское приложение работает в интерфейсе версии 8.2, и изменение режима не возможно. Наличие режимов совместимости позволяет выполнять постепенный перевод конфигураций на новые версии.

В связи с тем, что развитие технологической платформы 1С:Предприятие 8.3 связано с охватом мобильных устройств, свойства конфигурации пополнились пунктом Назначения использования. Свойство определяет, что конфигурация может быть использована на ПК и/или мобильном устройстве.

Замечание. При разработке конфигурации под мобильное устройство доступны не все классы объектов (подробнее о мобильной платформе 1С:Предприятия 8 http://v8.1c.ru/overview/Term_000000818.htm).

Подсистемы как составляющая командного интерфейса

Управляемый интерфейс пользователя Такси, который генерируется платформой на основе имеющихся объектов, созданных в рамках 1-й лабораторной работы, требует доработки.

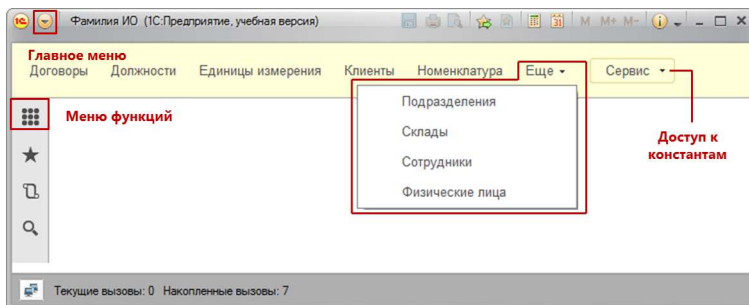


Рис. 13 – Пользовательский интерфейс Такси с настройками по умолчанию (для ПК)

Средства платформа 1С:Предприятие 8.3 позволяют настроить интерфейс из пользовательского режима (с помощью команд из Главное меню > Вид и Меню функций (Настройка навигации, Настройка действий). Но значительных изменений в интерфейсе можно добиться только из Конфигуратора.

Объекты, влияющие на пользовательский интерфейс приложения, сгруппированы в ветви Общие дерева конфигурации. Основой построения интерфейса являются объекты Подсистемы. Кроме этого, для настройки пользовательского интерфейса служат Группы команд, Элементы стиля, Общие картинки.

Подсистемы имеют разное назначение в обычном и управляемом приложении. В первом случае подсистемы позиционируются как инструмент разработчика: распределив объекты по подсистемам, можно

включить в дереве конфигурации отбор по подсистемам, тем самым скрыв ряд объектов. В управляемом приложении подсистемы определяют структуру функциональности прикладного решения именно так, как она будет представляться пользователю.

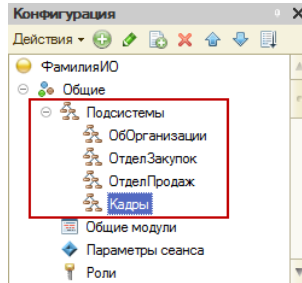


Рис. 14 – Подсистемы в конфигурации

Подсистемы могут иметь дочерние элементы (количество уровней иерархии подсистем ограничено только здравым смыслом). Основные свойства подсистем – флаг Включать в командный интерфейс и Состав. Если флаг снят, то подсистема не будет отображаться в пользовательском интерфейсе и останется лишь контейнером для группировки объектов в Конфигураторе. Для разрабатываемой конфигурации определим следующие подсистемы:

Таблица 15. Перечень подсистем для разработки учебной конфигурации

Подсистема	Состав подсистемы
ОбОрганизации	Константы, форма констант
ОтделЗакупок	Справочники Контрагенты, Договоры, ЕдиницыИзмерения, Номенклатура, Склады
ОтделПродаж	Справочники Контрагенты, Договоры, ЕдиницыИзмерения, Номенклатура, Склады
Кадры	Справочники Подразделения, ФизическиеЛица, Должности, Сотрудники

Если объект не включить ни в одну подсистему, то он не будет отображаться в пользовательском интерфейсе.

Для подсистемы можно установить картинку через одноименное свойство. Выбор картинки производится из стандартного набора или путем добавления внешних графических файлов в состав конфигурации. Картинки, добавленные в состав конфигурации, становятся доступны в пределах любых объектов (не только подсистем), для которых предусмотрено сопоставление им графических изображений. В соответствии с этим принципом добавленные картинки отображаются в ветви Общие > Общие картинки дерева конфигурации.

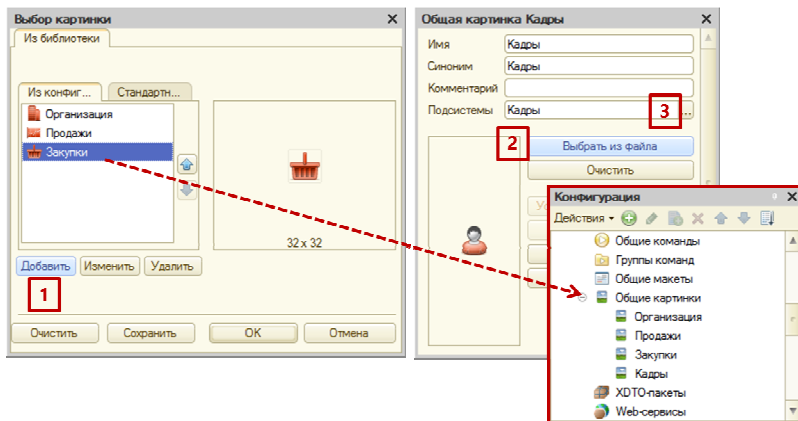


Рис. 15 – Добавление картинки для подсистемы

- 1 – вызов диалога для добавления картинки в состав конфигурации;
- 2 – выбор графического файла с помощью стандартного диалога Windows;
- 3 – поле, связывающее картинку с одной или несколькими подсистемами

Определение прав доступа к объектам конфигурации

Настройка прав доступа к объектам конфигурации является одним из этапов конфигурирования. Для описания полномочия различных категорий пользователей на доступ к информации, обрабатываемой системой, предназначены общие объекты Роли. Как правило, роли создаются отдельно для каждого вида деятельности, и каждому пользователю системы ставится в соответствие одна или несколько ролей. При создании ролей исходят из того, какие полномочия требуются различным группам пользователей на доступ к информации.

Механизм ролей, который изначально (до версии 8.2 платформы) служил только для назначения полномочий пользователям, приобрел в управляемом приложении дополнительный смысл: управление интерфейсом. Другими словами, управляемый интерфейс не прорисовывается разработчиком, а порождается из прав пользователей на доступ к объектам.

Редактор Ролей (вызывается автоматически при создании или редактировании нового объекта Роль) содержит две области: слева, в списке объектов, перечислены все объекты и виды объектов конфигурации, а справа, в окне прав, перечислены доступные права для выбранного объекта или видов объектов конфигурации.

Права бывают двух типов: основные (проверяются всегда, независимо от способа обращения к объекту ИБ) и интерактивные (проверяются при выполнении интерактивных операций, т.е. операций в фор-

мах). Очень аккуратно нужно быть с правом Интерактивное удаление, которое позволяет удалять данные в пользовательском режиме без контроля ссылочной целостности.

Для одной из ролей обязательно следует установить полные права на доступ ко всем объектам и группам объектов конфигурации. Для разрабатываемой конфигурации определим набор ролей в соответствии с таблицей:

Таблица 16. Набор ролей и прав доступа к объектам конфигурации

Роль	Объекты	Права
Администратор	Полные права на все созданные объекты <input checked="" type="checkbox"/> Устанавливать права для новых объектов	
ОтделПродаж	Корневой объект	<input checked="" type="checkbox"/> Тонкий клиент <input checked="" type="checkbox"/> Веб-клиент <input checked="" type="checkbox"/> Режим «Все функции» <input checked="" type="checkbox"/> Сохранение данных польз. <input checked="" type="checkbox"/> Вывод
	Подсистемы ОбОрганизации и ОтделПродаж	<input checked="" type="checkbox"/> Просмотр
	Общий реквизит Комментарий	<input checked="" type="checkbox"/> Просмотр <input checked="" type="checkbox"/> Редактирование
	Общая форма КонстантыОрганизации	<input checked="" type="checkbox"/> Просмотр
	Все константы	<input checked="" type="checkbox"/> Просмотр <input checked="" type="checkbox"/> Чтение
	Справочники Контрагенты, Договоры, ЕдиницыИзмерения, Номенклатура, Склады	Все права, кроме: <input type="checkbox"/> Интерактивное удаление <input type="checkbox"/> Интерактивное удаление предопределенных
ОтделЗакуп	Настройки аналогично роли ОтделПродаж	
Бухгалтерская	Корневой объект	Аналогично роли ОтделПродаж
	Подсистемы ОбОрганизации и Кадры	<input checked="" type="checkbox"/> Просмотр
	Общий реквизит Комментарий	<input checked="" type="checkbox"/> Просмотр <input checked="" type="checkbox"/> Редактирование
	Общая форма КонстантыОрганизации	<input checked="" type="checkbox"/> Просмотр
	Все константы	Полные права
	Все справочники	Все права, кроме интерактивного удаления

Для того чтобы иметь возможность отличать друг от друга пользователей, работающих с информационной базой, в системе 1С:Предприятие существует штатный механизм ведения списка пользователей. В этом режиме администратор базы имеет возможность создавать и удалять пользователей системы, назначать им роли и т.д.

Важно! Перед добавлением пользователей необходимо выполнить обновление конфигурации базы данных.

Добавление новых пользователей информационной базы выполняется в окне Список пользователей (меню Администрирование > Пользователи). Диалог для создания/изменения пользователя содержит поля Имя пользователя (это идентификатор, который будет появляться в окне выбора пользователей при запуске системы в режиме 1С:Предприятие); Полное имя (строка, которая может быть использована внутри конфигурации при выводе различной справочной информации., как правило – ФИО без сокращений); область со способами аутентификации пользователя. На второй вкладке для пользователя можно назначить одну или несколько ролей. Для пользователя, которому сопоставлено несколько ролей, интерфейс строится исходя из того, что права из разных ролей объединяются.

Важно! Первый пользователь ИБ обязательно должен иметь роль с правами на административные функции, иначе он не будет создан. Аналогично, нельзя удалить пользователя с административными функциями, если у всех остальных этих прав нет.

Для разрабатываемой конфигурации создадим следующих пользователей:

Таблица 17.

Имя пользователя	Назначенные роли
Администратор (ФамилияИО)	Администратор
МенеджерПоЗакупкам	ОтделЗакупок
МенеджерПоПродажам	ОтделПродаж
ГлавныйБухгалтер	Бухгалтерская
Директор	Бухгалтерская

После сохранения конфигурации и перезапуска приложения появится запрос имени пользователя и пароля.

С паролями следует быть осторожнее: вся информация о пользователях хранится в общем файле базы данных. В отличие от версии 7.7, где все пароли пользователей убирались простым удалением каталога usrdef, в версии 8 это не получится.

Важно! Учебная версия платформы 1С:Предприятие не рассчитана на работу с паролями пользователей.

Конструирование пользовательского интерфейса

Добавление в конфигурацию подсистем приводит к появлению в пользовательском интерфейсе панели разделов. Заметим, разделы однозначно соответствуют первому уровню подсистем конфигурации.

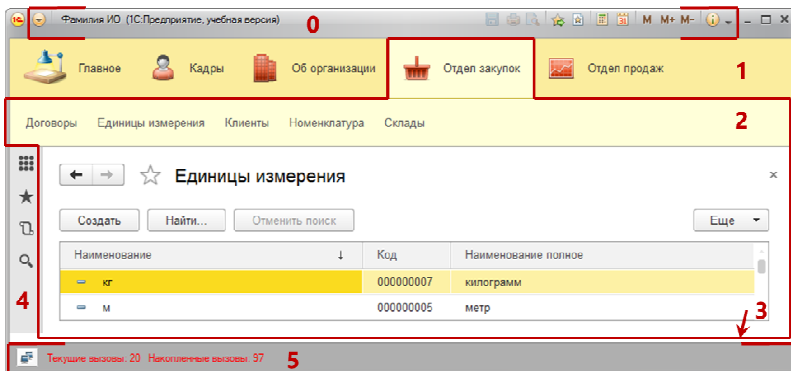


Рис. 16 – Панели интерфейса Такси

- 0 – область системных команд; 1 – панель разделов;
 2 – панель команд текущего раздела; 3 – активный раздел Отдел закупок;
 4 – панель инструментов; 5 – информационная панель

Первым в панели разделов расположен раздел **Главное**. Это аналог рабочего стола в операционных системах, предназначен для размещения наиболее важной, часто используемой информации.

По умолчанию подсистемы в панели разделов и команды в панели текущего раздела располагаются в алфавитном порядке. Изменить порядок на произвольный можно как из конфигуратора через командный интерфейс, так и в пользовательском режиме командами:

- Открыть командный интерфейс конфигурации (в режиме Конфигуратор, для изменения порядка следования разделов);

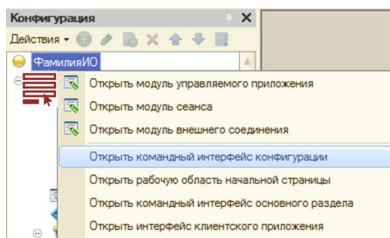


Рис. 17 – Доступ к командному интерфейсу конфигурации

- Командный интерфейс подсистем (в режиме Конфигуратор, для изменения порядка следования команд внутри раздела, соответствующего подсистеме);

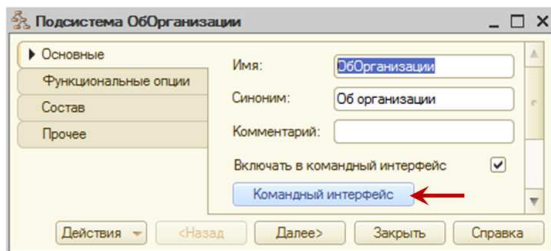


Рис. 18 – Доступ к командному интерфейсу отдельной подсистемы

- Главное меню > Вид > Настройка панели разделов... (в режиме 1С:Предприятие для изменения порядка следования разделов);
- Меню функций > Настройка навигации (в режиме 1С:Предприятие для изменения порядка следования команд текущего раздела).

Взаимное расположение панелей в пользовательском интерфейсе также можно поменять (из Конфигуратора – командой Открыть интерфейс клиентского приложения контекстного меню корневого элемента дерева метаданных; из режима 1С:Предприятие – командой Главное меню > Вид > Настройка панелей...).

Настройки командного интерфейса и расположения панелей, выполненные в Конфигураторе, становятся стандартными настройками всех пользователей. Далее в режиме 1С:Предприятие каждый пользователь может самостоятельно конструировать своё рабочее пространство, изменяя стандартные настройки (к ним всегда можно вернуться).

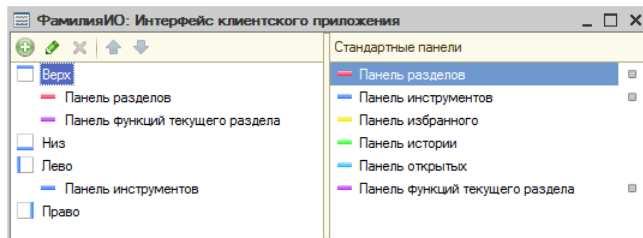


Рис. 19 – Редактор интерфейса со всеми стандартными панелями (инструмент разработчика)

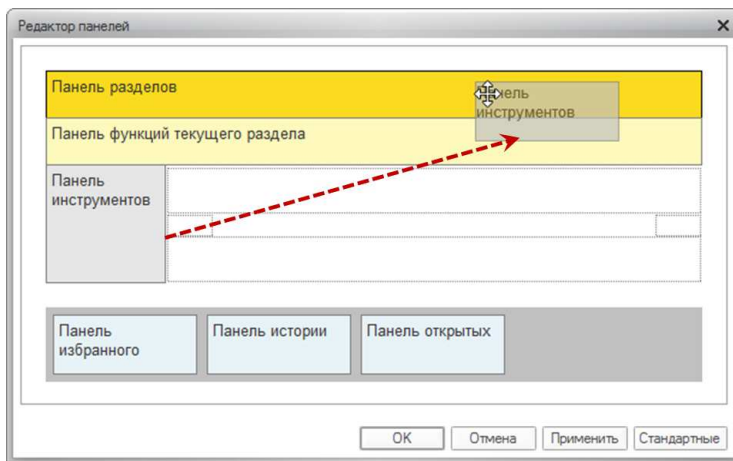


Рис. 20 – Редактор панелей (инструмент пользователя)

Команды и группы команд

Обратимся к командному интерфейсу любой подсистемы. Для него характерно наличие стандартных групп команд (Важное, См. также, Создать, Отчеты, Сервис), внутри которых помещены стандартные команды (по созданию элементов и групп справочников, по открытию списков элементов справочников). Отметим, что с помощью объекта Группы команд (ветвь Общие дерева метаданных) к перечню стандартных групп можно добавить и свои.

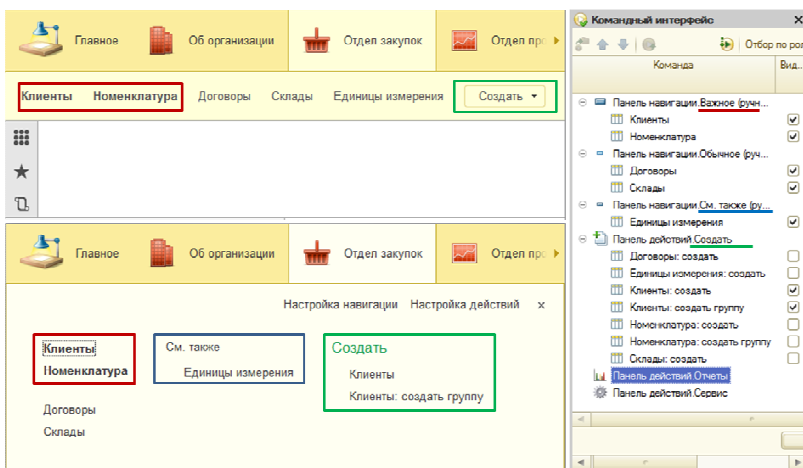


Рис. 21 – Стандартные группы команд в пользовательском интерфейсе

Перемещение команд между группами команд позволяет влиять на вид пользовательского интерфейса.

Далее целесообразно рассмотреть несколько признаков, по которым можно классифицировать команды в системе 1С:Предприятие 8.

По отношению к тому, каким образом команда появилась в конфигурации, все команды можно разделить на стандартные и созданные разработчиком. Стандартные команды поставляются автоматически, платформой. Поставлять стандартные команды могут объекты конфигурации. Например, по рис. 21 видно, что справочник Клиенты может поставлять команды Клиенты, Клиент: создать и Клиент: создать группу. Разработчик может сам создать произвольные команды в конфигурации: добавить Общие команды в ветвь Общие, добавить объект Команда в конкретный объект конфигурации или описать команду в виде процедуры на встроенном языке в конкретной форме.

По отношению к тому, что является результатом выполнения команды, все команды можно разделить на навигационные и команды действий. Навигационные команды открывают формы в текущем окне. Как правило, таким образом выполняется переход к каким-либо спискам. Например, навигационная команда Договоры (см. рис. 21) откроет список договоров. В результате выполнения команды действия открывается вспомогательное окно. Как правило, таким образом выполняется переход к форме объекта или к форме отчета/обработки. Например, команда действия Клиенты: создать группу (см. рис. 21) откроет вспомогательное окно для редактирования данных новой группы справочника Клиенты.

Позже, при знакомстве с механизмом работы форм, будет рассмотрен еще один признак классификации команд – по отношению команд к некоторой форме.

Для сравнения интерфейса Такси и интерфейса версии 8.2 покажем вид клиентского приложения с включенным режимом совместимости (Основной режим запуска = Управляемое приложение, Режим совместимости интерфейса = Версия 8.2, Режим совместимости = Версия 8.2.16). Для пользовательского интерфейса версии 8.2 предусмотрена настройка панели разделов, панели навигации, панели действий – можно изменить порядок, видимость команд (через контекстное меню соответствующей стандартной панели). Взаимное положение стандартных панелей жестко зафиксировано на уровне платформы (не может быть изменено ни пользователем, ни разработчиком), варьируется их ширина.

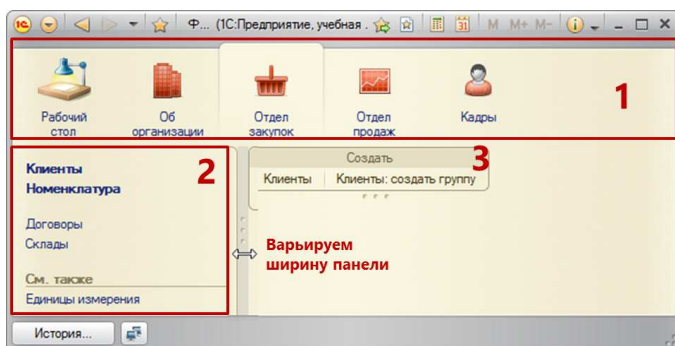


Рис. 22 – Вид управляемого интерфейса для версии 8.2

1 – панель разделов; 2 – панель навигации; 3 – панель действий

Понятие «форма»

Форма – это специально созданный объект для ввода и просмотра какой-либо информации, а также для выполнения управления различными процессами. Другими словами, форма – интерфейсный объект, посредством которого пользователь взаимодействует с данными информационно-аналитической базы.

Форма может быть подчиненной объекту конфигурации, либо общей (например, форма констант, располагающаяся в ветви Общие > Общие формы дерева метаданных). Создание форм выполняется в Конфигураторе с помощью одноименного конструктора. На этапе конструирования формы задается ее тип, который определяет дальнейшее поведение формы по умолчанию.

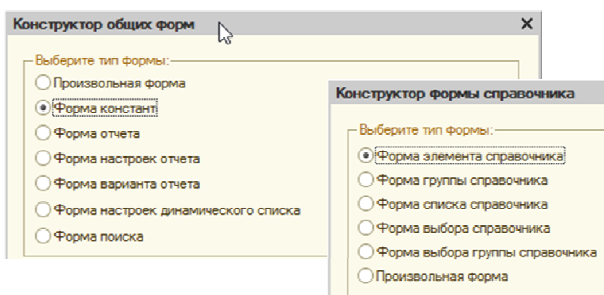


Рис. 23 – Типы общих форм и форм, подчиненных справочнику

На втором шаге в конструкторе следует указать набор реквизитов, которые будут отображаться на форме.

Все дальнейшие манипуляции выполняются в редакторе формы. При создании новой формы редактор вызывается автоматически (по-

сле завершения работы конструктора). Редактор для существующей формы вызывается при ее открытии.

Редактор управляемой формы представлен несколькими вкладками.

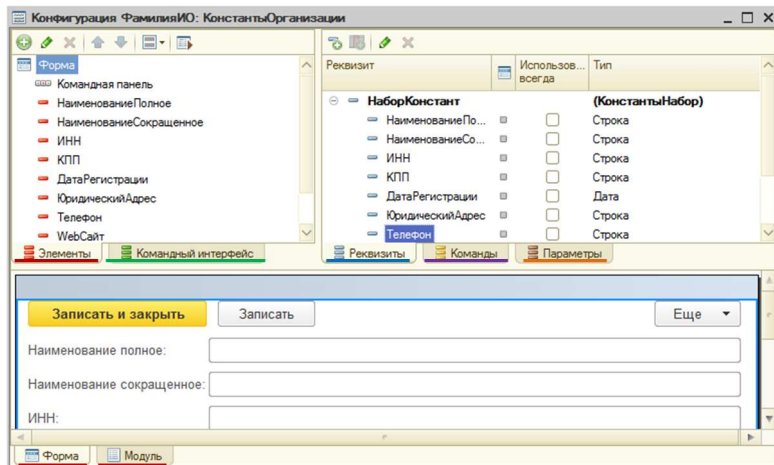


Рис. 24 – Редактор управляемой формы

Реквизиты. Вкладка содержит список реквизитов формы – объектов различных типов, принадлежащих форме. В реквизитах формы содержатся данные (реквизиты составляют контекст формы). Другими словами, реквизиты формы используются, чтобы связать форму с данными объекта, для которого форма конструируется. Жирным шрифтом выделяется так называемый Основной реквизит формы. Он определяет основную функциональность формы и некоторые автоматические действия, которые будет делать платформа относительно этой формы. Основной реквизит мы определяем при выборе типа формы в конструкторе. Соответственно, для разных типов форм имеют место разные основные реквизиты. В нашей учебной конфигурации на данный момент имеется форма с типом Форма констант.

Элементы. Вкладка содержит дерево элементов управления, каждый из которых имеет связь с одним из реквизитов формы. Таким образом, элементы управления определяют способ отображения данных, содержащихся в реквизитах формы. Любая управляемая форма модифицируется путем добавления элементов в дерево, перетаскивания их в пределах дерева. Можно воспользоваться следующими элементами управления: Группа, Поле, Кнопка, Таблица, Декорация (надпись, картинка). Элемент управления характеризуется набором свойств и событий, настраиваемых в панели свойств [Atl+Enter].

Замечание. Если элемент управления отображается в дереве элементов, но отсутствует на форме, то элементу не был сопоставлен Путь-КДанным. В таком случае элементу попросту нечего отображать на форме.

Форма. Вкладка представляет собой область предварительного просмотра и отображает форму так, как ее увидит пользователь.

Модуль. Программный модуль, содержащий процедуры-обработчики событий формы и функции, написанные на встроенном языке системы 1С:Предприятие.

Формы, наряду с конфигурацией и подсистемами, характеризуются командным интерфейсом, который настраивается на одноименной вкладке редактора формы. Через эту вкладку можно модифицировать панель навигации формы и ее командную панель.

Таким образом, вид управляемых форм зависит от структуры дерева элементов управления и структуры прав доступа. При этом форма является не графическим объектом, а декларативным описанием в виде дерева. В связи с этим для управляемых форм не действуют абсолютные координаты; ширина и высота элементов управления задается не в пикселях, а в символах; положение элементов определяется относительно друг друга. Все эти аспекты облегчают передачу управляемых форм по слабым каналам связи.

Совет. Для изменения взаимного расположения элементов на созданной форме нужные элементы необходимо объединить с помощью элемента управления Группа, для Группы настроить свойства Поведение, Отображение, Группировка.

Совет. Чтобы посмотреть на создаваемую форму в рабочем режиме, можно обойтись без сохранения конфигурации базы данных и запуска приложения в режиме отладки. Для тестирования/проверки формы (в Конфигураторе) воспользуйтесь сочетанием клавиш [Ctrl+R].

Самый быстрый способ удалить форму (и любой другой объект конфигурации) – клавиша [Delete]. Однако нужно помнить: при удалении любого объекта конфигурации происходит автоматический контроль ссылочной целостности, и система не даст удалить объект, на который ссылаются другие объекты конфигурации. Если сейчас попытаться удалить имеющуюся в конфигурации форму констант, платформа проинформирует, что «Объект не может быть удален, так как на него имеются ссылки в других объектах». И это верно, ведь ссылка на форму констант содержится в свойстве Основная форма всех констант. Поэтому перед удалением любого объекта конфигурации рекомендуется проверить наличие ссылок на него командой Поиск ссылок на объект (Поиск ссылок в объекте) из контекстного меню объекта.

Важной особенностью системы 1С:Предприятие 8 является механизм форм по умолчанию. Этот механизм освобождает разработчика от необходимости создания всех возможных форм для каждого из объектов прикладного решения. Разработчику достаточно создать новый объект прикладного решения, а система сама сгенерирует в нужные моменты работы пользователя необходимые формы по умолчанию для отображения информации, содержащейся в этом объекте. Таким образом, разработчику нужно создавать собственные формы объектов прикладного решения лишь в том случае, если они должны иметь отличия (другой дизайн или специфическое поведение) от форм, генерируемых системой по умолчанию. В предыдущей работе мы использовали именно формы по умолчанию (например, формы типа Список для отображения элементов справочника). Формы, создаваемые платформой «на лету», вполне устраивали нас для заполнения созданных структур (справочников) данными и их просмотра. Если требуется организовать более сложные решения, то необходимо создать требуемый вид формы и произвести её редактирование.

Настройка форм справочников

Разработаем экранные управляемые формы для заполнения реквизитов справочника, а также формы для просмотра списка элементов справочника (на примере справочника Контрагенты).

Откроем в конфигураторе справочник контрагентов и перейдём на вкладку Формы мастера объекта.

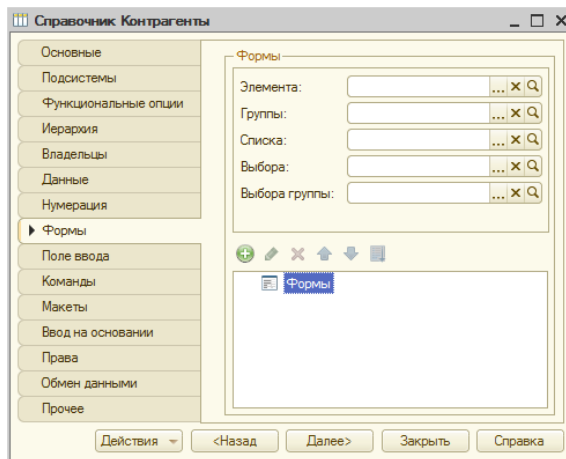


Рис. 25 – Вкладка Формы мастера справочника

Вкладка *Формы* справочника разделена на две области. В нижней области содержится список форм разных типов, подчиненных данному справочнику. В верхней области назначаются основные формы каждого типа: основная форма элемента, основная форма списка и т.д. (выбор осуществляется из нижнего списка). Назначить форму основной значит указать платформе, что именно эту форму нужно показывать пользователю. Если основную форму не назначить, то система создает ее «на лету» в процессе работы пользователя (см. конец раздела *Понятие «форма»*).

Создадим форму элемента справочника. Формы создаются в нижней части окна, методика такая же, как и при создании реквизитов. По нажатию кнопки *Добавить* (из командной панели нижней области окна) запускается конструктор формы справочника.

Для справочника предусмотрены формы 6-ти типов (элемента, группы, списка, выбора, выбора группы, произвольная). В зависимости от выбранного типа формы некоторые поля настройки могут становиться недоступными. Выбираем тип *Форма элемента справочника*. Вариант *Произвольная форма* отличается от других тем, что не имеет Основного реквизита, который определяет основную функциональность формы. Если установить флаг *Назначить форму основной*, она после создания автоматически будет присвоена соответствующему полю в списке основных форм. По флагу *Основная форма элемента и группы* данная форма станет основной формой и для элемента, и для группы иерархического справочника (продублируется).

Если в настройках параметров (*Сервис > Параметры > Общие*) указан режим редактирования конфигурации *Управляемое приложение* и обычное приложение, то в конструктор формы добавляется еще одно поле выбора типа создаваемой формы (*управляемая или обычная*). При создании обычной формы становятся доступны флажки для размещения на форме командных панелей.

Нажимаем кнопку *Далее* для настройки списка реквизитов, или *Готово*. Настройка списка реквизитов нужна тогда, когда требуется изменить состав реквизитов, отображаемых на форме. При создании обычной формы на этом этапе доступно изменить тип элемента управления формы, связанного с реквизитом. По умолчанию, создается элемент *Поле ввода*, но можно поставить переключатель, флажок, надпись и т.п. в зависимости от типа данного реквизита.

В открывшемся редакторе формы можем настроить ее внешний вид, изменив состав дерева элементов управления и их свойства.

Важно! В версии 8 есть возможность изменить родителя элемента справочника непосредственно в форме как реквизит, в 1С:Предприятии 7.7 для этого пришлось бы писать программный код.

Ранее отмечалось, что элементы управления выполняют свою задачу только в случае, если установлена связь между элементом управления и реквизитом формы. Для этого предназначено свойство ПутьКДан-ным элемента управления.

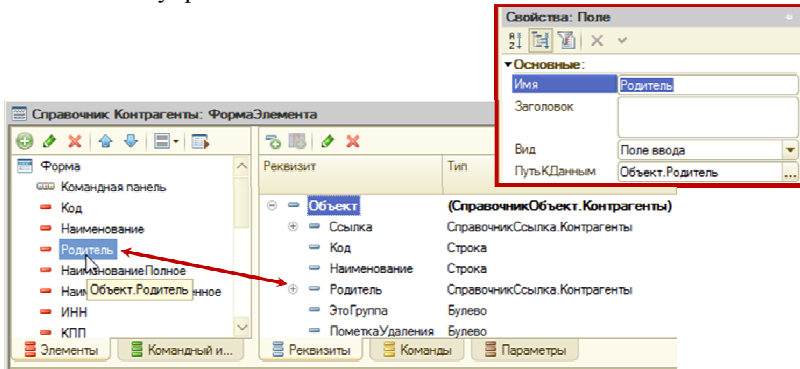


Рис. 26 – Иллюстрация связи элемента управления с реквизитом формы

Отредактируем стандартный вид формы элемента справочника. Разместим элементы более компактно, разнеся данные о клиенте и его контактах по разным вкладкам. Для решения задачи дерево элементов управления должно иметь следующую структуру:

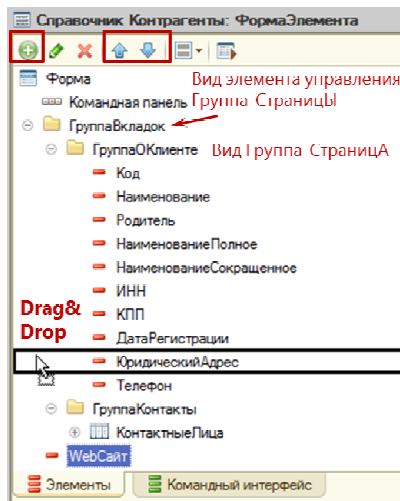






Рис. 27 – Иллюстрация связи элемента управления с реквизитом формы

Элементы управления распределяются по вкладкам с помощью вспомогательных элементов Группа-Страницы и Группа-Страница. Все типы элементов управления доступны по кнопке Добавить  (командная панель вкладки Элементы). Перемещение элемента в группу осуществляется по методу Drag&Drop. Порядок элементов внутри группы можно изменить с помощью управляющих стрелок  и .

Ни разработчику, ни пользователю нет необходимости заботиться о том, как на программном уровне создаются элементы и группы справочников, происходит открытие списков элементов справочников и запись их в базу данных. Все эти и другие команды поставляются платформой автоматически. Для изменения стандартного поведения объектов, форм предназначен механизм обработки событий и встроенный язык программирования системы 1С:Предприятие. Познакомимся с этими механизмами, организовав в разработанной форме автоматический запуск почтовой программы для отправки письма контактному лицу клиента.

События – это определенные изменения или действия, произошедшие в системе, с формой или определенным элементом формы. Например: запуск системы, открытие формы, нажатие кнопки, обновление и т.п. События для каждого объекта системы заранее жестко определены на уровне платформы.

Список предопределенных событий для любого элемента управления приведен в его палите свойств (категория События). Кнопка , расположенная справа от поля с наименованием события, служит для создания / просмотра процедуры-обработчика этого события. Нажатие на «лупу» запускает конструктор создания обработчика события (если процедура-обработчик события еще не создана), либо позволяет перейти в модуль формы.

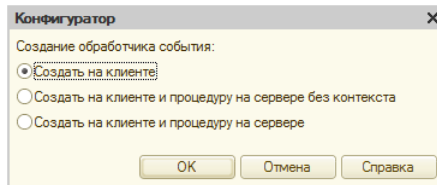


Рис. 28 – Создание обработчика события

Отметим, что конструктор создания обработчика события появился с версии 8.3 платформы. Ранее в модуле формы автоматически создавалась одна процедура, выполняющаяся на клиенте.

После выбора одного из вариантов в модуль формы добавляется каркас процедур(ы)-обработчика события.

При использовании конструктора название процедуры-обработчика события формы складывается по шаблону «Имя элемента управления+Имя события», но может быть и произвольным.

Описание каждой процедуры/функции в модуле формы предваряется *директивой компиляции* (&НаКлиенте, &НаСервере, др.). Эта конструкция определяет место выполнения программного кода:

```
&НаКлиенте
Процедура КонтактныеЛицаАдресЭлПочтыОткрытие(Элемент, СтандартнаяОбработка)
    // Вставить содержимое обработчика.
КонiecПроцедуры
```

Если директива компиляции не указана, по умолчанию считается, что подпрограмма выполняется &НаСервере.

Вернемся к реализации возможности отправки письма контактному лицу клиента. Для этого открываем свойства элемента управления КонтактныеЛицаАдресЭлПочты табличной части КонтактныеЛица и в категории Использование устанавливаем свойство КнопкаОткрытия = Да. Далее создаем обработчик для события Открытие с командой:

```
ЗапуститьПриложение("mailto:" + Элемент);
```

Здесь мы используем переданный параметр – ссылку на элемент управления, по которой он связан с реквизитом табличной части, содержащим данные. Теперь при выборе конкретного e-mail при двойном щелчке мыши будет появляться кнопка открытия, нажав на которую мы запустим почтовый клиент для данного адреса.

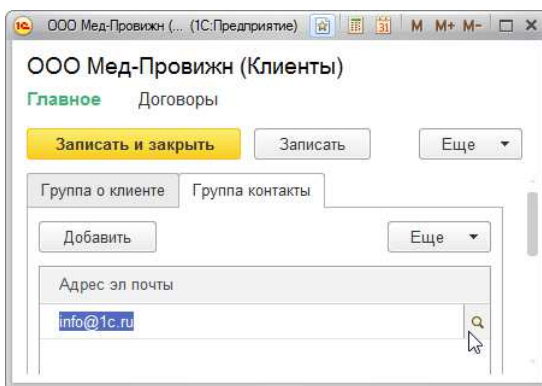


Рис. 29 – Кнопка для запуска почтового клиента

Единственным элементом управления, для которого не предусмотрено списка событий в свойствах, является Кнопка. Для обработки

нажатия элементу Кнопка сопоставляется команда (через свойство элемента управления ИмяКоманды). Команды создаются на одноименной вкладке редактора управляемой формы.

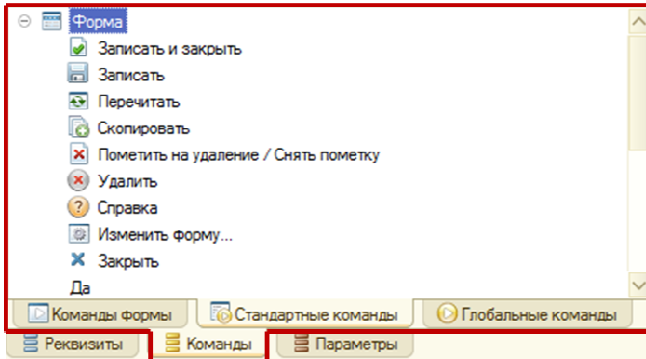


Рис. 30 – Источники команд для кнопок

Судя по структуре вкладки Команды, по отношению команд к некоторой форме, все команды можно разделить на локальные и глобальные. Глобальные команды не являются частью формы, они поставляются платформой и могут быть включены в эту или в другие формы. Локальные команды формы являются частью самой формы и не могут быть включены в другие формы. Локальные команды поставляются формой и ее элементами (Стандартные команды), а также создаются разработчиком в форме (Команды формы).

Если перетащить команду в область дерева элементов управления, то кнопка будет создана автоматически.

Поскольку разработка сложных конфигураций связана с программированием, платформа предоставляет средства отладки программных модулей (в Конфигураторе меню Отладка): точки останова, точки основа с условием, замер производительности и др. Запуск приложения в режиме отладки производится по нажатию [F5]. Режим отладки предусмотрен для всех видов клиентских приложений (меню Отладка > Начало отладки).

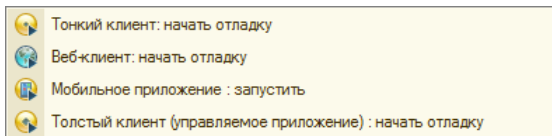


Рис. 31 – Варианты запуска в режиме отладки

Механизм работы управляемых форм

Рассмотрим механизм работы управляемой формы на примере формы элемента справочника. Пользователь дает команду на открытие формы. Далее выполняются следующие действия:

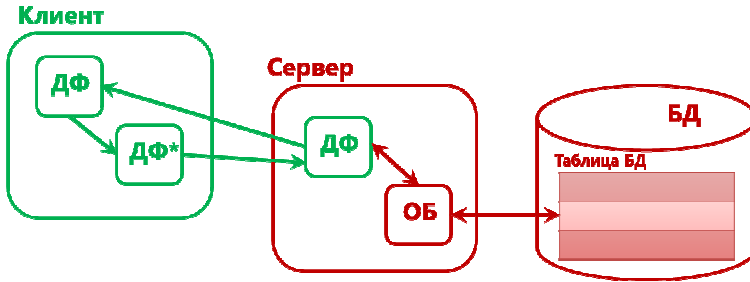


Рис. 32 – Механизм работы управляемой формы

БД – база данных; ОБ – объект БД;

ДФ – данные формы; ДФ* – модифицированные на клиенте данные

Сервер 1С:Предприятие обращается к БД, происходит выборка записи (соответствующей элементу справочника) из таблицы (соответствующей справочнику).

При сборке формы на сервере платформа:

- получает объект БД (который не может существовать на стороне клиента),
- автоматически преобразует ОБ в некое «представление» (ДФ), доступное и на сервере, и на клиенте,
- помещает ДФ в основной реквизит формы элемента справочника.

Преобразование ОБ в ДФ/ДФ в ОБ автоматическое, но есть и явные методы.

Далее копия формы передается на сторону клиента, пользователь изменяет данные на форме. При выполнении команды Записать и Открыть происходит серверный вызов, и форма в своем основном реквизите «приносит» с клиента на сервер все изменения, сделанные пользователем. ДФ* и ДФ синхронизируются. Выполняется автоматическое преобразование ДФ в ОБ. Посредством ОБ изменения сбрасываются в базу.

Важно! Таким образом, форма содержит собственную копию данных из базы (это контекст формы). У любого объекта несколько состояний: на форме на стороне клиента, на форме на стороне сервера, в базе.

При программировании управляемых форм (и разработке управляемого приложения в целом) необходимо помнить, что программный код должен разрабатываться в рамках клиент-серверного подхода.

Как было сказано выше, данные формы содержатся в ее основном реквизите. В модуле формы &НаКлиенте к этим данным можно напрямую обращаться, изменять их. Иначе приходится работать с полями ссылочного типа. Например, на форме элемента справочника Сотрудники поле ФизическоеЛицо имеет тип СправочникСсылка.ФизическиеЛица. Эта ссылка представлена на форме в виде наименования. В модуле формы &НаКлиенте по этой ссылке напрямую мы не сможем получить никаких данных (пол, дату рождения, ИНН и др.), поскольку в реквизите формы есть только ссылка на данные, но не сами данные. Данные по ссылке можно получить, обратившись на сервер, например, так:

```
&НаКлиенте
Процедура ФизическоеЛицоПриИзменении(Элемент)
    Сообщить(ПолучитьИнфоНаСервере(Объект.ФизическоеЛицо));
КонецПроцедуры

&НаСервереБезКонтекста
Функция ПолучитьИнфоНаСервере(ссылкаНаФизлицоСФормы)
    Возврат ссылкаНаФизлицоСФормы.МестоРождения;
КонецФункции
```

Разберем приведенный листинг.

Клиентская процедура – это обработчик события ПриИзменении элемента ФизическоеЛицо. В этой процедуре вызывается встроенная функция Сообщить (выводит сообщение пользователю) с параметром, содержащим ссылку, взятую из реквизита формы. По ссылке на сервере возможно получить место рождения физического лица. Серверная процедура описана директивой &НаСервереБезКонтекста. Это означает, что при ее вызове на сервер передается только управление и параметр процедуры, но не больше (контекст формы не передается на сервер при таком описании).

О разработке обычного интерфейса и обычных форм

Как отмечалось ранее, управляемый интерфейс, появившийся с версии 8.2 системы 1С:Предприятие, формируется следующим образом. Разработчик описывает возможность отображения команд, предоставляемых платформой или созданных самим разработчиком, в заранее определенных элементах интерфейса прикладного решения (имеется в виду панель разделов, панель действий, панель навигации, командные панели форм объектов конфигурации и т.п.). Окончательное же построение конкретного пользовательского интерфейса возлагается на платформу, которая «анализирует» роли, назначенные пользовате-

лю, параметры, настроенные в конфигурации, а также настройки, выполненные самим пользователем, формируя индивидуальный интерфейс с исключением ненужной для данного пользователя функциональности.

Кратко рассмотрим процесс построения обычного интерфейса и обычных форм. Средства работы с обычным интерфейсом и формами становятся доступны при изменении свойства Редактирование конфигурации для режимов запуска с варианта Управляемое приложение на Управляемое приложение и обычное приложение (меню Сервис > Параметры > Общие). Кроме того, следует изменить свойство конфигурации Основной режим запуска на Обычное приложение. После этого, во-первых, в дереве конфигурации появляется дополнительный объект – Интерфейсы. В обычном приложении эти объекты используются для проектирования индивидуальных пользовательских интерфейсов в виде наборов команд главного меню и панелей инструментов.

При таком подходе интерфейсы являются статическими наборами картинок, с которыми связаны те или иные действия. Такой подход к организации интерфейса не позволяет динамически его перестраивать в зависимости функционирования прикладного решения.

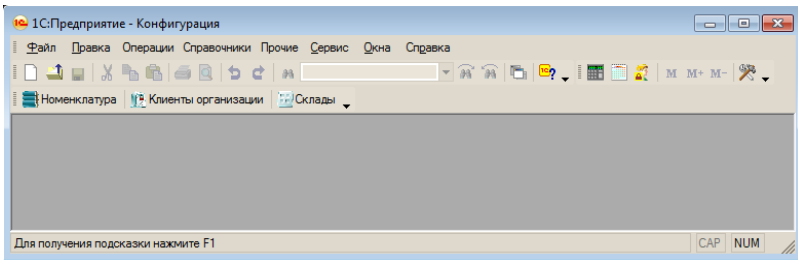


Рис. 33 – Обычный интерфейс

Как правило, для каждой категории пользователей создается свой интерфейс, который ставится в соответствие конкретному пользователю. В отличие от ролей, каждому пользователю можно назначить только один интерфейс по умолчанию, однако средствами встроенного языка можно управлять видимостью других интерфейсов.



Во-вторых, на вкладке Формы мастера справочника появляется область дополнительных форм:

Рис. 34 – Основные и дополнительные формы справочника

В разделе **Формы > Дополнительные** можно назначить обычные формы элемента, группы, списка справочника и т.д., которые будут использоваться в случае запуска конфигурации в режиме обычного приложения.

Редактор обычной формы состоит из трех вкладок: **Форма**, **Модуль**, **Реквизиты**. Элементы управления можно свободно и абсолютно произвольно перемещать (Drag&Drop) непосредственно на вкладке **Форма** редактора:

Рис. 35 – Вид обычной формы элемента справочника Договоры

При добавлении новых элементов на обычную форму можно пользоваться мастером Вставки элемента управления - кнопка  главной панели управления, а также мастером размещения данных - кнопка .

При оформлении дизайна обычной формы очень удобно пользоваться выравнивающими линиями, которые можно настроить, вызвав контекстное меню формы.

В версии 8 появилась возможность настраивать поведение элементов обычной формы при изменении ее размеров. Это делается при по-

мощи механизма привязок. Привязки конкретного элемента формы настраиваются в его палитре свойств в разделе Расположение > Привязка границ.

Механизмы интеграции

Система 1С:Предприятие – открытая система, может взаимодействовать с другими открытыми системами. Рассмотрим два механизма интеграции: Automation сервер и СОМ-соединение.

Работа системы 1С:Предприятие в качестве Automation сервера. Система 1С:Предприятие может использоваться внешними приложениями в качестве Automation сервера. При работе в качестве Automation сервера 1С:Предприятие предоставляет доступ ко всем свойствам и методам своего глобального контекста, а также включение и выключение пользовательского интерфейса (главного окна 1С:Предприятия 8). Кроме того, Automation сервер 1С:Предприятия имеет дополнительные свойства и методы для выполнения действий, специфичных для работы в режиме Automation. Основное назначение Automation сервера 1С:Предприятие в управлении приложением 1С:Предприятия 8 из других приложений и выполнение действий аналогичных интерактивным действиям, например, построение отчетов.

Для запуска системы 1С:Предприятие в качестве Automation сервера из внешнего приложения, выполняется следующая последовательность действий:

- создается СОМ объект с идентификатором V8.Application;
- выполняется инициализация системы 1С:Предприятие методом Connect;
- вызываются свойства и методы системы 1С:Предприятие как Automation сервера.

Многие современные программные продукты (MS Office, MS FoxPro и т.п.) могут выступать в роли клиентов Automation, что позволяет обмениваться данными между этими продуктами и системой 1С:Предприятие.

Поскольку система 1С:Предприятие может создавать и использовать Automation серверы, предоставляемые внешними приложениями (т.е. являться Automation клиентом), то имеется возможность из 1С:Предприятия обращаться к другой копии 1С:Предприятия (например, к другой конфигурации) для обмена данными.

Нелокализованные версии внешних программ, обращающихся к программе 1С:Предприятие посредством Automation, могут неправильно интерпретировать русские идентификаторы объектов, например, реквизитов справочников. Данное замечание не относится к про-

дуктам MS Office и к программам, использующим в качестве языка обращения к СОМ-объектам Microsoft Visual Basic. Рекомендуется использовать локализованные версии программных продуктов, либо в конфигурации использовать идентификаторы без символов кириллицы. Для обращения к свойствам и методам объектов системы 1С:Предприятие из внешних приложений рекомендуется использовать их англоязычные синонимы.

Пример использования Automation сервера 1С:Предприятия, фрагмент для C++

```
Variant v8=Variant::CreateObject("V8.Application");
v8.OleFunction("Connect", "File=c:\\InfoBases\\trade;
                    Usr='Федоров (администратор)';");
v8.OlePropertySet ("Visible" , true);
Variant pDoc = v8.OleFunction ("NewObject",
"ТекстовыйДокумент");
pDoc.OleFunction ("ВставитьСтроку", 1, "Новая строка");
int Count=pDoc.OleFunction("КоличествоСтрок");
ShowMessage ( "Количество строк = "+(AnsiString)Count);
```

и VBA

```
Dim v8 As Object
Set v8=CreateObject("V8.Application")
v8.Connect("File="&"c:\\InfoBases\\Trade"&";
          Usr="&"Федоров (администратор)"&"");
v8.Visible = True
ТДокумент = V8.NewObject("ТекстовыйДокумент");
ТДокумент.ВставитьСтроку(1, "Новая строка");
Сообщить(ТДокумент. КоличествоСтрок());
```

СОМ-соединение. Основная задача, решаемая с помощью СОМ-соединения – обеспечение надежного и быстрого программного доступа к данным 1С:Предприятия 8 из внешних приложений. В общем и целом работа с 1С:Предприятием 8 через СОМ-соединение подобна работе с 1С:Предприятием в режиме Automation сервера. Основные отличия заключаются в следующем:

- В случае Automation сервера запускается полноценное приложение 1С:Предприятия 8, а в случае СОМ-соединения запускается относительно небольшой внутрипроцессный СОМ-сервер.
- При работе через СОМ-соединение не доступны функциональные возможности так или иначе связанные с организацией пользовательского интерфейса 1С:Предприятия 8;
- При работе СОМ-соединения не используется модуль приложения конфигурации 1С:Предприятия 8. Его роль при работе с СОМ-соединением играет модуль внешнего соединения. Назначение ука-

занных программных модулей будет рассмотрено в материале следующих работ.

При использовании для доступа к данным 1С:Предприятия 8 СОМ-соединения имеются следующие преимущества по сравнению с использованием Automation сервера:

- Более быстрая установка соединения, так как не требуется создания отдельного процесса операционной системы, а все действия производятся в рамках вызывающего процесса.
- Более быстрое обращение к свойствам и методам объектов 1С:Предприятия, так как для организации обращения не требуется организации межпроцессной коммуникации.
- Меньший расход ресурсов операционной системы.

Для организации доступа к данным 1С:Предприятия 8 через СОМ-соединение, выполняется следующая последовательность действий:

- создается СОМ-объект с идентификатором V8.COMConnector, с помощью которого производится установка соединения;
- производится обращение к методу Connect ранее созданного объекта V8.COMConnector. Метод Connect возвращает ссылку на объект СОМ-соединения с информационной базой 1С:Предприятия 8;
- через полученный объект СОМ-соединения производится обращение к допустимым методам, свойствам и объектам информационной базы, с которой установлено соединение.

Замечание. В связи с отсутствием пользовательского интерфейса не все объекты, а также свойства и методы можно использовать в СОМ соединении.

С помощью СОМ-соединения можно, например, организовать веб-доступ к базе данных 1С:Предприятие 8 или обмен XML-документами системы 1С: Предприятие 8 с внешним приложением.

Пример использования аналогичен рассмотренному ранее для Automation сервера, за исключением использования сервера автоматизации с именем «V8.COMConnector».

Содержание работы

Задания *Intermediate*.

1. Добавьте в состав конфигурации подсистемы (по табл. 15).
2. Определите для конфигурации набор ролей и список пользователей. Опирайтесь на таблицы 16, 17.
3. Приведите пользовательский интерфейс к следующему виду:

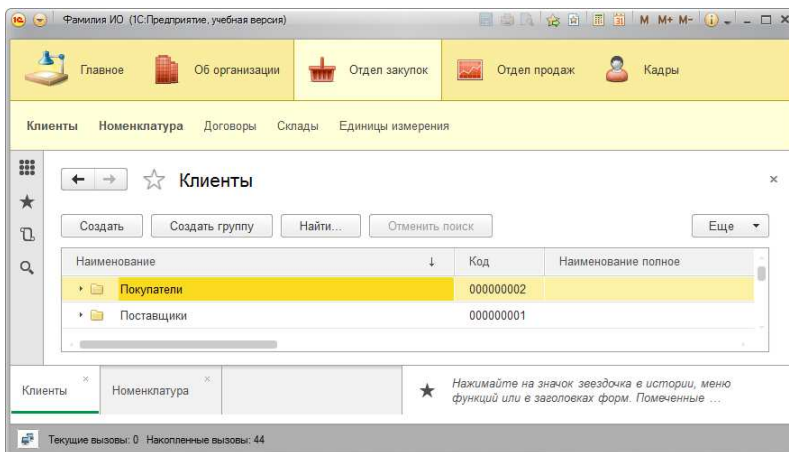


Рис. 36 – Сконструированный интерфейс Такси для пользователя Администратор

Для остальных пользователей настройте интерфейс по своему вкусу.

4. Измените вид «по умолчанию» формы констант. Элементы управления расположите в соответствии с рисунком:

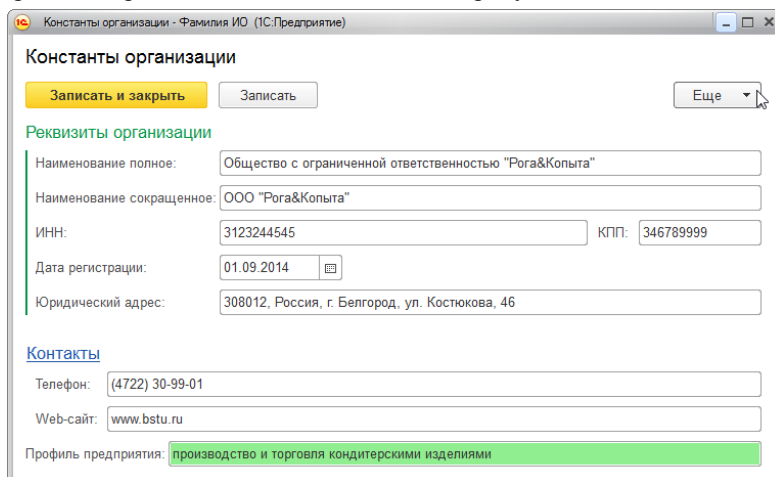


Рис. 37 – Модифицированная форма констант

5. Выполните пример из текста по реализации запуска почтового клиента из строки табличной части КонтактныеЛица элемента справочника Контрагенты. Дизайн формы:

Рис. 38 – Вид формы элемента справочника

Организуите из формы переход на веб-сайт контрагента по нажатию на кнопку (аналогично запуску почтового клиента). Либо можно воспользоваться более сложной конструкцией:

Если НЕ ЗначениеЗаполнено(Объект.WebСайт) Тогда
 Предупреждение ("Укажите веб-сайт!");
 Иначе ЗапуститьПриложение(Объект.WebСайт);
 КонецЕсли;

6. Создайте форму элемента справочника Договоры:

Рис. 39 – Вид формы элемента справочника Договоры

7. Создайте и настройте по своему усмотрению экранные формы (элемента, списка) для остальных справочников, созданных в первой лабораторной работе.

8. Освойте штатный механизм загрузки списка элементов справочника в табличный/текстовый документ (командная панель формы списка справочника > Еще > Вывести список...).

Задания Upper-Intermediate.

9. Создайте и настройте обычные формы для отображения элементов справочников Контрагенты и Номенклатура.

10. Усовершенствуйте программный код по переходу на веб-сайт контрагента. Необходимо предусмотреть информирование пользователя о вводе некорректного веб-адреса (без домена верхнего уровня, www, и др.). Один из вариантов решения – использование регулярных выражений¹.

Задания Advanced.

11. Разработайте программный модуль для импорта информации (список элементов для любого справочника, имеющего в своей структуре реквизиты ссылочного типа) из файла Excel в учебную информационную базу с использованием 1С:Предприятия в качестве сервера Automation или с помощью COM-соединения.

¹ <http://novichok1c.ru/ispolzovanie-regulyarnyh-vyrazheniy-v-1s> - о регулярных выражениях в системе 1С:Предприятие

Контрольные вопросы

1. В чем отличие управляемого приложения и обычного приложения?
2. Что такое командный интерфейс?
3. Каково назначение подсистем в управляемом и обычном приложении?
4. Что такое Роли?
5. Для чего используется список пользователей и как происходит его конфигурирование?
6. Управляемый интерфейс Такси: стандартные панели и их настройка.
7. Что такое форма?
8. В чем отличие управляемых и обычных форм?
9. Каков принцип разработки управляемой формы?
10. Какие элементы управления можно разместить на управляемой форме?
11. Какие возможности предоставляет редактор управляемой формы?
12. Классификация команд в системе 1С:Предприятие 8.
13. Что такое событие объекта системы 1С:Предприятия?
14. Каким образом можно запустить внешнее приложение на исполнение из 1С:Предприятия?
15. Для чего предназначены объекты, располагающиеся на ветви конфигурации «Общие»?
16. Для чего предназначен механизм привязок в формах?
17. Как обеспечить работу 1С:Предприятия в качестве Automation сервера?
18. Что такое СОМ-соединение, какие его отличия от Automation сервера?

Лабораторная работа №3

КОНФИГУРИРОВАНИЕ ПРИКЛАДНОГО РЕШЕНИЯ В СИСТЕМЕ 1С:ПРЕДПРИЯТИЕ В СООТВЕТСТВИИ С ПРИНЦИПОМ УЧЕТА «ОТ ДОКУМЕНТА»

Цель работы: освоение основных приемов работы с документами и регистрами в системе 1С:Предприятие, изучение правил работы с элементами управления и общим модулем, а также принципов применения и формирования макетов.

Теоретические сведения

Концепцию системы учета в системе 1С:Предприятие можно изобразить в следующем виде:

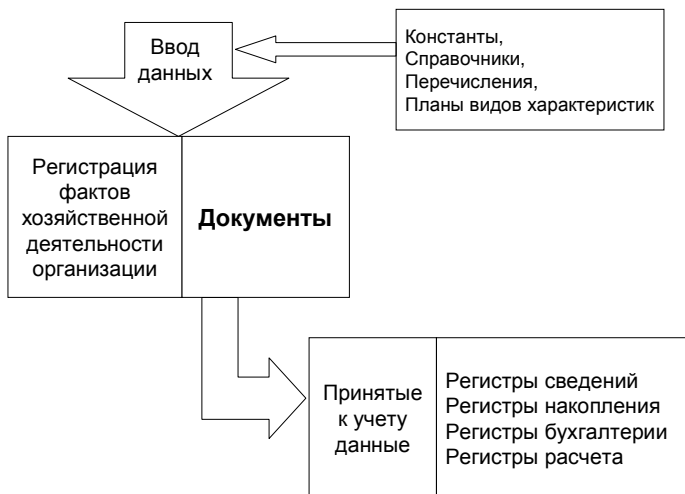


Рис. 40 – Принцип учета «от документа»

Суть принципа учета «от документа»:

- деятельность организации разбивается на хозяйственные операции;
- под каждую операцию создается тип документа, фиксирующего первичную информацию о хозяйственной операции;
- большинство реквизитов документов связывается с классификаторами, т.е. справочниками и другими объектами, предназначенными для хранения нормативно-справочной информации списочного характера;

– вводится различие между *фактом регистрации* хозяйственной операции через документ и *фактом принятия данных к учету*. Принятые к учету данные записываются в учетные объекты (регистры) и влияют на финансовое состояние организации.

Документы в системе 1С:Предприятие

Документ – одно из ключевых понятий системы 1С:Предприятие (как и любой системы учета). При помощи документов организуется ввод в систему информации о совершаемых хозяйственных операциях.

Как правило, документы, которые создаются в процессе конфигурирования, являются электронными аналогами стандартных бумажных документов предприятия некоторого профиля.

Документ, как и большинство объектов конфигурации, характеризуется набором стандартных реквизитов (Ссылка, Номер, Дата, Проведен, ПометкаУдаления), который можно расширить за счет добавления новых реквизитов и табличных частей.

Ссылка. Служебное поле в виде уникального (в пределах всей ИБ) набора символов. Ссылка присваивается при первичной записи объекта (документа) и однозначно его идентифицирует.

Номер, Дата. Каждый документ характеризуется номером, датой и временем создания. Таким образом, все документы привязываются к оси времени и выстраиваются в хронологическом порядке, что необходимо для правильного отражения хозяйственных операций. Номер и Дата присваиваются документу в момент записи.

ПометкаУдаления (тип Булево) служит для обозначения объектов, подлежащих удалению.

Документ в системе 1С:Предприятие может специальным способом (этот способ называется проведение) влиять на другие объекты (на регистры). Состояние документа (проведен/не проведен) фиксируется через стандартный реквизит Проведен (тип Булево).

Механизм проведения отличает документы от остальных объектов конфигурации. В системе 1С:Предприятие принята следующая договоренность: если документ проведен, то он повлиял на состояние данных (состояние учета) в системе, иначе, когда документ только записан, он считается «черновиком». Поэтому запись документа с незаполненными обязательными реквизитами возможна, проведение – запрещено.

При проведении документ обычно отражает зафиксированное им событие (хозяйственную операцию) в регистрах, т.е. формирует движения.

Движения документа – это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом. Таким образом, документ является центральным элементом 1С:Предприятия: с одной стороны, он заполняется на основании справочников и перечислений, с другой стороны, он выполняет проведение по регистрам, на основании чего формируется вся необходимая на предприятии отчетность.

Под термином Документ на этапе конфигурирования подразумевается не конкретный экземпляр документа определенного типа (Накладная №435 от 12.04.04), а шаблон объекта, содержащий определенный набор реквизитов и механизмов управления и проведения. В режиме 1С:Предприятие создается необходимое количество экземпляров данного объекта, имеющих общую предопределенную структуру, но различное содержание реквизитов.

Все документы создаются и редактируются в ветке Документы дерева конфигурации. Помимо документов, в данной ветке также находятся служебные объекты – Нумераторы и Последовательности.

Для настройки алгоритма формирования Номера документа предназначена вкладка Нумерация в мастере документа со следующими свойствами:

- флаг Автонумерация – при установке данного режима номер для нового документа будет устанавливаться автоматически путем увеличения на «1».
- Нумератор – вспомогательный объект, позволяющий установить сквозную нумерацию для документов нескольких видов (например, для кадровых документов). Нумераторы предварительно создаются в ветке Документы > Нумераторы дерева конфигурации. Все документы, которым будет назначен конкретный нумератор, будут нумероваться по единым правилам, которые в нем прописаны.
- Длина номера – количество разрядов (знакомест) в номере. Длина номера в версии 8 увеличена до 50 (в версии 7.7 - 24).
- флаг Контроль уникальности аналогичен справочникам – при попытке записать документ с номером, который уже есть в системе, платформа выдаст ошибку, препятствующую записи в базу.
- Периодичность – свойство устанавливает пределы контроля уникальности номеров документов и период повторяемости номеров. Если установлен Контроль уникальности, то в свойстве Периодичность указывается, в каких пределах будет осуществляться этот контроль. При установленном флаге Автонумерация система 1С:Предприятие будет присваивать очередной порядковый номер каждому новому доку-

менту. После завершения периода, установленного в свойстве Периодичность, нумерация документов начнется заново.

Механизмы создания документов

Создание новых типов документов в конфигураторе не всегда удобно выполнять с нуля. Если новый тип документа предполагает схожий состав реквизитов, табличных частей и поведение с одним из существующих в системе объектов ветви Документы, целесообразно воспользоваться механизмом копирования.

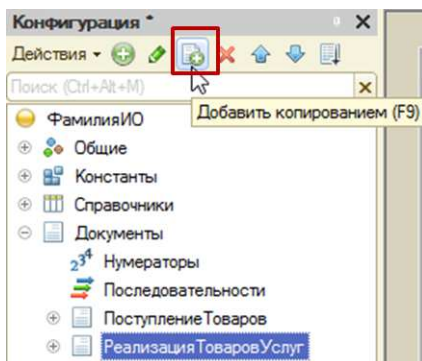


Рис. 41 – Механизм добавления объекта копированием (в конфигураторе)

Механизм создания копированием применим как для объектов одного типа, так и разного. В худшем случае копируется только наименование и к нему добавляется префикс. Если объекты одного типа (рекомендуемый вариант), то в объект-копию копируются: реквизиты и табличные части; формы и модули форм с процедурами-обработчиками событий; права доступа; содержимое модуля объекта и модуля менеджера.

Важно! Принадлежность подсистемам при добавлении новых объектов копированием существующих игнорируется.

Механизм добавления копированием применим как на уровне объекта конфигурации, так и на уровне подчиненных объектов (форм, табличных частей, макетов и др.).

Добавление копированием работает и в режиме 1С:Предприятие.

На уровне платформы заложен и другой механизм – Ввод на основании (предусмотрен для Справочников, Документов, Планов видов характеристик, Планов счетов и Планов видов расчета). Ввод на основании настраивается на одноименной вкладке мастера документа (справочника или др.) в конфигураторе. После в режиме 1С:Предприятие этот механизм позволяет создать экземпляр документа (элемент справоч-

ника) одного типа с уже заполненными реквизитами (или частью реквизитов); значения заполнения берутся из экземпляра документа другого типа, который является основанием для создания нового.

Примеры использования ввода на основании: создание нового документа типа Реализация товаров и услуг на основании документа Заказ клиента 000000001 от 13.10.14; создание нового документа типа Выплата зарплаты на основании документа Начисление зарплаты 000000001 от 31.10.14.

Для настройки ввода на основании в Конфигураторе предусмотрен конструктор ввода на основании.

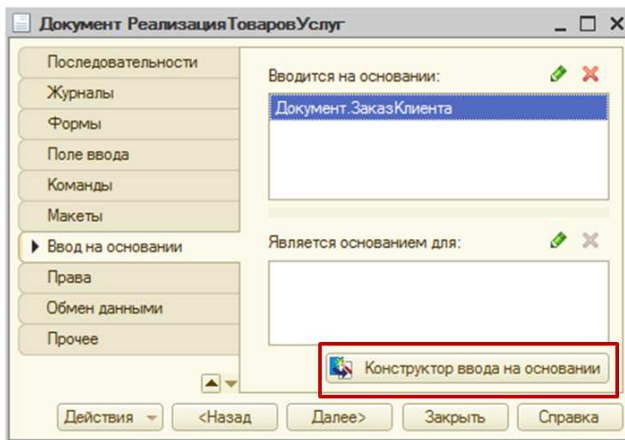


Рис. 42 – Настройка ввода на основании в Конфигураторе

В конструкторе нужно указать соответствие между реквизитами объекта-основания и объекта, который вводится на основании. Конструктор сформирует типовой программный код в модуле объекта (вкладка Прочее > Модуль объекта) в виде процедуры ОбработкаЗаполнения; в интерфейсе автоматически появится соответствующая команда (см. рисунок ниже).

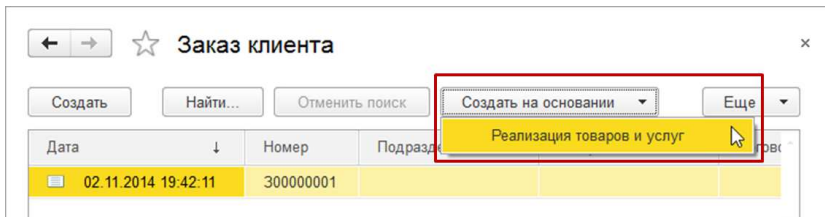


Рис. 43 – Кнопка «Создать на основании»
в командной панели формы документа-основания

Конструктор ввода на основании – один из множества конструкторов в системе 1С:Предприятие, который позволяет визуальными средствами расширить функциональные возможности прикладного решения.

***Вспомогательные для документов объекты конфигурации:
последовательности***

Последовательности служат для контроля логической целостности хронологических цепочек документов, для поддержания актуальности итогов. Например, при внесении изменений задним числом все движения последующих документов могут стать некорректными. Чтобы восстановить корректность, нужно перепровести все нужные документы с момента внесения изменений и внести в них при необходимости определенные изменения. Для решения таких задач и предназначены последовательности.

Системе необходимо знать, с какой даты (с какого документа) нужно восстановить последовательность, когда были внесены изменения «задним числом». Для этого введено понятие Граница последовательности – это маркер, устанавливаемый на позицию последнего проведенного документа. Если какие-то движения изменяются задним числом (при перепроведении документа с новым набором данных, удалении документа или отмене проведения), то Граница последовательности устанавливается на этот документ (или на предыдущий в случае удаления документа). Для каждой введенной в конфигурации последовательности документов система будет автоматически поддерживать границу последовательности. Получаем возможность видеть, с какой позиции, с какой даты у нас могут быть неактуальные итоги, неверные данные. Все документы последовательности, проведенные на основе ставших неактуальными итогов, будут находиться за Границей последовательности.

Последовательности могут иметь подчиненные объекты, называемые измерениями, задаваемые на вкладке Данные. Если для последовательности не создано ни одного измерения, то при восстановлении данной последовательности будут перепроводиться все входящие в последовательность документы. Если требуется, чтобы данная последовательность учитывала не все, а вполне определенные ситуации, то в последовательность включают измерение. В этом случае перепроводить нужно будет только те документы, которые изменяют состояние регистра с учетом свойств измерения. Восстановление последовательностей производится штатным средством Проведение документов в режиме 1С:Предприятие.

Примеры настройки формы документа

При заполнении реквизитов ссылочного типа документов (или других объектов) часто требуется организовать вызов форм выбора с установленным отбором. Рассмотрим две такие ситуации.

Ситуация 1. На форме документа размещены реквизиты Контрагент (тип СправочникСсылка.Контрагенты) и ДоговорКонтрагента (тип СправочникСсылка.Договоры); справочник Контрагенты является владельцем справочника Договоры. Пользователь при заполнении документа выбирает контрагента; далее, при выборе договора контрагента, пользователю должна быть показана форма выбора, содержащая договоры, принадлежащие указанному на форме контрагенту, а не все содержимое справочника Договоры. Другими словами, необходимо открыть форму выбора элемента справочника Договоры с установленным в ней отбором по владельцу. Самый простой способ – использование свойства Связи параметров выбора реквизита ДоговорКонтрагента документа.

Ситуация 2. На форме документа размещена табличная часть с реквизитом НоменклатурнаяПозиция типа СправочникСсылка.Номенклатура. При этом для каждого элемента справочника Номенклатура установлен/сброшен признак Услуга (реквизит типа Булево). При выборе номенклатурной позиции пользователю нужно предлагать только элементы, для которых Услуга = Ложь. В данном случае разумно использовать свойство Параметры выбора реквизита НоменклатурнаяПозиция табличной части документа.

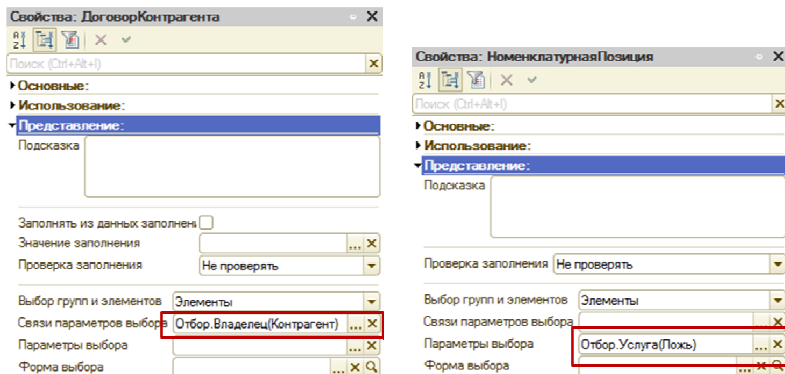


Рис. 44 – Связи параметров выбора (слева) и Параметры выбора (справа)

Документ, как и справочник, может иметь произвольное количество табличных частей. Несколько табличных частей требуются в тех случаях, когда одним документом необходимо зарегистрировать разные по сути, но связанные события. Например, в документе Поступле-

ние товаров в конфигурации Управление торговлей (УТ) есть 3 табличные части: Товары, Услуги и Тара.

Любая табличная часть состоит из нескольких областей: шапка, строки табличной части и подвал. По умолчанию отображение подвала отключено. Для управления отображением подвала предназначен флаг Подвал в свойствах элемента управления формы, связанного с табличной частью объекта.

В подвале удобно размещать итоговые значения по числовым колонкам табличной части. Для этого нужно выбрать поле ввода на форме, соответствующее числовому реквизиту табличной части. Одно из его свойств – ПутьКДаннымПодвала – позволяет указать данные, с которыми связан подвал. Для выбора предлагается несколько агрегатных функций:

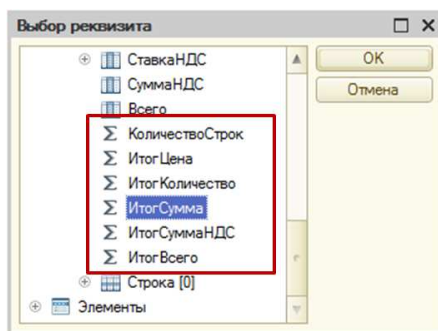


Рис. 45 – Заполнение свойства ПутьКДаннымПодвала

Таким образом, путем настройки свойств подвала табличной части можно отобразить итоговые значения ее числовых колонок без привлечения средств программирования.

N	Номенклатурная позиция	Цена	Количество	Сумма	Ставка НДС	Всего
					Сумма НДС	
1	Белочка	120,00	5	600,00	18%	708,00
					108,00	
2	Грильяж в шоколаде	170,00	3	510,00	18%	601,80
					91,80	
Подвал			ИТОГО:	1 110,00		1 309,80
					199,80	

Рис. 46 – Форма документа с табличной частью, в подвале которой отображены итоговые значения по колонкам Сумма, СуммаНДС и Всего

Номер: Дата:

Подразделение: Контрагент:

Склад: Договор контрагента:

Ответственный:

Сумма документа: Тип цен:

N	Номенклатурная позиция	Цена	Количество	Сумма	Ставка НДС	Всего
1	Балочка	120,00	5	600,00	18%	708,00
					Сумма НДС	
					108,00	
			ИТОГО:	1 110,00		1 309,80
					199,80	

Рис. 47 – Форма документа ПоступлениеТоваров с табличной частью

Программный доступ из формы документа (или элемента справочника, или др. объектов) к данным активной строки табличной части имеет особенности: обращение к реквизитам документа выполняется через основной реквизит формы, *обращение к реквизитам табличной части – через элементы управления*:

&НаКлиенте

Процедура НоменклатурнаяПозицияПриИзменении(Элемент)

//обращаемся к активной строке табл. части Номенклатура

ТекущиеДанные =

Этаформа.Элементы.Номенклатура.ТекущиеДанные;

//обращаемся к реквизиту Цена табличной части

ТекущиеДанные.Цена = 0;

КонецПроцедуры

Журналы документов

Журнал документов – объект конфигурации, предназначенный для группировки связанных по смыслу документов разного вида. Журналы документов позволяют просматривать в одном списке документы разных видов. Фактически журналы не хранят данных, а содержат набор ссылок на группируемые документы.

Свойство Регистрируемые документы определяет, документы каких типов будут отображаться в журнале. Каждый тип документа может быть показан в нескольких журналах.

Для журналов документов предопределен набор стандартных реквизитов (Ссылка, Дата, Номер, Тип, Проведен, ПометкаУдаления). Реквизит Тип содержит тип регистрируемого в журнале документа.

Набор стандартных реквизитов можно расширить путем создания граф. При создании графы из каждого регистрируемого документа нужно выбрать по реквизиту (только по одному, но можно разных типов), который будет в ней отображаться.

***Многообразие программных модулей:
модуль объекта и модуль менеджера, общие модули***

С понятием программного модуля связано понятие контекст модуля. Владелец модуля объекта – объект информационной базы (модуль доступен в мастере объекта конфигурации на вкладке Прочее); владелец модуля менеджера – объект конфигурации (модуль также доступен на вкладке Прочее); владелец общих модулей – прикладное решение (модули располагаются на ветви Общие дерева конфигурации). Таким образом, каждый программный модуль связан с какой-то частью конфигурации. Такая привязка и называется контекстом модуля. Контекст модуля определяет набор объектов, свойств и методов (процедур и функций), доступных для написания и исполнения алгоритма.

Частично данный контекст можно увидеть при редактировании модуля при помощи контекстного меню: через Ctrl + пробел, через точку от указанного объекта, через знак =.

Таким образом, контекст определяет возможность исполнения той или иной строки кода в определенном модуле или его части.

Модуль объекта. На этапе исполнения модуль объекта существует в контексте конкретного экземпляра объекта БД (например, объекта элемента справочника Номенклатура или документа Поступление товаров №0000000001). Из модуля имеется доступ к реквизитам и табличным частям объекта, а также его методам и событиям.

Важно! Поскольку владельцем модуля объекта является объект информационной базы (который не может существовать на стороне клиента), поэтому контекст (набор свойств, методов) модуля объекта не доступен на клиенте.

Если тип прикладного объекта конфигурации не предполагает модификацию хранимых в БД данных на пользовательском уровне (перечисления), либо не связан с хранением данных (журналы документов), то для таких объектов не будет модулей объектов.

У объекта, как и у формы, есть набор предопределенных событий, процедуры-обработчики этих событий располагаются в модуле объекта (ПриЗаписи, ПриКопировании, ПередУдалением, и др.). Имена этих процедур предопределены на уровне платформы и не подлежат изменению. Заполняя тело процедуры-обработчика предопределенного события своим кодом, мы фактически задаем правила поведения системы в отношении объектов ИБ при записи, при копировании и прочих операциях. Перечень предопределенных процедур-обработчиков доступен в выпадающем списке Процедуры и функции [Ctrl+Alt+P], кото-

рый отображается в панели инструментов конфигуратора при активном окне модуля объекта.

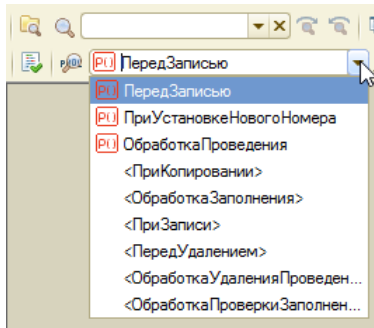


Рис. 48 – Процедуры-обработчики предопределенных событий объекта

Кроме процедур-обработчиков предопределенных событий контекст модуля объекта может включать пользовательские процедуры и функции, которыми разработчик расширяет функциональность объекта.

В модуле объекта контекст доступен через конструкцию `ЭтотОбъект`. Конструкция не является обязательной, зато поможет ориентироваться в контексте модуля.

Следующий листинг содержит пример заполнения реквизита документа `СуммаДокумента` при событии объекта `ПередЗаписью`. Для заполнения используется стандартный метод табличной части `Итог` с параметром – имя реквизита табличной части.

```
Процедура ПередЗаписью(Отказ)
// в реквизит документа СуммаДокумента записывается
// итог по колонке Всего табличной части Номенклатура
ЭтотОбъект.СуммаДокумента =
    ЭтотОбъект.Номенклатура.Итог( "Всего" );
КонецПроцедуры
```

Модуль менеджера существует в контексте набора объектов некого типа (например, документов типа `ПоступлениеТоваров`). С помощью модулей менеджера мы только расширяем набор методов по работе с типами объектов ИБ, т.е. с множеством значений. Поэтому наличие модуля менеджера зависит от того, позволяет ли указанный тип объекта конфигурации работать с наборами значений. Например, константа может содержать только одно значение, поэтому для данного объекта нет модуля менеджера.

Общие модули предназначены для размещения в них текстов функций и процедур, которые могут вызываться из любого другого модуля

конфигурации (требуется использовать ключевое слово Экспорт при описании таких подпрограмм).

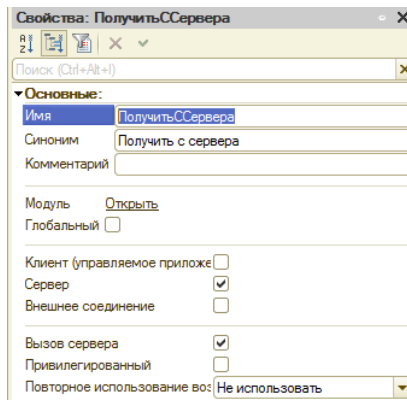


Рис. 49 – Настройки общего модуля

Свойство Глобальный общего модуля. Если это свойство установлено в значение Истина, то экспортируемые процедуры и функции общего модуля являются частью глобального контекста и их имена в различных глобальных общих модулях должны быть уникальными. При вызове таких процедур и функций из других модулей нет необходимости использовать имя глобального модуля.

Свойства Клиент, Сервер, Внешнее соединение определяют место исполнения процедур и функций общего модуля.

Если свойство Вызов сервера установлено в значение Истина, то экспортные процедуры и функции модуля могут быть использованы на клиенте.

В следующем листинге приведена функция ПолучитьЗначениеНДС, определенная в общем модуле ПолучитьССервера (его настройки на рис. 49). Функция используется для получения численного значения ставки НДС, в качестве формального параметра - ссылка на одно из значений перечисления СтавкаНДС. Также в программном коде обозначен способ обращения к значениям перечисления.

```
//функция определена в общем модуле
Функция ПолучитьЗначениеНДС(Ставка) Экспорт
Если Ставка = Перечисления.СтавкиНДС.НДС18
Тогда Возврат 0.18;
ИначеЕсли Ставка = Перечисления.СтавкиНДС.НДС10
Тогда Возврат 0.10;
ИначеЕсли Ставка = Перечисления.СтавкиНДС.НДС20
Тогда Возврат 0.20;
Иначе
```

```

Возврат 0;
КонецЕсли;
КонецФункции

```

Вызов описанной функции возможен из других общих модулей и, например, из модулей форм объектов (документов, справочников). Поскольку функция определена в неглобальном общем модуле, при ее вызове нужно использовать имя самого модуля.

Макеты и создание печатных форм

Объект конфигурации Макет предназначен для хранения различных форм представления данных, которые могут потребоваться каким либо объектам конфигурации или всему прикладному решению в целом. Макеты бывают общие (доступны из любого модуля) и подчиненные прикладным объектам (макеты справочников, документов, журналов документов, отчетов, обработок, регистров и др.).

В системе 1С:Предприятие предусмотрено 9 типов макетов: табличный документ, текстовый документ, двоичные данные, HTML-документ, Active Document и др. Одно из предназначений макета, подчиненного объекту конфигурации и содержащего табличный документ – создание печатной формы этого объекта. Создание печатной формы заключается в конструировании ее составных частей - именованных областей, из которых затем собирается готовая печатная форма. Порядок заполнения областей данными и порядок вывода их в итоговую форму описывается при помощи встроенного языка.

Печатная форма может включать в себя различные графические объекты: картинки, OLE-объекты, диаграммы и т.д.

Макет типа ТабличныйДокумент состоит из ячеек, которые могут быть объединены в именованные области. Каждая ячейка имеет имя и может быть заполнена текстом, через параметр или по шаблону (способ заполнения ячейки указывается через ее свойство Заполнение):

- Текст, содержащийся в ячейке, будет показан на экране;
- Параметр будет заменен некоторым значением, которое может быть присвоено ему средствами встроенного языка. Текст, содержащийся в ячейке, является именем этого параметра;
- Шаблон представляет собой текстовую строку, в определенные места которой будут вставлены значения параметров.

В макете на рис. 50 присутствуют ячейки с разными вариантами Заполнения:

- ячейка R2C2 (2-я строка, 2-й столбец): шаблон с параметрами Номер и Дата;

- ячейка R5C2: текст;
- объединенная ячейка R5C3:R5C4: параметр Номер;
- ячейка R27C5: шаблон с параметром ГлавБух.

	1	2	3	4	5	6	7	8	9	10
Заголовок	1									
	2	<Реализация товаров и услуг [Номер] от [Дата]>								
	3									
Шапка	4									
	5	Номер			<Номер>					
	6	Дата			<Дата>					
	7	Подразделение			<Подразделение>					
	8	Склад			<Склад>					
	9	Контрагент			<Контрагент>					
	10	Договор контрагента			<ДоговорКонтрагента>					
	11	Ответственный			<Ответственный>					
	12	Тип цен			<ТипЦен>					
	13	Сумма документа			имаДокумента>					
	14									
Подписи	17	Главный бухгалтер			</ [ГлавБух] />					
	18									
	19	Директор			</ [Дир] />					
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29									
	30									
	31									

Рис. 50 – Макет типа ТабличныйДокумент с именованными областями Заголовок, Шапка, Подписи

Особенностью макета типа ТабличныйДокумент является то, что возможно задать разный формат (ширину) колонок для разных строк.

В системе 1С:Предприятие возможно создание макетов как вручную, так и с помощью конструктора печати, который доступен на вкладке Макеты мастера объекта либо из его контекстного меню.

На первом шаге в конструкторе печати задается имя процедуры, в которую будет помещен программный ход по заполнению печатной формы. На втором, третьем и последующих шагах выбираются реквизиты объекта, которые нужно печатать в стандартных областях макета: шапке, табличных частях и подвале. На последнем шаге указывается группа команд, которая обозначает, в какое место в командном интерфейсе будет помещена команда формирования печатной формы.

В результате работы конструктора печати происходят следующие изменения в конфигурации:

- создается макет объекта (документа, элемента справочника или др.);
- создается команда Печать и связанная с ней кнопка в командном интерфейсе;
- создается процедура Печать (с программным кодом по заполнению макета данными объекта) в модуле менеджера объекта.

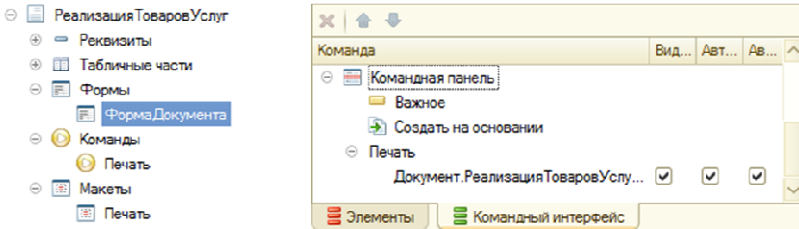


Рис. 51 – Подчиненные документу команда и макет (слева) и элемент командного интерфейса (справа), сформированные автоматически конструктором печати

В следующем листинге приводится типовый код, генерируемый конструктором печати, а также некоторые пояснения.

//процедуры ОбработкаКоманды и Печать определены в модуле
//команды Печать документа РеализацияТоваровУслуг

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды,
ПараметрыВыполненияКоманды)

```
//{_КОНСТРУКТОР_ПЕЧАТИ(Печать)}
//в оперативной памяти создается объект ТабДок
ТабДок = Новый ТабличныйДокумент;
Печать(ТабДок, ПараметрКоманды);
```

```
ТабДок.ОтображатьСетку = Ложь;
ТабДок.Защита = Ложь;
ТабДок.ТолькоПросмотр = Ложь;
ТабДок.ОтображатьЗаголовки = Ложь;
//Заполненный данными ТабДок выводится пользователю
ТабДок.Показать();
//}}
```

КонецПроцедуры

&НаСервере

```
Процедура Печать(ТабДок, ПараметрКоманды)
//вызывается метод Печать документа
//РеализацияТоваровУслуг;
//метод определен в модуле менеджера этого документа
Документы.РеализацияТоваровУслуг.Печать(ТабДок,
ПараметрКоманды);
```

КонецПроцедуры

Основной программный код по заполнению областей и параметров макета Печать данными вынесен в модуль менеджера. Общий алгоритм:

- подключается макет командой ПолучитьМакет;
- выполняется запрос на выборку из БД данных, которые следует поместить в печатную форму;

- из макета выбирается именованная область;
- в параметры именованной области записываются значения;
- именованная область с заполненными параметрами включается в табличный документ.

Следующий листинг содержит пример заполнения параметра ГлавБух области Подписи макета с рис. 50.

```
// получаем именованную область Подписи
ОбластьПодписи = Макет.ПолучитьОбласть ("Подписи");
// находим сотрудника с нужной должностью
ГлавБух =
    Справочники.Сотрудники.НайтиПоРеквизиту
        ("Должность", Справочники.Должности.ГлавныйБухгалтер);
СтруктураПодписи = Новый Структура;
СтруктураПодписи.Вставить ("ГлавБух", ГлавБух.Наименование);
ОбластьПодписи.Параметры.Заполнить (СтруктураПодписи);
ТабДок.Вывести (ОбластьПодписи);
```

Регистры как учетные объекты в системе 1С:Предприятие

В системе 1С:Предприятие определено 4 типа регистров: регистры сведений, регистры накопления, регистры расчета и регистры бухгалтерии. Формальное отличие между ними – набор предоставляемых методов и свойств.

Регистр сведений универсален, остальные типы регистров решают специфические задачи (регистр накопления – накопление остатков и оборотов по различным показателям, бухгалтерии – организация бухгалтерского учета, расчета – выполнение периодических и не периодических расчетов: зарплата и т.п.).

Информация в регистрах хранится в виде записей. Регистры, в отличие от справочников/перечислений/документов, не создают новый тип (не являются типобразующими объектами). Поэтому нельзя завести ссылку на запись регистра.

Напоминание. При добавлении нового справочника (документа, перечисления) в системе появляется новый тип данных СправочникСсылка.<Имя справочника> (ДокументСсылка.<Имя документа>, ПеречислениеСсылка.<Имя перечисления>).

На физическом уровне регистр – таблица, где колонки могут быть нескольких типов: измерения, ресурсы, реквизиты. Измерение – единица детализации хранимой информации (в каких разрезах хранятся данные), ресурс – единица хранения информации, реквизит – некая сопроводительная информация для записей регистра.

Получаем, что регистр представляет собой многомерный массив данных: индексы массива – измерения, элементы массива – ресурсы.

Доступ к ресурсам может осуществляться как по полному набору измерений, так и по их подмножеству, и тогда результатом будет не одна запись регистра, а множество:

$$w: (x, y, z) \rightarrow w_{xyz}$$

$$W: (x, y) \rightarrow W = \{w_{z1}, w_{z2} \dots w_{zn}\}$$

Важно! В отличие от справочников и документов информация в регистре находится не по ссылке (ссылок на записи регистра не существует, нет стандартного реквизита *Ссылка*), а по адресу, т.е. по набору измерений.

Основная цель использования регистров – это накопление информации о хозяйственных операциях, которая вводится с использованием документов или при помощи расчетов. В дальнейшем эта информация извлекается и анализируется с предоставлением пользователю отчетных форм. С одной стороны, сохранение части данных документов в регистрах (при проведении документов) приводит к хранению избыточной информации. Но с другой стороны, только регистры поставляют методы, позволяющие быстро анализировать данные (а именно: получать актуальные значения реквизитов, вычислять остатки, обороты и итоги по некоторым показателям), которые аккумулируются в регистрах. Этим и обосновывается наличие регистров в системе 1С:Предприятие.

Регистры сведений и регистры накопления

Основная задача регистра сведений – хранить существенную для прикладной задачи информацию, состав которой развернут по определенной комбинации значений и при необходимости развернут во времени. Примеры использования регистра сведений: Курсы валют (хранение курсов в разрезе валют), Цены номенклатуры (хранение цен в разрезе товаров, поставщиков, типов цен). *Как правило, в один регистр сведений объединяются ресурсы, значения которых обычно изменяются одновременно.*

На тип ресурсов и измерений не накладывается ограничений, но рекомендуется для измерений использовать ссылочные типы.

Регистры сведений позволяют хранить историю изменения реквизитов и констант и получать их на заданную точку на временной оси. Важно отметить, что, если не нужно разворачивать информацию по какому-либо набору измерений, кроме времени, то вообще не нужно создавать измерения. Штатный механизм для хранения истории изменения информации – свойство Периодичность регистра сведений. Если выбран какой-то вариант периодичности (в пределах секунды, дня, месяца, квартала, года), то время добавляется как дополнительное из-

мерение (активируется стандартный реквизит Период). Информация в регистре не может изменяться чаще, чем указано в свойстве Периодичность. Если выбирается значение Непериодический, то вновь введенное значение ресурса с конкретным набором измерений затрет старое. Если же задана периодичность и период, за который было сделано изменение, не меньше установленного, то старое значение автоматически сохранится. В этом случае при добавлении записи в регистр происходит округление даты до начала периода.

Свойство Режим записи регистра сведений определяет способ изменения его записей:

- Независимый режим, при котором изменения в регистр вносятся вручную;
- Режим Подчинение регистратору, при котором изменения в регистр вносит документ (т.е. регистратор). Такой способ воздействия на регистр называется проведением, в результате которого по нашему заданному алгоритму данные из реквизитов документа дублируются в регистре. Если регистр подчинен регистратору, то в командном интерфейсе видимость регистра по умолчанию снята (даже если регистр включен в подсистемы), т.к. в этом случае регистр считается служебным.

В режиме записи Подчинение регистратору запись регистра жестко подчинена документу-регистратору (и может быть удалена при удалении документа). Кроме того, появляется дополнительный вариант периодичности – По позиции регистратора, при котором хронологически запись регистра будет характеризоваться парой Дата+Ссылка на документ-регистратор. Это обеспечивает максимальную детализацию, с точностью больше секунд.

В режиме записи Подчинение регистратору задействуются стандартные реквизиты Регистратор (ссылка на документ-регистратор), Номер строки (с одним документом может быть связано несколько записей регистра).

При независимом режиме записи регистра подчиняются так называемым ведущим измерениям.

Свойство измерения Ведущее устанавливает явную связь между объектом и ссылками на него в записях регистра сведений. Благодаря этой связи при удалении объекта могут быть удалены записи из регистра сведений, ссылающиеся на удаляемый объект.

Свойство Основной отбор устанавливает измерение как одно из основных для оптимизации запросов. При включении свойства Запрет незаполненных значений нельзя будет сохранить значение в регистр с

незаполненным значением данного измерения. Свойство Индексировать доступно для редактирования, если измерение не является ведущим. Для таких измерений создается отдельный индекс, что увеличивает производительность при работе с регистром. Но чрезмерное индексирование может замедлить работу системы.

В регистре сведений не может существовать две записи с одинаковым набором значений измерений и периодом (для периодических регистров). Это принципиально отличает регистр сведений от регистра накоплений, в котором такая ситуация является вполне нормальной (например, два прихода одного товара на один и тот же склад). Если мы запишем какое-либо значение в ресурс с набором измерений, который уже был введен в регистре, то возникнет ошибка (в общем случае):

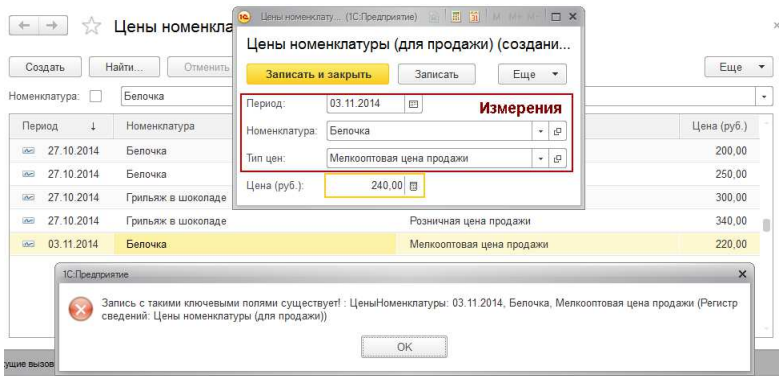


Рис. 52 – Ошибка, вызванная нарушением уникальности записи в регистре сведений

Все регистры имеют нечто общее: записи регистров не имеют объектной природы (на записи регистров нельзя завести ссылки); структура регистров определяется набором измерений, ресурсов, реквизитов; для каждого типа регистров на уровне платформы определены «специфические умения».

Говоря о регистрах сведений, среди таких «специфических умений» выделяют возможность быстро получать срезы первых (последних) значений ресурсов.

Регистры накопления используются для накопления информации о наличии и движении средств (товарных, денежных и других) и позволяют быстро получать информацию об остатках, оборотах, остатках с оборотами, итогах на интересующие даты, за указанные периоды. Т.е. регистр накопления оптимизирован для решения задач получения остатков и оборотов показателей – то «специфическое» умение (мето-

ды объекта), за которое регистр накопления выделен в отдельный тип объектов.

Регистр накопления не может существовать без ресурсов. Тип ресурсов - Число (т.к. при получении остатков, оборотов, итогов регистра имеют место арифметические операции).

Изменение состояния регистров накопления выполняется при проведении документа. Процедура проведения документа содержится в модуле документа. Такая процедура содержит алгоритм формирования сведений об изменениях в регистрах, которые необходимо выполнить при проведении документа. Эти сведения называются движениями регистра и представляют собой записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

Движения регистров содержат только приращения (со знаком «+» или «-») значений ресурсов регистра, а не итоговые величины значений.

Изменения в регистрах вносятся движениями регистра. Движения регистров оказывают влияние на его итоги. Итоги – это сводная информация регистров накопления, которая получается путем суммирования значений, вносимых движениями регистров.

Специалист, выполняющий конфигурирование системы, имеет возможность предоставить конечному пользователю средства просмотра движений регистров. Конфигуратор позволяет создавать экранные и печатные формы для просмотра и анализа движений регистров.

В отличие от движений регистра, нет возможности непосредственно просматривать итоги регистра накопления. Для обращения к итогам в конфигурации может быть создано необходимое число отчетов, которые будут обращаться к итогам и выдавать их в виде товарных отчетов, складских карточек и ведомостей и т.д.

В зависимости от того, показатели какого вида должны храниться в регистре, определяется вид регистра (регистр накопления с видом остатки или регистр накопления с видом обороты).

Показатели вида Остатки: остаток товара на складе, деньги в кассе или на расчетном счете, взаиморасчеты с контрагентами (увеличение/уменьшение долга). Значение такого показателя может увеличиваться и уменьшаться. Для регистра накопления вида Остатки предусмотрен стандартный реквизит ВидДвижения = Приход/Расход, который хранит знак приращения значений ресурсов.

Показатели вида Обороты: как правило, изменяется только в одну сторону (увеличивается). Примеры: количество проданного товара, прибыль (стоит на месте или увеличивается).

Для регистра остатков методы встроенного языка позволяют получить остатки регистра накопления на заданный момент времени. Есть возможность фильтрации по значениям измерений, а также получения остатков в разрезе других измерений.

Физически регистр накопления представляется в виде двух таблиц БД – таблица движений (хранит исходные данные, т.е. приращения) и таблица итогов (формируется в БД системой автоматически). Итоги в регистре накопления хранятся с периодичностью месяц. При получении итогов на момент, отличный от начала месяца, система будет анализировать движения от ближайшего рассчитанного месяца.

← → ☆ Товары на складах

Найти... Отменить поиск

Еще ▾

Период	Регистратор	Номер строки	Товар	Склад	Количе...	Сумма	Сумма НДС
+ 02.11.2014 18:50:28	Поступление товаров P000000001 от 02.11.2014 18:50:28	1	Белочка		5	600,00	108,00
+ 02.11.2014 18:50:28	Поступление товаров P000000001 от 02.11.2014 18:50:28	2	Грильяж в шоко...		3	510,00	91,80
- 02.11.2014 20:11:37	Реализация товаров и услуг P000000002 от 02.11.2014 20:11:...	1	Грильяж в шоко...	Основной	3	102,00	18,36
- 02.11.2014 20:11:37	Реализация товаров и услуг P000000002 от 02.11.2014 20:11:...	2	Белочка	Основной	4	800,00	144,00

Рис. 53 – Регистр накопления ТоварыНаСкладах (регистр вида Остатки)

← → ☆ Продажи

Найти... Отменить поиск

Еще ▾

Период	Регистратор	Номер строки	Номенклатура	Контрагент	Количе...	Сумма	Сул
02.11.2014 20:11:37	Реализация товаров ...	1	Грильяж в шоко...	ООО Мед-П...	3	102,00	
02.11.2014 20:11:37	Реализация товаров ...	2	Белочка	ООО Мед-П...	4	800,00	
02.11.2014 20:11:37	Реализация товаров ...	3	Доставка	ООО Мед-П...	1	45,00	

Рис. 54 – Регистр накопления Продажи (регистр вида Обороты)

Регистры используются для хранения информации, к которой требуется получать оперативный доступ. Также регистры позволяют получать наиболее достоверную информацию о состоянии средств. Так как процессы сохранения документа и записи изменений в регистрах разделены (допускается сохранение документа без его проведения), может возникать расхождение между данными документов и информацией в регистрах. Но регистр, в отличие от документа, является хранилищем итоговой информации, поэтому именно запись изменений в регистры служит подтверждением того, что хозяйственная операция совершена.

Проведение документа может осуществляться одновременно по нескольким регистрам, если этого требует учетная политика предприятия. Для осуществления такой возможности, необходимо выбрать необходимые регистры, по которым будут осуществляться движения, на

вкладке Движения требуемого документа. Затем можно воспользоваться конструктором движений для формирования алгоритма внесения изменений в регистры (т.е. сформировать необходимые движения по выбранным регистрам).

Проведение документов по регистрам

Проведение документа – это создание движений по различным регистрам системы, то есть это и есть регистрация движений документа в системе. Обычно движения основаны на данных, введенных в реквизиты документа. Потом на основании этих движений строятся различные отчеты, определяются остатки и обороты чего-либо. Конечно, отчеты можно сформировать на основании самих реквизитов документов, без формирования движений, но неэффективный и нерекомендуемый подход.

Свойство Проведение на вкладке Движения мастера документа устанавливает, разрешено ли проведение документа при записи. Если выбрано Разрешить, то документ осуществляет движения регистров (изменяет их состояние).

Свойство Оперативное проведение устанавливает, разрешено ли оперативное проведение документа при записи. Для документов с разрешенным оперативным проведением при выборе не текущей даты, проведение осуществляется в неоперативном режиме, так как учитывается уже свершившийся факт, который не требует контроля, осуществляемого в оперативном режиме (например, проверка остатка указанного в расходной накладной товара).

Свойство Удаление движений позволяет выбрать между автоматическим и неавтоматическим удалением движений регистров сведений, накопления, бухгалтерии и расчета. Движения могут удаляться при перепроведении, удалении и отмене проведения документа. Если проведение документа запрещено, то данное свойство недоступно. Стандартным считается режим автоматического удаления движений. Если его отменить, то ответственность за состояния движений ложится на плечи разработчиков - необходимо средствами встроенного языка контролировать и удалять движения. При помощи отключения автоудаления можно, например, сделать так, чтобы при введении новой строки в проведенный документ совершались движения только по этой строке, а по другим они не трогались.

Ниже находится окно списка регистров. В нем нужно отметить только те регистры, по которым возможны движения данного документа. Не стоит отмечать все регистры, если у вас их большое количество – это может плохо сказаться на производительности системы в

целом. Будьте аккуратны со снятием пометки регистра – после реструктуризации базы движения по этому регистру будут безвозвратно уничтожены, установка пометки обратно их не вернет. Потребуется перепроводить все документы данного типа.

Если выбрать хотя бы один регистр, то станет активной кнопка Конструктор движений. Конструктор движений позволяет визуально и очень гибко настроить алгоритм проведения документа по нужным регистрам и сформировать типовую процедуру проведения в автоматическом режиме.

Поле	Выражение
Товар	ТекСтрока Товары.Товар
Склад	Склад
Количество	ТекСтрока Товары.Количество
Сумма	ТекСтрока Товары.Сумма
СуммаНДС	ТекСтрока Товары.СуммаНДС



Рис. 55 – Конструктор движений для документа РеализацияТоваровУслуг

Рассмотрим ситуацию, когда в конфигурации имеется документ РеализацияТоваровУслуг с двумя табличными частями (Товары и Услуги). Документ отражает продажу товаров и оказание услуг клиентам. Эта хозяйственная операция влияет на остатки единиц товаров на складах, а также на сумму прибыли от продаж. Поэтому при проведении документа изменяет состояние двух регистров накопления: ТоварыНаСкладах (вида Остатки) и Продажи (вида Обороты).

Основная задача при работе с конструктором движений – указать, по какому правилу (выражению) будут заполняться поля регистра. Выражение, как правило, составляют реквизиты документа. Соответствие между полями (измерениями, ресурсами) регистра и реквизитами документа устанавливается в первую очередь по типу. Если есть

совпадения по типам и именам, то срабатывает кнопка Заполнить выражения. Корректность выражения конструктором не проверяется.

При необходимости проводить документ по нескольким регистрам (как на рис. 55), в области Регистры по кнопке Добавить создается нужное количество движений.

Маркеры  и  обозначают, что при проведении документа по соответствующему регистру в нем формируется столько записей, сколько строк присутствует в табличной части документа. Для обеспечения такого поведения в одноименном поле в конструкторе движений следует выбрать табличную часть документа.

При создании движений по регистру накопления также в конструкторе следует указать Тип движения регистра. Поскольку отражение в учете документа РеализацияТоваровУслуг связано с уменьшением единиц товаров на складе, то в данном случае тип движения – Расход.

После завершения работы конструктора в модуле объекта документа формируется процедура ОбработкаПроведения. Процедура является транзакцией, т.е. если предполагается проведение документа по нескольким регистрам, но в процессе выполнения произошел сбой, то данные не запишутся ни в один регистр (действие будет полностью отменено).

Объектная техника доступа к данным регистров

Для работы с регистром сведений средствами встроенного языка применяются следующие объекты:

```
// РегистрСведенийМенеджер.<Имя регистра сведений>
// объект этого типа предназначен для управления данными
// конкретного регистра сведений. Позволяет осуществлять поиск,
// выбирать и создавать записи регистра сведений

// РегистрСведенийМенеджерЗаписи.<Имя регистра сведений>
// объект этого типа позволяет читать, записывать и удалять
// отдельную запись регистра сведений
ЦенаТовара =
    РегистрыСведений.ЦеныНоменклатуры.СоздатьМенеджерЗаписи();
ЦенаТовара.Период = ТекущаяДата();
// заполняем остальные измерения и ресурсы
ЦенаТовара.Записать(Истина);
// метод Записать(<Замещать>), параметр <Замещать> позволяет
// установить или отключить режим замещения существующей записи
// с тем же набором значений измерений регистра и с тем же пе-
// риодом, если Замещать = Истина, то происходит замещение,
```

иначе – та же ошибка, что и при интерактивном добавлении одинаковых записей в регистр сведений

В первой строке листинга создается экземпляр объекта РегистрСведенийМенеджерЗаписи, который связывается с переменной ЦенаТовара. Далее заполняются ресурсы и измерения регистра. Запись будет производиться для всех ресурсов. Если какому-то ресурсу не будет присвоено значение, то запишется пустое значение. При помощи метода Записать измерения заносятся в БД.

Для периодических регистров сведений предусмотрены механизмы извлечения наиболее ранней/поздней информации с использованием следующих методов встроенного языка:

```
ПолучитьПервое(<НачалоПериода>, <Отбор>)
// Метод получает значения ресурсов наиболее ранней записи
регистра, соответствующей указанным периоду и значениям из-
мерений регистра.
ПолучитьПоследнее(<КонецПериода>, <Отбор>)
СрезПервых(<НачалоПериода>, <Отбор>)
// Получает наиболее ранние записи регистра, соответствующие
установленным в параметрах метода значениям ключевых полей.
Записи подбираются для каждого сочетания из всех имеющихся
значений измерений регистра.
СрезПоследних(<КонецПериода>, <Отбор>)
```

Подробнее о перечисленных методах – в справке 1С:Предприятия по запросу РегистрСведенийМенеджер.<Имя регистра сведений>.

Для обращения к итогам регистра накопления также предусмотре- ны объекты и методы встроенного языка:

```
РегистрНакопленияМенеджер.<Имя регистра накопления>
// предназначен для управления данными конкретного регистра
накопления. Позволяет получать остатки и обороты по данным
регистра накопления, формировать выборки, создавать формы и
макеты
Обороты(<НачалоПериода>, <КонецПериода>, <Отбор>, <Измере-
ния>, <Ресурсы>)
Остатки(<МоментВремени>, <Отбор>, <Измерения>, <Ресурсы>)
```

Подробнее о перечисленных методах – в справке 1С:Предприятия по запросу РегистрНакопленияМенеджер.<Имя регистра накопления>.

Содержание работы

Задания *Intermediate*.

1. Создайте документ ПоступлениеТоваров, отражающий хозяйственную операцию по поступлению товаров от поставщика на склад организации.

Таблица 18. Свойства документа ПоступлениеТоваров

Свойство	Значение
Подсистемы	ОтделЗакупок > Документы (подчиненную подсистему Документы нужно предварительно создать)
Состав реквизитов	Подразделение, тип СправочникСсылка.Подразделения; Склад, тип СправочникСсылка.Склады; Контрагент, тип СправочникСсылка.Контрагенты; ДоговорКонтрагента, тип СправочникСсылка.Договоры; Ответственный, тип СправочникСсылка.Сотрудники; ТипЦен, тип СправочникСсылка.ТипыЦенНоменклатуры (справочник нужно предварительно создать); СуммаДокумента, тип Число:10:2, неотрицательное; Комментарий, тип Строка:100:переменная
Состав табличных частей	Одна табличная часть Номенклатура с реквизитами: НоменклатурнаяПозиция, тип СправочникСсылка.Номенклатура; Цена, тип Число:10:2, неотрицательное; Количество, тип Число:10:0, неотрицательное; Сумма, тип Число:10:2, неотрицательное; СтавкаНДС, тип ПеречислениеСсылка.СтавкиНДС; СуммаНДС, тип Число:10:2, неотрицательное; Всего, тип Число:10:2, неотрицательное
Нумерация	Периодичность = В пределах года

2. Создайте форму документа ПоступлениеТоваров. Вид формы приведен на рис. 46, 47. Обратите внимание на наличие подвала табличной части с итогами по колонкам Сумма, СуммаНДС и Всего.

Также необходимо предусмотреть, чтобы при заполнении документа пользователю предлагались только договоры, принадлежащие выбранному на форме контрагенту.

Совет. Настройте свойство Связи параметров выбора реквизита ДоговорКонтрагента.

Кроме того, в соответствии с логикой документа в качестве номенклатурной позиции можно выбирать только элементы справочника Номенклатура со значением Услуга = Ложь.

Совет. Настройте свойство Параметры выбора реквизита НоменклатурнаяПозиция табличной части Номенклатура.

3. Предусмотрите автоматические подстановки и расчеты в табличной части документа ПоступлениеТоваров в соответствии со следующей логикой:

- изменение поля СтавкаНДС активной строки табличной части инициирует перерасчет полей СуммаНДС и Всего;
- изменение поля Цена инициирует перерасчет полей Сумма, СуммаНДС, Всего;
- при изменении поля Количество выполняется перерасчет аналогично случаю с изменением поля Цена;
- при изменении номенклатурной позиции в соответствующее поле подставляется ставка НДС, поле Цена обнуляется, остальные поля перерасчитываются исходя из изменения цены.

Необходимые процедуры и функции расположите в модуле формы документа ПоступлениеТоваров.

Для числовых реквизитов табличной части настройте формат представления нулевых значений (по умолчанию поля с нулевыми значениями отображаются как пустые).

4. Организуйте заполнение реквизита СуммаДокумента перед записью документа ПоступлениеТоваров. Реквизит должен содержать итог по колонке Всего табличной части.

Совет. Алгоритм заполнения реквизита поместите в модуль объекта документа (используйте процедуру-обработчик ПередЗаписью).

Совет 2. Нет необходимости программно обходить в цикле табличную часть для суммирования всех значений. Используйте данные, с которыми связан подвал табличной части.

5. Измените алгоритм формирования номера для документов ПоступлениеТоваров. Вместо номеров 000000001, 000000002 и т.п. документам должны присваиваться номера с префиксом П: П00000001, П00000002 и т.п.

Совет. Алгоритм формирования номера поместите в модуль объекта документа (используйте процедуру-обработчик ПриУстановкеНовогоНомера).

6. Создайте документ РеализацияТоваровУслуг. Документ отражает хозяйственную операцию по продаже товаров и оказанию услуг клиентам. Документ входит в состав подсистемы ОтделПродаж > Документы (дочернюю подсистему Документы предварительно добавить).

Совет. Структура документа РеализацияТоваровУслуг схожа со структурой документа ПоступлениеТоваров, поэтому для выполнения п.б. целесообразно использовать механизм создания копированием, т.е.

создать документ РеализацияТоваровУслуг копированием документа ПоступлениеТоваров, далее внести изменения в копию.

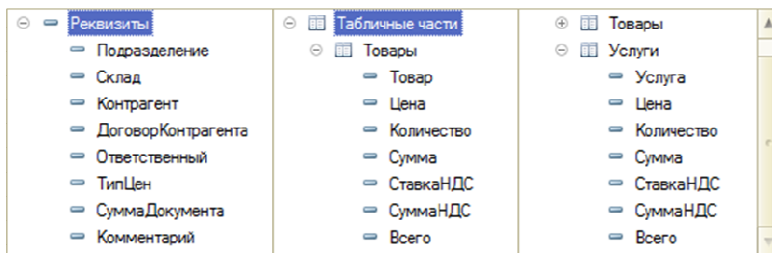


Рис. 56 – Состав реквизитов и табличных частей документа РеализацияТоваровУслуг

Для документа РеализацияТоваровУслуг выполнить п. 2-5 с учетом наличия двух табличных частей. Префикс номера для документа Р.

7. Создайте документ ЗаказКлиента (отражает хозяйственную операцию по предварительному заказу товаров клиентом) копированием документа РеализацияТоваровУслуг. Удалите лишние реквизиты и табличные части, префикс номера измените на 3.

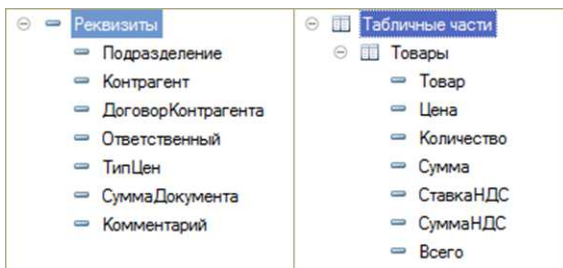


Рис. 57 – Состав реквизитов и табличных частей документа ЗаказКлиента

8. Создайте журнал документов ДокументыПоЗакупкамПродажам. В нем регистрируются документы, созданные в предыдущих заданиях. Журнал содержит три графы: Контрагент, ТипЦен, СуммаОперации.

9. На ветви Общие дерева конфигурации создайте группы команд: ЖурналыДокументов (категория Панель навигации формы) и Печать (категория Командная панель формы).

10. Разместите на форме документов всех видов навигационную команду по переходу к журналу документов, используйте ранее созданную группу команд ЖурналыДокументов.

Совет. п.10 выполняется через командный интерфейс формы документа с использованием глобальных команд.

Рис. 58 – Навигационная ссылка для перехода к журналу документов из формы одного из документов

11. Создайте макет для вывода документа РеализацияТоварУслуг на печать (с помощью конструктора печати). В конструкторе укажите, что команда печати должна располагаться в группе Командная панель формы. Печать, созданной в п. 9.

12. Добавьте в состав конфигурации регистр сведений ЦеныНоменклатуры.

Таблица 19. Свойства регистра сведений ЦеныНоменклатуры

Свойство	Значение
Синоним	Цены номенклатуры (для продажи)
Периодичность	В пределах дня
Режим записи	Независимый
Подсистемы	ОтделПродаж > Регистры (дочернюю подсистему Регистры предварительно добавить)
Состав измерений и ресурсов	<p><i>Ведущие измерения:</i></p> <p>Номенклатура, тип СправочникСсылка.Номенклатура; ТипЦен, тип СправочникСсылка.ТипыЦенНоменклатуры;</p> <p><i>Ресурсы:</i></p> <p>Цена, синоним Цена (руб.), тип Число:10:2, неотрицательное</p>

13. Добавьте в состав конфигурации регистр накопления ТоварыНаСкладах с видом регистра Остатки, изменения в регистр вносятся документами-регистраторами ПоступлениеТоваров и РеализацияТоваровУслуг.

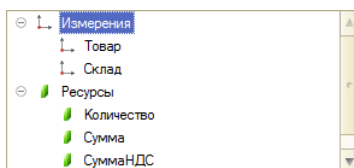


Рис. 59 – Структура регистра ТоварыНаСкладах

14. Добавьте в состав конфигурации регистр накопления Продажи с видом регистра Обороты, изменения в регистр вносятся документом-регистратором РеализацияТоваровУслуг.

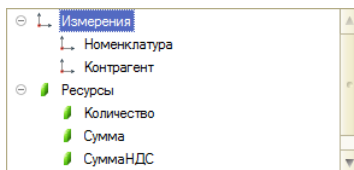


Рис. 60 – Структура регистра Продажи

15. Настройте видимость созданных регистров в пользовательском интерфейсе.

16. С помощью конструктора движений настройте проведение документов ПоступлениеТоваров (по регистру ТоварыНаСкладах) и документа РеализацияТоваровУслуг (по регистрам ТоварыНаСкладах и Продажи).

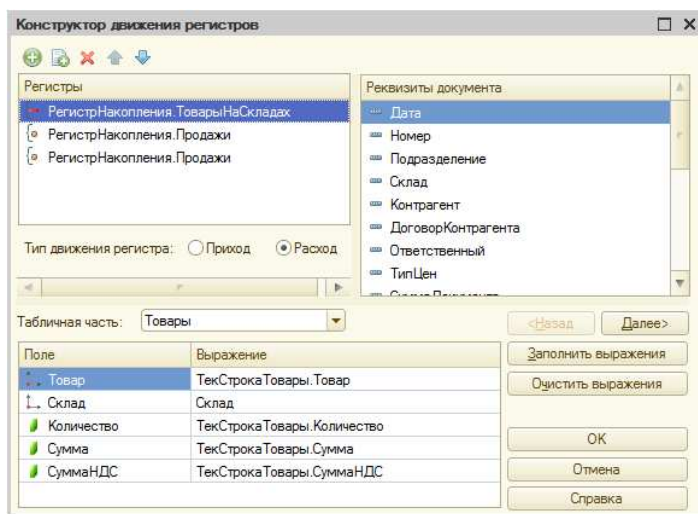


Рис. 61 – Настройка проведения документа РеализацияТоваровУслуг в конструкторе

17. В пользовательском режиме создайте по 4-5 экземпляров документов каждого вида, заполните их и проведите по регистрам.

Задания Upper-Intermediate.

18. Реализуйте возможность создания документа РеализацияТоваровУслуг на основании документа ЗаказКлиента. Используйте механизм ввода на основании.

19. Модифицируйте макет Печать и одноименную процедуру из модуля менеджера, созданные при выполнении п. 11.

Рис. 62 – Модифицированная печатная форма

В процедуре Печать, во-первых, нужно предусмотреть заполнение параметров Номер, Дата из области Заголовок и параметров ГлавБух и Дир из области Подписи. Во-вторых, если документ содержит пустые табличные части, не нужно выводить на печать пустые шапки этих таблиц.

20. Настройте форму списка регистра сведений ЦеныНоменклатуры: добавьте возможность отбора по номенклатуре и типу цен.

Период	Номенклатура	Тип цен	Цена (руб.)
27.10.2014	Белочка	Розничная цена продажи	250,00
27.10.2014	Грильяж в шоколаде	Оптовая цена продажи	300,00
27.10.2014	Грильяж в шоколаде	Розничная цена продажи	340,00
03.11.2014	Белочка	Мелкооптовая цена продажи	220,00

Рис. 63 – Форма списка регистра сведений

21. Ознакомьтесь, как созданные в рамках лабораторной работы документы и регистры реализованы в типовой конфигурации 1С:Бухгалтерия предприятия.

Задания Advanced.

22. С помощью инструмента Отладка > Замер производительности зафиксируйте время исполнения процедур и функций (отвечающих за вычисления в табличных частях документов) модулей формы документов ПоступлениеТоваров, ЗаказКлиента, РеализацияТоваровУслуг.

23. Перенесите алгоритмы расчета/перерасчета/подстановки значений табличных частей документов из модулей формы в общие модули.

24. Выполнив п. 21 еще раз, проанализируйте, как изменилось время выполнения алгоритмов расчета/перерасчета/подстановки.

25. Реализуйте подстановку цены продажи номенклатуры из регистра сведений в табличную часть документа РеализацияТоваровУслуг. Подстановка цены выполняется при изменении значения поля Товар/Услуга. Нужная цена выбирается из регистра по периоду, номенклатуре и типу цены.

Совет. Используйте метод СрезПоследних периодического регистра сведений.

Контрольные вопросы

1. Что такое регистр сведений и для чего он используется?
2. Для чего используется свойство Периодичность регистра сведений?
3. Что такое Режим записи для регистра сведений?
4. Что такое измерения и ресурсы для регистра сведений?
5. Какие объекты применяются для работы с регистром сведений средствами встроенного языка?
6. Что такое документ в системе 1С:Предприятие 8?
7. Какие существуют стандартные реквизиты для документов?
8. Каковы принципы нумерации документов, что такое Нумератор?
9. Что такое понятие «движения» и «проведение»?
10. Какие виды проведения документов существуют?
11. Что такое Последовательности, каковы принципы работы с последовательностями?
12. Для чего в системе 1С:Предприятие версии 8 применяются регистры?
13. Для чего предназначены регистры бухгалтерии?
14. Для чего используются регистры расчета?
15. Что такое регистр накопления?

16. Как можно представить измерения и ресурсы регистра?
17. Какие типы регистра накопления используются в системе 1С:Предприятие 8?
18. Как осуществляется проведение документа по нескольким регистрам?
19. Для чего используются отчеты в системе 1С:Предприятие 8?
20. Что такое Макет в системе 1С:Предприятие 8?
21. Каким образом можно создать макет?

Лабораторная работа №4 **МЕХАНИЗМ ЗАПРОСОВ В СИСТЕМЕ 1С:ПРЕДПРИЯТИЕ.** **ОБРАБОТКИ И ОТЧЕТЫ.**

Цель работы: освоение табличной техники доступа к учетным данным посредством языка запросов 1С:Предприятие; получение навыков разработки отчетов и обработок.

Теоретические сведения

Табличная техника доступа к учетным данным подразумевает использование механизма запросов для чтения данных из базы.

Информация хранится в БД в виде таблиц: каждому объекту конфигурации сопоставляется одна или несколько таблиц БД (в зависимости от наличия табличных частей в структуре объекта и других условий). Поэтому при использовании механизма запросов взаимодействие с БД осуществляется на уровне записей таблиц.

Декларативный язык запросов системы 1С:Предприятие

Запрос – инструкция системе относительно того, какие данные необходимо извлечь из БД. Такая инструкция описывается на встроенном языке запросов, который основывается на классическом языке SQL, но имеет ряд существенных расширений. Расширения направлены на поддержку работы с объектами и эффективное решение экономических задач. Язык запросов оптимизирован для чтения данных из больших массивов информации.

Запросы формируются на основе таблиц, результатом запроса также является таблица (частный случай: результат, состоящий из одной колонки/строки). В простейшем запросе задается два основных ограничения на чтение данных: 1) из какого источника следует получить данные, т.е. из каких таблиц БД; 2) что именно следует извлечь, т.е. значения каких полей таблиц БД.

В качестве источника данных для запроса могут выступать:

- реальные таблицы, данные которых физически хранятся в БД;
- виртуальные (расчетные) таблицы, данные которых вычисляются системой 1С:Предприятие в момент обращения к этому источнику;
- временные таблицы, которые хранятся в оперативной памяти компьютера и позволяют определить в качестве источников в табличном виде абсолютно любую информацию (из незаписанного документа, из внешней СУБД).

Виртуальные таблицы существуют для всех регистров. Например, обратившись к виртуальной таблице регистра накопления, можно получить распределение остатков и оборотов некоторых показателей (ресурсов регистра) в разрезе измерений, определенных в структуре регистра.

Структуру запроса можно описать следующим синтаксическим правилом:

ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>] <Список полей выборки | *>

[ИЗ <Список таблиц источников>]

[ГДЕ <Список условий отбора>] [ПОДОБНО]

**[СГРУППИРОВАТЬ ПО <Список полей группировки>]
[ИМЕЮЩИЕ <Список условий>]**

[ДЛЯ ИЗМЕНЕНИЯ <Список блокируемых данных>]

[УПОРЯДОЧИТЬ ПО <Список полей упорядочивания>] [ИЕРАРХИЯ] | [АВТОУПОРЯДОЧИВАНИЕ]

[ИТОГИ <Агрегатные функции> ПО <Список полей>]

Текст запроса состоит из нескольких секций, каждая из которых начинается с ключевого слова. Секции, заключенные в квадратные скобки, являются необязательными. Вертикальной чертой разделены взаимоисключающие выражения. Назначение секций будет рассмотрено позднее.

Применение запросов в системе 1С:Предприятие:

- построение отчетов;
- получение необходимых данных в алгоритмах;
- получение данных при отладке (служебные запросы для анализа данных, получаемых на разных этапах выполнения алгоритма, и способов их обработки).

Вспомогательным инструментом для создания текстов запросов является *конструктор запроса*, который позволяет в визуальном режиме добавить в запрос необходимые секции и заполнить их, тем самым формируя инструкцию по извлечению данных из БД, понятную платформе.

Принципы и механизмы формирования отчетов

Обязательным требованием к системам автоматизации учета является наличие средств обработки накопленной в системе информации и получения сводных данных в удобном для просмотра и анализа виде.

Для получения сводной выходной информации в системе 1С:Предприятие используются объекты конфигурации Отчеты. Для интерактивной настройки вывода информации (т.е. для создания отчетов без программирования) на уровне платформы (с версии 8.2) заложен механизм – *Система компоновки данных*, или сокращенно *СКД*. Данный механизм интегрирован в объект конфигурации Отчет (свойство Основная схема компоновки данных). СКД возникла из необходимости предоставить пользователю возможность самостоятельно представлять результаты одного и того же запроса в отчете в различных вариантах.

Важно! Основная особенность отчетов на основе СКД: результаты запроса, заложенного в отчет разработчиком, могут быть представлены пользователем в различных вариантах с помощью отборов, сортировки, условного оформления, использования параметров отчета и др.

После создания объекта конфигурации Отчет и отнесения его к существующим подсистемам необходимо вызвать конструктор макета, нажав на кнопку Открыть схему компоновки данных (вкладка Основные мастера отчета). В названном конструкторе создается макет отчета типа Схема компоновки данных. Дальнейшая работа по интерактивному созданию отчета производится в конструкторе схемы компоновки данных, который запускается после создания макета (или при редактировании отчета). Конструктор СКД позволяет описать исходные данные, которые будет использовать отчет: наборы данных (источники данных) и связи между ними, вычисляемые поля, параметры, ресурсы, настройки и др. Результат работы конструктора – схема компоновки данных, полностью описывающая отчет. С поля компоновки данных в дальнейшем работает и разработчик, и пользователь.

В качестве набора данных, который используется в отчете, может выступать Запрос, Объект или Объединение. Если определен набор данных Запрос, на вкладке Наборы данных конструктора СКД доступен конструктор запроса.

На вкладке Таблицы и поля конструктора запроса описывается список таблиц-источников данных и список полей для выборки из БД в запросе. Данная вкладка отвечает за формирование секций ВЫБРАТЬ и ИЗ текста запроса. Список полей выборки может содержать: поля реальных и виртуальных таблиц; выражения (вычисляемые, логиче-

ские, строковые), записанные на языке схемы компоновки данных; вложенные таблицы.

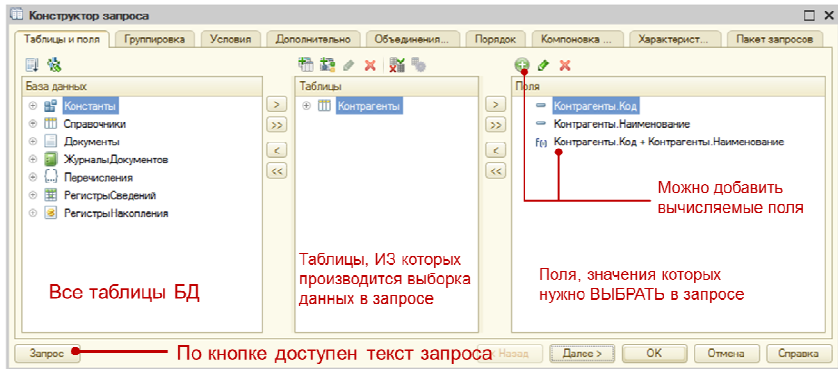


Рис. 64 – Вкладка «Таблицы и поля» конструктора запросов

При описании полей выборки можно присвоить им псевдонимы на вкладке Объединения/Псевдонимы конструктора запроса (колонка Имя поля), что соответствует секции КАК текста запроса.

На вкладке Дополнительно конструктора запроса можно указать ряд настроек для выборки записей:

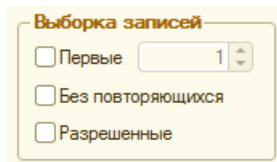


Рис. 65 – Дополнительные параметры выборки записей в запросе

Флаг **Первые** соответствует ключевому слову **ПЕРВЫЕ** в тексте запроса. Конструкция позволяет задать предельное количество строк в результате запроса. Будут отображены самые первые (в соответствии с правилами упорядочивания) строки.

Флаг **Без повторяющихся** соответствует ключевому слову **РАЗЛИЧНЫЕ** в тексте запроса. Конструкция убирает повторяющиеся записи из результата запроса.

Флаг **Разрешенные** соответствует ключевому слову **РАЗРЕШЕННЫЕ** в тексте запроса. Конструкция означает, что запрос выберет только те записи, на которые у текущего пользователя есть права. Если запрос возвращает записи, на которые у пользователя нет прав, генерируется сообщение «У пользователя недостаточно прав на исполнение операции над БД».

В ряде случаев необходимо блокировать считываемые в запросе данные для последующего изменения. Такая возможность реализуется добавлением секции ДЛЯ ИЗМЕНЕНИЯ в текст запроса, что в конструкторе запроса соответствует флагу Блокировать получаемые данные... на вкладке Дополнительно. Считанные данные становятся недоступными для чтения в других сессиях. Блокировка снимается после завершения транзакции.

Для сортировки записей результата запроса по алфавиту или числовому полю используются настройки вкладки Порядок конструктора запроса, которые соответствуют секции УПОРЯДОЧИТЬ ПО текста запроса. Предусмотрено несколько вариантов сортировки: возрастание (по умолчанию), убывание, иерархия, иерархия по убыванию. Действует сортировка по нескольким полям. Настройки Иерархия / Иерархия по убыванию имеют смысл для иерархических таблиц и добавляют в результат запроса группы элементов. При этом сортировка работает как для групп, так и для элементов внутри этих групп. При использовании опции АВТОУПОРЯДОЧИВАНИЕ поля для сортировки результата запроса будут формироваться автоматически по правилам, определенным на уровне платформы.

Для создания простейшего запроса обязательным условием является заполнение вкладки Таблицы и поля конструктора запроса. Как только запрос сформирован, поля запроса автоматически преобразуются в поля схемы компоновки отчета, с которыми может работать и разработчик, и пользователь.

Поля: **Поля схемы компоновки**

Поле	Путь	Ограничение поля	Роль	Выражения...	Проверка иерархии:	Тип значен...	Оформление
	Заголовок	П... У... Г... У...		Выражения упорядочив...	Набор данных	Доступные значения	Параметры редактиров...
		П... У... Г... У...			Параметр		
= Наименов...	Наименование	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					
	<input type="checkbox"/> Наименование	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>					
= Код	Код	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					
	<input type="checkbox"/> Код	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>					
= Поле1	Поле1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					
	<input checked="" type="checkbox"/> Поле1	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>					

Запрос: **Редактирование заголовка поля**

ВЫБРАТЬ
 Контрагенты.Код,
 Контрагенты.Наименование,
 Контрагенты.Код + Контрагенты.Наименование **КАК** Поле1

ИЗ
 Справочник.Контрагенты **КАК** Контрагенты

Конструктор запроса...

Рис. 66 – Поля схемы компоновки данных

В области Поля (конструктор СКД > вкладка Наборы данных) доступно множество опций, например:

- Заголовок полей, выводимых в отчет; опция используется для быстрого редактирования автозаголовка;
- Ограничение поля используется, если необходимо сделать поле недоступным для упорядочивания, группировки и др.;
- Оформление позволяет установить формат поля: цвет текста, шрифт, стиль границы, формат представления чисел/дат, ориентация текста и т.п.

Для вывода в отчет сформированного набора данных необходимо настроить вариант отображения данных (вкладка Настройки конструктора схемы компоновки данных), т.е. указать платформе, как именно будет представлена информация пользователю. По умолчанию для отчета определен один вариант представления данных, его необходимо настроить с помощью Конструктора настроек компоновки данных.

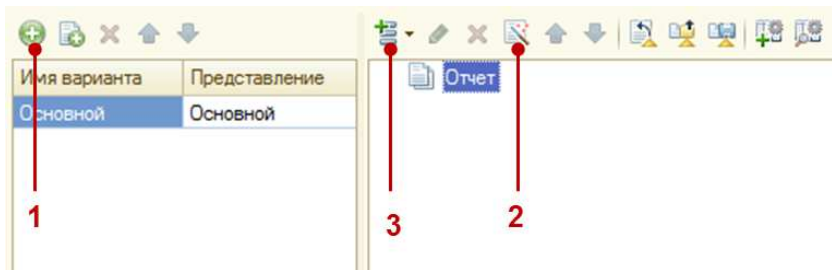


Рис. 67 – Настройки варианта отчета в конструкторе СКД

1 – кнопка для добавления варианта представления отчета;

2 – конструктор настройки текущего варианта отчета;

3 – меню для ручной настройки варианта отчета

На первом шаге в конструкторе настроек следует выбрать тип отчета (список, таблица или диаграмма). Далее нужно выбрать поля, которые будут отображаться в отчете. Для отчета типа Список или Таблица также можно указать поля, по которым будет производиться группировка и упорядочивание данных. Настройка отчета с диаграммой будет рассмотрена позднее.

Представление отчета, созданного на основе СКД, может быть модифицировано в пользовательском режиме и сохранено как новый вариант отчета. Для этого используются команды Изменить вариант... и Сохранить вариант... из меню Еще формы отчета.

Табличная модель получения данных из справочников и документов

В качестве примера простого отчета на основе СКД рассмотрим отчет ПоставщикиНоменклатуры, который должен выводить номенклатур-

ные позиции, сгруппированные по основному поставщику. Перед созданием отчета внесем необходимые изменения в конфигурацию:

- добавим новый реквизит ОсновнойПоставщик в справочник Номенклатура;
- создадим форму элемента справочника Номенклатура;
- для элемента формы, соответствующего реквизиту ОсновнойПоставщик, определим обработчик события НачалоВыбора. В процедуре обработчике обеспечим, чтобы выбор основного поставщика производился только из группы Поставщики справочника Контрагенты. Например, можно использовать следующий подход: открыть программным способом форму выбора справочника Контрагенты, установив в ней отбор по Родителю.

```
Процедура ОсновнойПоставщикНачалоВыбора
    (Элемент, ДанныеВыбора, СтандартнаяОбработка)
СтандартнаяОбработка = Ложь;
ЗначениеОтбора = Новый Структура
    ("Родитель", ПолучитьСсылкуПоставщиков());
ПараметрыВыбора = Новый Структура
    ("Отбор", ЗначениеОтбора);
ОткрытьФорму("Справочник.Контрагенты.ФормаВыбора",
    ПараметрыВыбора, Элемент);
КонецПроцедуры
```

где ПолучитьСсылкуПоставщиков() – серверная функция, возвращающая ссылку на группу с наименованием Поставщики справочника Контрагенты. **p.s.** С листингом разобраться самостоятельно!

Перейдем к созданию отчета на основе СКД. Набор данных - запрос к таблице Контрагенты, поля выборки – Основной поставщик и Ссылка (стандартный реквизит, уникально идентифицирующий любой объект). Добавление поля Ссылка в список полей выборки имеет в готовом отчете следующий эффект: в пользовательском режиме из строки отчета возможно открыть форму объекта, который идентифицируется по ссылке.

На вкладке Объединения/Псевдонимы конструктора запроса определим для поля Ссылка более понятное имя – Номенклатура.

Наложим несколько условий на результат запроса (вкладка Условия конструктора запроса). Заданные условия отобразятся в секции ГДЕ текста запроса. Механизм следующий: в результат запроса попадают только те записи, для которых условия отбора оказываются истинными. Условия отбора начинают действовать после того, как из таблицы-источника выбраны все записи. Поля-условия можно не указывать в полях выборки, а задействовать только в секции с условиями. В усло-

виях отбора можно использовать логические операторы НЕ, И, ИЛИ, а также формировать произвольные условия на языке выражений системы компоновки данных (с помощью конструктора произвольных выражений, который осуществляет синтаксический контроль выражений). Подробнее про язык выражений – справка 1С:Предприятия > Система компоновки данных > Язык выражений системы компоновки данных.

Условимся, что в отчет не должны попадать группы номенклатуры. Добавляем на вкладку *Условия* конструктора запроса первую запись:

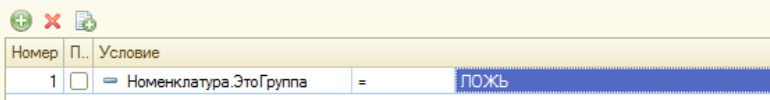


Рис. 68 – Условие по отбору в запросе элементов справочника

Если оставить условие в таком виде, то платформа воспримет ЛОЖЬ как параметр запроса, а не как значение типа Булево. Поэтому необходимо преобразовать первое условие в произвольное выражение, установив флаг П. в конструкторе.

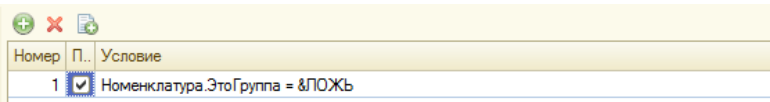


Рис. 69 – Условие в виде произвольного выражения

Параметры в тексте запроса задаются через &. В данном случае следует удалить & из выражения.

Также условимся, что в отчет не должны попадать номенклатурные позиции, для которых не задан основной поставщик. Зададим второе произвольное условие, в котором будем сравнивать значение поля ОсновнойПоставщик со значением пустой ссылки (с помощью функции ЗНАЧЕНИЕ языка выражений компоновки данных).

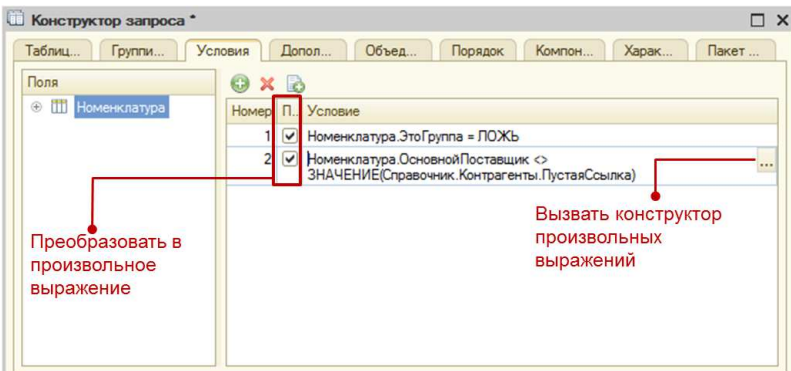
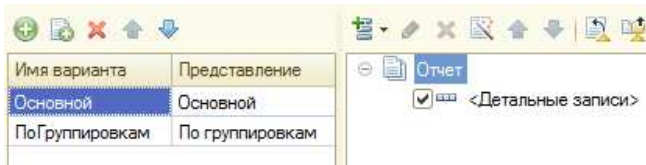


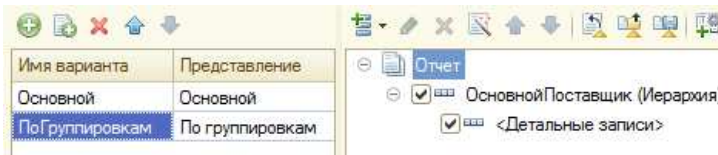
Рис. 70 – Условия, сформированные в конструкторе произвольных выражений

На этом работа в конструкторе запроса завершена. Переходим в конструктор СКД на вкладку Настройки для формирования варианта отображения отчета. Вызвав конструктор настроек схемы компоновки данных, выбираем Список в качестве типа отчета. Далее укажем, что в отчет нужно выводить поля Номенклатура и ОсновнойПоставщик.

Создадим второй вариант отчета - ПоГруппировкам. В конструкторе настроек схемы компоновки данных сделаем аналогичные действия. Дополнительно укажем поле ОсновнойПоставщик в качестве поля, по которому будет производиться группировка номенклатурных позиций (тип группировки Иерархия).



(a)



(б)

Рис. 71 – Структура вариантов отчета Поставщика номенклатуры

(а) – основной вариант отчета;

(б) – вариант отчета с группировкой по поставщику

Между созданными вариантами отчета можно переключаться в пользовательском режиме:

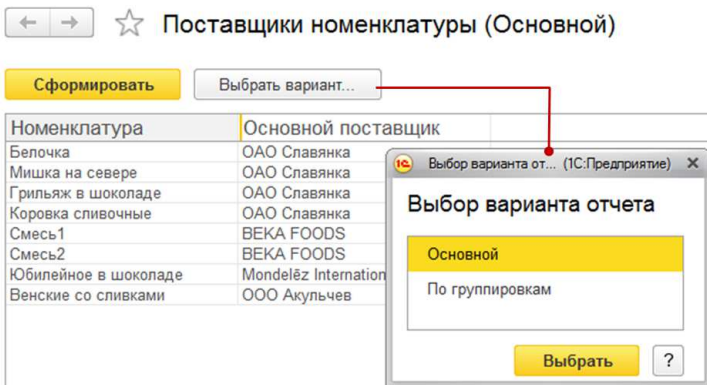


Рис. 72 – Выбор варианта отчета в пользовательском режиме

← → ☆ Поставщики номенклатуры (По группировкам)

Сформировать		Выбрать вариант...	
		Основной поставщик	
		Номенклатура	
[-]		Поставщики	
[-]		BEKA FOODS	
		Смесь1	
		Смесь2	
[-]		Mondelēz International	
		Юбилейное в шоколаде	
[-]		ОАО Славянка	
		Белочка	
		Мишка на севере	
		Грильяж в шоколаде	
		Коровка сливочные	
[-]		ООО Акульчев	
		Венские со сливками	

Рис. 73 – Второй вариант отчета Поставщики номенклатуры (с группировкой)

В отчете Поставщики номенклатуры (вариант По группировкам) на первом уровне отображается группа Поставщики справочника Контрагенты, на втором уровне – элементы этой группы, на третьем уровне – номенклатурные позиции.

В предыдущей работе рассматривался принцип учета «от документа» в системе 1С:Предприятие. В соответствии с этим принципом данные о хозяйственной деятельности обрабатываются по схеме Документы > Регистры > Отчеты. Поэтому для системы учета, реализованной на платформе 1С:Предприятие, корректным подходом является построение отчетов по данным регистров. И только в учебных целях в лабораторной работе будут приведены примеры и задания по построению отчетов на основе данных из документов.

Выполняя запросы к документам и их табличным частям (ТЧ), нужно помнить о том, что документ с ТЧ представлен в базе несколькими таблицами (каждая табличная часть хранится в виде отдельной таблицы). Общая рекомендация такова: если необходимо получить данные и из документа, и из его табличной части, то в качестве источника данных в запросе следует выбирать таблицу нижнего уровня (т.е. таблицу, соответствующую табличной части документа). В такой таблице всегда присутствует поле Ссылка (на документ, которому принадлежит табличная часть). Путем *разыменования* этого поля можно обратиться к реквизитам верхнего уровня (т.е. к реквизитам самого документа) и добавить их в список полей выборки. В ряде случаев разыменование позволяет избежать использования в запросе нескольких таблиц. При наличии нескольких таблиц-источников данных воз-

никает необходимость установки связей между ними (в конструкторе запросе появляется вкладка Связи).

Рассмотрим ряд возможностей конструктора запроса и конструктора схемы компоновки данных, которые часто применяются при подготовке отчетов.

Например, на вкладке Условия конструктора запроса определено ограничение на дату документа РеализацияТоваровУслуг (обращаемся к дате документа путем разыменования поля Ссылка табличной части Товары этого документа):

```
РеализацияТоваровУслугТовары.Ссылка.Дата      |      МЕЖДУ      |
ДатаНачала | ДатаКонца
```

В данном случае ДатаНачала и ДатаКонца являются параметрами, которые определил разработчик. В тексте запроса ограничение будет выглядеть следующим образом:

ГДЕ

```
РеализацияТоваровУслугТовары.Ссылка.Дата      МЕЖДУ
&ДатаНачала И &ДатаКонца
```

Как только в конструкторе запроса определены параметры, они автоматически появляются и в схеме компоновки данных (вкладка Параметры конструктора схемы компоновки). На вкладке Параметры можно изменить тип параметра, снять Ограничение доступности (если параметр должен быть доступен пользователю), сделать параметр обязательным для заполнения (флаг Запрещать незаполненные значения). Если предполагается, что пользователь должен работать с параметрами в форме отчета, то их необходимо включить в пользовательские настройки. Это действие выполняется на вкладке Настройки конструктора схемы компоновки в одноименном разделе:

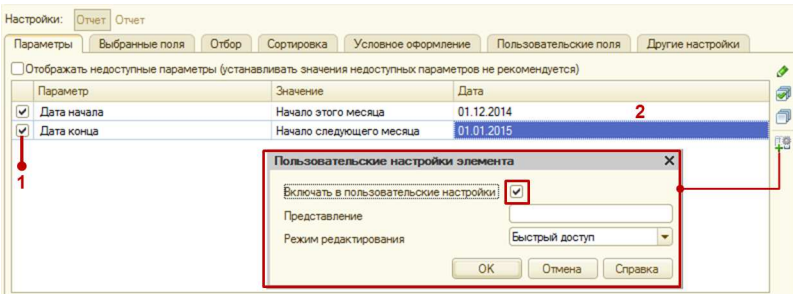


Рис. 74 – Включение параметров схемы компоновки в пользовательские настройки

- 1 – флажки, отвечающие за активность полей с параметрами;
- 2 – поля для установки значений по умолчанию для параметров

При использовании ограничений по периоду нужно помнить, что отчет формируется по начало дня, указанного в параметре (штатное поведение платформы). Обычно же пользователи ожидают, что отчет формируется по указанную дату включительно. Тогда параметр, соответствующий концу периода, за который формируется отчет, можно принудительно привести к концу дня (на вкладке Параметры конструктора схемы компоновки, поле Выражение). Как упоминалось ранее, выражение составляется на языке выражений системы компоновки данных. Следующее выражение позволяет привести дату, переданную в запрос через параметр КонецПериода, к концу дня:

КонецПериода (&КонецПериода, "День")

На вкладке Ресурсы конструктора схемы компоновки данных можно перевести числовые поля схемы компоновки в ресурсы. Это означает, что система компоновки данных сможет выполнять арифметические операции (сумма, среднее, максимум, минимум, количество) над этими полями и автоматически формировать итоги по указанным полям.

Получение данных из регистров для построения отчетов

Ранее отмечалось, что на практике источником данных для построения отчетов являются регистры. Например, преимущество регистров накопления в том, что они хранят просуммированные данные, что облегчает получение остатков и оборотов по каким-либо показателям.

Для любого регистра на уровне платформы заложен набор виртуальных таблиц, которые можно использовать в запросах: СрезПервых и СрезПоследних для регистра сведений; Обороты для регистра накопления вида Обороты; Обороты, Остатки и ОстаткиИОбороты для регистра накопления вида Остатки. Физическая таблица содержит данные обо всех движениях регистра. При построении отчета часто требуются остатки, обороты показателей за конкретный период, а не все движения регистра. Остатки и обороты как раз можно получить из виртуальных таблиц. Виртуальные таблицы регистров физически не хранятся в БД, а формируются автоматически (в зависимости от своих параметров) в момент запроса к ним.

Рассмотрим задачу построения отчета, показывающего сумму долга каждого клиента перед организацией и долгов организации перед поставщиками на заданную дату. Предварительно потребуется внести изменения в рабочую конфигурацию:

- создать документ ВыпискаБанка с двумя табличными частями - ПриходДенежныхСредств и РасходДенежныхСредств; реквизиты табличных

- частей – Контрагент и Сумма; строки первой табличной части будут отражать уменьшение задолженности некоторого клиента перед организацией, а строки второй табличной части – уменьшение задолженности организации перед поставщиками;
- создать регистр накопления Взаиморасчеты с видом Остатки, измерением Контрагент и ресурсом Сумма; регистраторы – документы ПоступлениеТоваров, РеализацияТоваровУслуг и ВыпискаБанка;
 - настроить проведение документа ВыпискаБанка по регистру Взаиморасчеты: тип движения по табличной части ПриходДенежныхСредств – Расход, по второй табличной части – Приход;
 - настроить проведение документа ПоступлениеТоваров по регистру Взаиморасчеты:

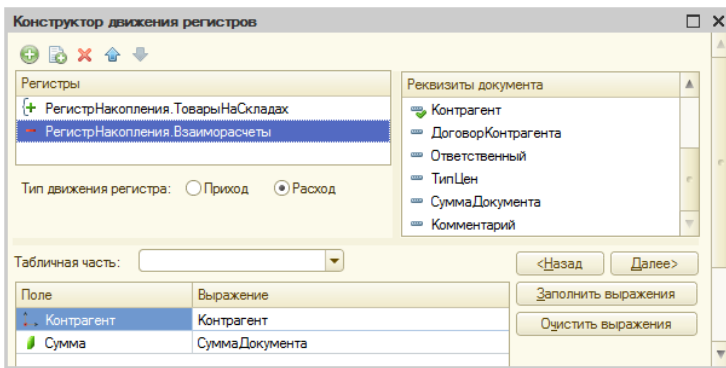


Рис. 75 – Создание движений регистра Взаиморасчеты

Аналогичную рисунку 75 настройку нужно выполнить и для последнего документа-регистратора, но тип движения регистра – Приход.

Добавим в состав конфигурации новый объект – отчет ОстаткиВзаиморасчетов. Создадим для него схему компоновку данных и набор данных Запрос. Дальнейшие действия будем выполнять в конструкторе запроса.

В конструкторе запроса (вкладка Таблицы и поля > область База данных) регистр накопления Взаиморасчеты представлен сразу четырьмя таблицами: физическая таблица записей регистра и виртуальные таблицы. В соответствии с постановкой задачи выходной информацией являются остатки долгов в разрезе контрагентов. Поэтому источником данных для отчета будет выступать виртуальная таблица Взаиморасчеты.Остатки, в которой определено поле СуммаОстаток (постфикс Остаток означает, что по полю Сумма регистра подсчитываются остатки).

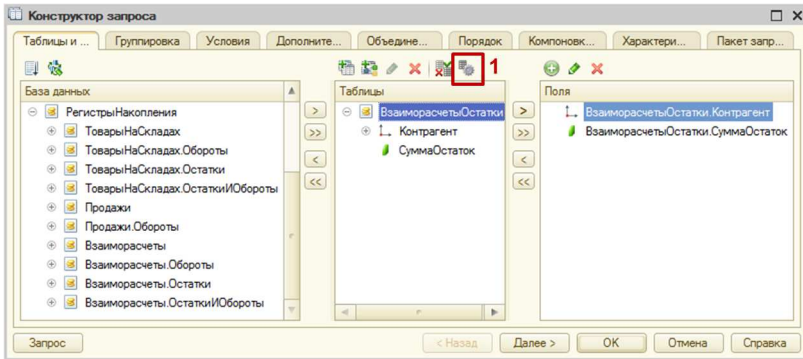


Рис. 76 – Создание запроса для отчета Остатки взаиморасчетов
1 – доступ к параметрам виртуальной таблицы регистра

Виртуальные таблицы регистров имеют предопределенные параметры. Для виртуальной таблицы остатков определены параметры Период и Условие:

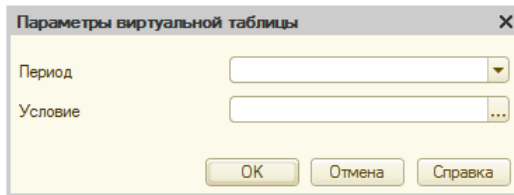


Рис. 77 – Параметры виртуальной таблицы остатков

Далее зададим псевдоним Долг для поля СуммаОстаток (вкладка Объединения/Псевдонимы конструктора запроса). Сохраняем запрос и возвращаемся в конструктор схемы компоновки данных. Обеспечим вывод в отчет общих итогов: перенесем поле Долг в ресурсы (вкладка Ресурсы конструктора СКД).

Остается обеспечить в отчете ОстаткиВзаиморасчетов получение остатков на заданную пользователем дату. Без этого параметра остатки будут выдаваться на момент последней операции в системе. В параметрах виртуальной таблицы (рис.77) присутствует параметр Период, его и будем использовать, чтобы пользователь мог выбирать период, на который нужно сформировать остатки. Данный параметр по умолчанию отображается в списке на вкладке Параметры конструктора СКД. Оставим для параметра тип Дата, но изменим состав даты с Дата и время на Дата, чтобы пользователю не пришлось задавать значение параметра с точностью до секунд.

Получение остатков из регистра накопления на заданную дату имеет несколько особенностей, которые определяются штатным поведением платформы 1С:Предприятие.

Во-первых, остатки формируются на начало периода, т.е. на начало дня, указанного в параметре Период, что совсем не очевидно для пользователя. Поэтому принудительно будем приводить значение параметра к концу дня:

КонецПериода (&Период, "День")

Функция КонецПериода языка выражений системы компоновки данных с параметром День преобразует дату с 0:00:00 ко времени 23:59:59.

Во-вторых, все записи регистра накопления (вида Остатки), которые имеют период с точностью до секунды, условно говоря, на оси времени находятся старше этой секунды (между текущей секундой и следующей). Поэтому когда мы получаем остатки на конкретную секунду, система не включает сюда все данные, проведенные в рамках этой секунды. Все записи, имеющие конкретное время t с точностью до секунд, они старше этого времени, но младше следующей секунды, поэтому для получения корректного остатка необходимо получать данные на следующую секунду. С учетом вышесказанного выражение для преобразования параметра Период отчета ОстаткиВзаиморасчетов выглядит следующим образом:

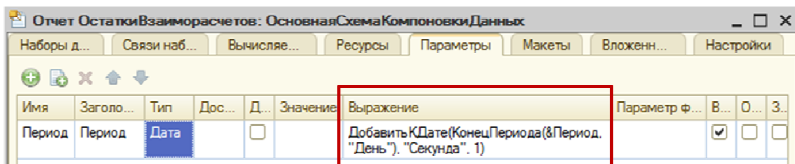


Рис. 78 – Выражение для преобразования параметра Период

Остается настроить представление отчета (в виде обычного списка без группировок) и включить параметр Период в пользовательские настройки (на вкладке Настройки конструктора СКД). Окончательный вид отчета см. в п.5 содержания работы. В отчете присутствуют положительные числа (дебиторская задолженность, т.е. долг клиентов перед организацией) и отрицательные числа (кредиторская задолженность, т.е. долг организации перед поставщиками).

Для оборотных регистров накопления предопределена одна виртуальная таблица - Обороты, из которой можно получить обороты ресурсов за определенный период. Виртуальная таблица Обороты получает данные по конец секунды (не нужно принудительно добавлять одну секунду как при получении данных из виртуальной таблицы Остатки

регистра накопления). Виртуальная таблица Обороты имеет следующие параметры:

Рис. 79 – Параметры вирт. таблицы Обороты оборотного регистра накопления

В параметрах виртуальных таблиц допускается использование подзапросов (вложенных запросов), что позволяет организовать фильтрацию выходных данных. Используем этот подход для решения усложненной задачи получения данных о продажах товара: в отчет должна попадать информация о продажах тех товаров, которые содержатся в табличной части документа РеализацияТоваровУслуг, указанного в качестве параметра в форме отчета.

Для решения поставленной задачи можно использовать параметр Условие виртуальной таблицы Продажи.Обороты регистра накопления Продажи (типа Обороты). Условие можно накладывать на измерения в редакторе произвольных выражений, который вызывается из поля параметра. Для решаемой задачи условие следующее: номенклатура, включаемая в отчет, должна входить в список товаров табличной части выбранного документа. Условие запишем на языке выражений системы компоновки данных, используя оператор В (оператор проверки вхождения в список). Для получения списка товаров обратимся в подзапросе к табличной части документа. Для написания подзапроса удобнее всего воспользоваться конструктором запроса.

Рис. 80 – Заполнение параметра Условие виртуальной таблицы регистра

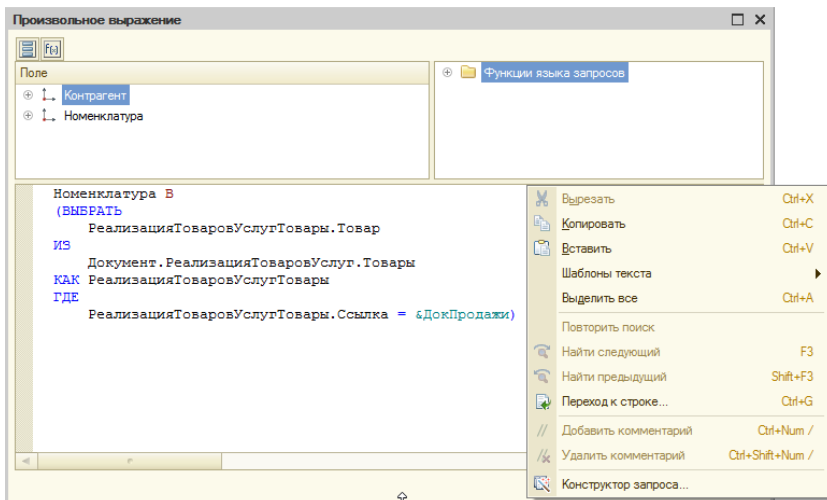


Рис. 81 – Вложенный запрос в параметре виртуальной таблицы

Для того чтобы параметр ДокПродажи вложенного запроса корректно распознавался, на вкладке Параметры конструктора СКД нужно назначить ему тип ДокументСсылка.РеализацияТоваровУслуг.

Ограничение по номенклатуре можно было бы задать и через вкладку Условия конструктора запроса, не используя вложенный запрос. Но общая рекомендация такова: условия на измерения регистров нужно задавать в параметрах виртуальной таблицы. Это требование обусловлено вопросами производительности (скорости формирования отчета).

Построение отчетов типа Таблица и Диаграмма

Система компоновки данных позволяет формировать отчеты, в которых одна и та же информация может быть представлена в разных разрезах и разных формах представления (в виде линейных списков, списков с группировками, таблиц, диаграммам).

Наряду со списками еще одним способом вывода информации в отчет являются таблицы (кросс-таблицы). В таких таблицах измерения регистра располагаются в строках и колонках, на их пересечении выводятся значения ресурсов.

Условимся, что в конфигурации присутствует отчет, отражающий обороты продаж товаров (в количественном и суммовом выражении) за указанный период в разрезе контрагентов. Сформируем в конструкторе СКД настройку варианта отчета в виде таблицы (шаг 1). Для это-

го в схеме компоновки данных должны быть определены ресурсы, иначе на пересечении строк и колонок нечему будет выводиться.

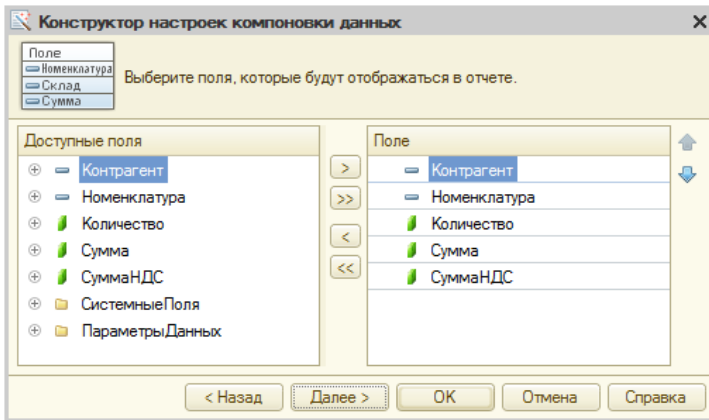


Рис. 82 – Шаг 2: выбор полей, которые будут отображаться в таблице

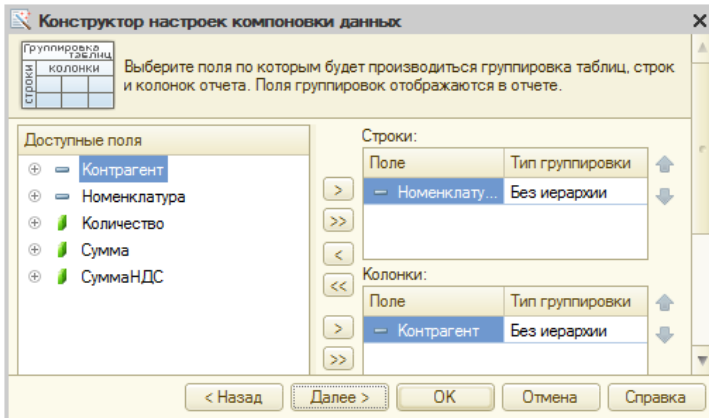


Рис. 83 – Шаг 3: выбор полей (измерений регистра), которые будут отображаться в строках и колонках таблицы

Отметим, что на третьем шаге ресурсы не выбираем, они автоматически будут выводиться на пересечении строк и колонок. Нажимаем ОК.

Усложняем задачу. В рамках этого же отчета нужно получить второй вариант представления данных, не прибегая к созданию нового варианта отчета. Второй вариант представления выводных данных должен отражать продажи по клиентам, без разделения на номенклатуру (в виде линейного списка). Конструктор СКД позволяет формировать сколько угодно способов представления данных отчета в рам-

ках одного варианта. В предыдущих примерах описывалось использование конструктора настроек. Он позволяет сформировать первый способ представления выходной информации, все последующие способы формируются вручную путем добавления новых группировок, таблиц, диаграмм.

Важно. Поля, доступные для корневого элемента варианта отчета, доступны и для всех подчиненных элементов.

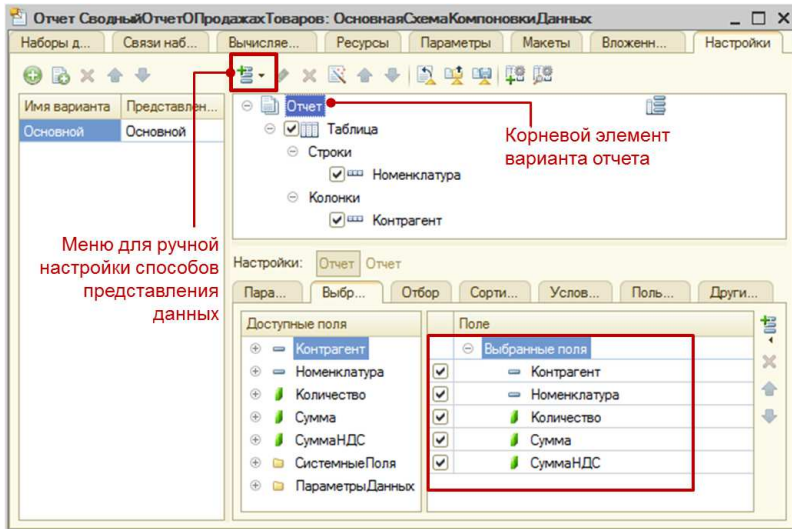


Рис. 84 – Настройка способов представления выходной информации

Используя меню для ручной настройки способов представления выходной информации, добавим новую группировку, подчиненную корневому элементу Отчет:

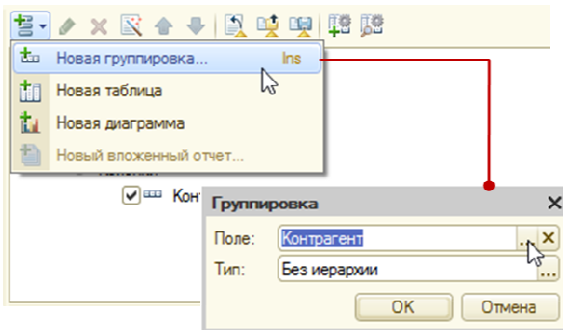


Рис. 85 – Создание вручную второго способа представления выходных данных

Рис. 88 – Структура диаграммы

Для добавления диаграммы в конструкторе настроек нужно выбрать в качестве типа отчета соответствующий вариант. На втором шаге указываются поля выборки, которые будут отображаться в отчете (на этом этапе обязательно нужно выбрать ресурс в числе прочих полей). На третьем шаге указываются поля, по которым будет производиться группировка серий и точек диаграммы. Далее при необходимости указываются поля упорядочивания. На последнем шаге выбирается тип диаграммы. Система 1С:Предприятие предлагает обширный набор типов диаграмм: от стандартных (график, гистограмма, круговая диаграмма) до специфических (воронки продаж, биржевые диаграммы). Тип диаграммы по умолчанию – объемная гистограмма.

Если возникает необходимость изменить тип диаграммы, необходимо спозиционироваться на ней в структуре отчета, щелкнуть по имени диаграммы в строке Настройки и перейти на вкладку Другие настройки (параметр Тип диаграммы).

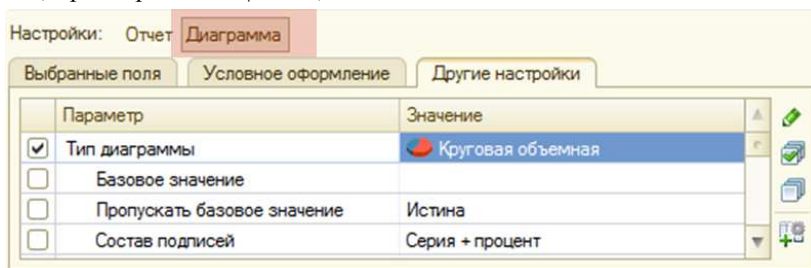


Рис. 89 – Изменение типа диаграммы

Параметры виртуальных таблиц регистров можно применять не только для фильтрации выходных данных, но и при построении диаграмм. Продемонстрируем на примере построения диаграммы типа График. Такие диаграммы, как правило, строятся в разрезе оси времени.

Задача: сформировать график продаж по дням в разрезе клиентов. Обратимся к виртуальной таблице Продажи.Обороты. Полей СуммаОборот и СуммаНДСОборот не достаточно, ведь нас интересует сумма продаж на каждый день. В параметрах виртуальной таблицы присутствует Периодичность. Выбрав подходящий вариант, мы обеспечим, что продажи будут собираться в разрезе этой периодичности. При Периодичность = День в результат запроса будут попадать только те дни, за которые были продажи. С учетом сказанного запрос на выборку данных имеет вид:

ВЫБРАТЬ

ПродажиОбороты.Контрагент,

ПродажиОбороты.СуммаОборот +
 ПродажиОбороты.СуммаНДСОборот КАК Всего,
 ПродажиОбороты.Период

ИЗ

РегистрНакопления.Продажи.Обороты(, , День,)
 КАК ПродажиОбороты

Настроенный отчет имеет следующий вид:

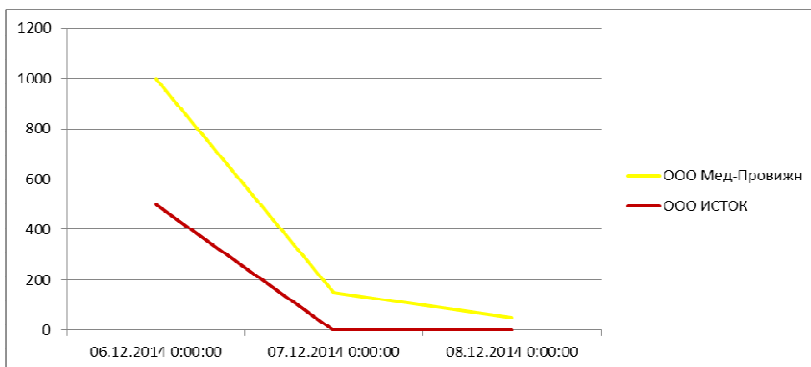


Рис. 90 – Диаграмма типа «График с накоплением», отражающая суммы продаж

Использование запросов в алгоритмах

Общая схема работы с запросом из встроенного языка:

```
Запрос = Новый Запрос;
// с помощью конструктора Новый создаем объект встроенного
// языка типа Запрос.
// тип данных Запрос предназначен для создания статических
// запросов, тип ПостроительОтчета – для динамических.
Запрос.Текст = "ВЫБРАТЬ...";
// через свойство Текст указываем текст запроса
РезультатЗапроса = Запрос.Выполнить();
// запрос выполняется с помощью метода Выполнить(),
// на выходе получаем тип данных РезультатЗапроса.
// РезультатЗапроса – таблица.
Если НЕ РезультатЗапроса.Пустой() Тогда
// посредством типа данных РезультатЗапроса можно проверить,
// является ли результат пустой таблицей.
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
// с помощью метода Выбрать() курсор устанавливается на
// первую строку результат запроса.
// на выходе получаем тип данных ВыборкаИзРезультатаЗапроса.
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
// с помощью метода Следующий() последовательно обходим все
// строки результата в цикле Пока
```

Типовой код по обработке результата запроса можно сформировать, вызвав из контекстного меню программного модуля конструктор запроса с обработкой результата.

Используем механизм запросов для того, чтобы контролировать достаточность товара на складе при реализации товаров. Потребуется модифицировать процедуру `ОбработкаПроведения` из модуля объекта документа `РеализацияТовароУслуг`. Схема решения:

- записать данные в регистры;
- выполнить запрос на поиск отрицательных остатков в регистре накопления `ТоварыНаСкладах`;
- если отрицательные остатки найдены, документ не должен быть проведен; необходимо выдавать сообщение пользователю, какого товара и в каком количестве не хватает.

Процедура `ОбработкаПроведения` имеет два параметра: `Отказ` (используя этот параметр, можем отказаться от проведения документа) и `Режим` (параметр показывает, как документ проводится: оперативно – т.е. проведение осуществляется в реальном времени и дата документа устанавливается равной текущему времени, или неоперативно – задним числом). Идеальная система учета – та, в которой все документы проводятся в оперативном режиме. Будем вести контроль остатков только для оперативно проводимых документов.

В штатном режиме движения попадают в регистр по окончании исполнения обработчика `ОбработкаПроведения` (неявные штатные действия платформы). Для решения задачи контроля остатков товаров по предложенной схеме необходимо принудительно записать движения до окончания исполнения обработчика.

Метод `Записать()` коллекции `Движения` вызовет принудительную запись тех наборов записей, для которых свойство `Записывать` установлено в `Истина`.

Модифицированная процедура `ОбработкаПроведения` имеет следующий вид:

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// формирование коллекции Движения для регистров
// ТоварыНаСкладах, Продажи, Взаиморасчеты
//}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
Движения.Записать();
// принудительная запись движений в регистры
Если Режим = РежимПроведенияДокумента.Оперативный Тогда
// контролируем остатки, только если документ проводится
// в оперативном режиме
```

```

// пишем текст запроса на поиск отрицательных остатков
РезультатЗапроса = Запрос.Выполнить();
Если НЕ РезультатЗапроса.Пустой() Тогда
    Отказ = Истина; // отказываемся от проведения документа
    ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
        // выводим пользователю сообщение: сколько и каких
        // товаров не хватает на складе
    КонецЦикла;
КонецЕсли;
КонецЕсли;
КонецПроцедуры

```

Рекомендации по созданию запроса, находящего отрицательные остатки в регистре накопления ТоварыНаСкладах:

- используйте конструктор запроса с обработкой результата, тип обработки – обход результата;
- источник данных – виртуальная таблица Остатки регистра накопления ТоварыНаСкладах;
- достаточно трех полей выборки – Товар, Склад, КоличествоОстаток;
- используйте параметр Условие виртуальной таблицы, чтобы наложить ограничение на измерения регистра; в Условиестройте подзапрос. Через подзапрос нужно получить список товаров табличной части проводимого документа. Список нужен чтобы проанализировать, остатки каких товаров и с какого склада следует проверить;
- для запроса нужно определить условие КоличествоОстаток < 0, т.к. нас интересуют отрицательные остатки.

Обработки

Для выполнения различных действий над информацией в системе 1С:Предприятие используются объекты конфигурации, называемые Обработка. Концептуальным отличием отчетов и обработок является то, что в случае работы последних информация, хранящаяся в системе, претерпевает какие-либо изменения.

Обработка – прикладной объект конфигурации, предназначенный для реализации различных обработок информации и сервисных функций.

Обработки содержат: алгоритмы обработки и получения информации; формы для ввода параметров работы алгоритмов и представления результатов; макеты для вывода результатов выполнения алгоритма в табличный документ.

Обработки бывают внешние (*.erf) и входящие в состав конфигурации. Любую обработку можно преобразовать во внешнюю, и добавить внешнюю обработку в состав конфигурации.

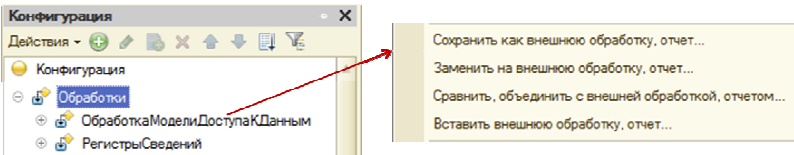


Рис. 91 – Команды контекстного меню для преобразования обработок во внешние файлы и наоборот

Для обеспечения целостности конфигурации внешние обработки рекомендуется использовать, в основном, в отладочных целях. После отладки алгоритма обработки необходимо включить в состав конфигурации.

Примеры использования обработок:

- загрузка/выгрузка данных для обмена с другими приложениями;
- регламентные операции над данными БД (например, удалить помеченные объекты, обновить полнотекстовый индекс ...);
- групповые действия над экземплярами объектов (например, перенести элементы справочника из одной группы в другую ...).

Содержание работы

1. Добавьте в состав конфигурации новую подсистему (Имя = Отчеты, Синоним = Отчеты по организации), которая будет предоставлять пользователю доступ ко всем объектам конфигурации типа Отчет.
2. Создайте простейший отчет с именем ОтчетПоКонтрагентам. В отчете выводится список всех контрагентов из одноименного справочника.

← → ☆ **Отчет по контрагентам**

Сформировать

Код	Наименование
000000001	Поставщики
000000002	Покупатели
000000003	ООО ИСТОК
000000004	ООО Мед-Провижн
000000005	ОАО Славянка
000000006	BEKA FOODS
000000007	Караван
000000008	Mondelēz International
000000009	ООО Акульчев

Рис. 92 – Форма отчета по контрагентам

3. Выполните пример по созданию отчета ПоставщикиНоменклатуры (с двумя вариантами представления данных).
4. Создайте отчет с именем ПродажиТоваровПоДокументам. Отчет отражает данные о продажах товаров (в количественном и суммовом выражении) в разрезе клиентов и номенклатурных позиций за заданный пользователем период.

← → ☆ **Продажи товаров (по данным документов)** ×

Сформировать

Начало периода: ☒ Начало этого месяца Конец периода: ☒ 08.12.2014

Контрагент	Товар	Количество	Сумма	Сумма НДС
[-] ООО ИСТОК	Белочка	11	1 425,00	256,50
	Юбилейное в шоколаде	6	1 200,00	216,00
	Юбилейное в шоколаде	5	225,00	40,50
[-] ООО Мед-Провижн	Венские со сливками	6	480,00	86,40
	Венские со сливками	6	480,00	86,40
Итого		17	1 905,00	342,90

Рис. 93 – Форма отчета Продажи товаров

Источник данных для построения отчета – табличные части документов РеализацияТоваровУслуг. Данные о продажах товаров из непроведенных документов не должны попадать в отчет.

Границы периода, за который нужно сформировать отчет, передаются в запрос через параметры НачалоПериода и КонецПериода (тип Дата, состав Дата). Необходимо предусмотреть автоматическое приведение даты, заданной в поле КонецПериода, к концу дня.

Вариант отчета содержит группировки по клиенту и номенклатуре, а также итоги по каждому клиенту и в целом по всему отчету.

5. Выполните пример по созданию отчета ОстаткиВзаиморасчетов.

Контрагент	Долг
ВЕКА FOODS	-4 248,00
ООО Акульчев	-708,00
ООО ИСТОК	1 681,50
ООО Мед-Провижн	-743,40
ОАО Славянка	-3 136,00
Итого	-7 153,90

Рис. 94 – Форма отчета Остатки взаиморасчетов

6. Создайте отчет с именем ПродажиТоваров. В п.4 содержания работы для получения данных о продажах были использованы табличные части документов. Реализуйте задание более корректным способом (на основе виртуальной таблицы соответствующего регистра накопления). Форма отчета аналогична рис.93.

Рекомендации по разработке отчета. На результат запроса наложите условие: в отчете не должны отражаться суммы за оказание услуг. Все числовые поля выборки переведите в ресурсы. Предусмотрите возможность получения в отчете оборотов продаж за определенный период (используйте параметры НачалоПериода и КонецПериода виртуальной таблицы Обороты). Тип отчета - список, по полю Контрагент производится группировка данных.

7. Создайте отчет с именем ПродажиНоменклатурыДокумента, реализующий вывод информация о продаж (в количественном и суммовом выражении) тех товаров, которые содержатся в табличной части документа РеализацияТоваровУслуг, указываемого в качестве параметра

в форме отчета. Используйте подзапрос в параметрах виртуальной таблицы (рис. 80, 81).

← → ☆ Продажи номенклатуры документа ×

Сформировать

Начало периода: ☐ Из документа: ☒ Реализация товаров и услуг ▾

Конец периода: ☐

Номенклатура	Количество	Сумма	Сумма НДС
Юбилейное в шоколаде	5	225,00	40,50
Белочка	6	1 200,00	216,00
Итого	11	1 425,00	256,50

Рис. 95 – Форма отчета Продажи номенклатуры документа

8. Добавьте в состав конфигурации новый отчет копированием отчета, созданного по п.6. Имя нового объекта – СводныйОтчетОПродажахТовара. Отчет содержит один вариант с тремя способами представления информации о продажах: 1) в виде таблицы, по строкам – товары, по колонкам – контрагенты; 2) в виде линейного списка, по клиентам и без разделения на номенклатуру; 3) в виде списка, по клиенту и в разрезе номенклатуры.

Таблица продаж

Номенклатура	ООО ИСТОК			ООО Мед-Провижн			Итого		
	Количество	Сумма	Сумма НДС	Количество	Сумма	Сумма НДС	Количество	Сумма	Сумма НДС
Белочка	6	1 200,00	216,00				6	1 200,00	216,00
Ванские со сливками				6	480,00	86,40		480,00	86,40
Юбилейное в шоколаде	5	225,00	40,50				5	225,00	40,50
Итого	11	1 425,00	256,50	6	480,00	86,40	17	1 905,00	342,90

Продажи в разрезе клиентов

Контрагент Количество Сумма Сумма

Рис. 96 – Форма отчета Сводный отчет о продажах товаров (часть 1)

Продажи в разрезе клиентов

Контрагент	Количество	Сумма	Сумма НДС
ООО ИСТОК	11	1 425,00	256,50
ООО Мед-Провижн	6	480,00	86,40
Итого	17	1 905,00	342,90

Продажи в разрезе клиентов и товаров

Контрагент Количество Сумма Сумма

Рис. 97 – Форма отчета Сводный отчет о продажах товаров (часть 2)

Продажи в разрезе клиентов и товаров			
Контрагент	Количество	Сумма	Сумма НДС
Номенклатура			
ООО ИСТОК	11	1 425,00	256,50
Юбилейное в шоколаде	5	225,00	40,50
Белочка	6	1 200,00	216,00
ООО Мед-Провижн	6	480,00	86,40
Венские со сливками	6	480,00	86,40
Итого	17	1 905,00	342,90

Рис. 98 – Форма отчета Сводный отчет о продажах товаров (часть 3)

9. Создайте новый отчет ПродажиДиаграмма. Источник данных – виртуальная таблица оборотного регистра накопления Продажи. Поля выборки: Контрагент, Номенклатура, СуммаВключаяНДС (вычисляемое поле, получаемое как сумма полей СуммаОборот и СуммаНДСОборот). Отчет предполагает один вариант представления данных, включающий три диаграммы, которые позволят анализировать данные о продажах в различных разрезах:

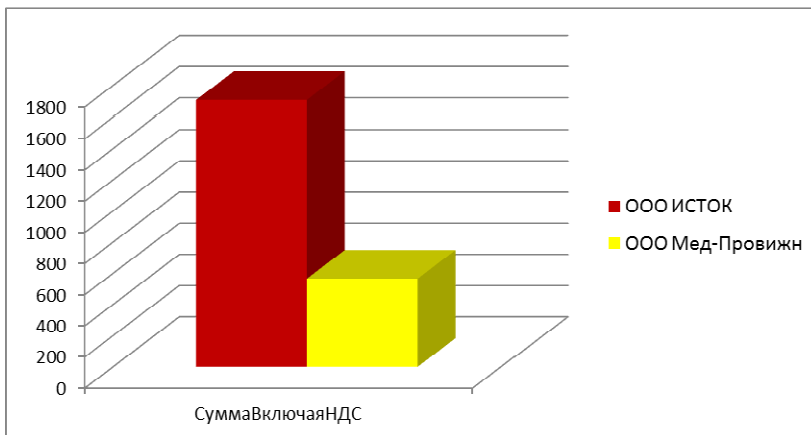


Рис. 99 – Объемная тистограмма, показывающая сумму выручки по клиентам

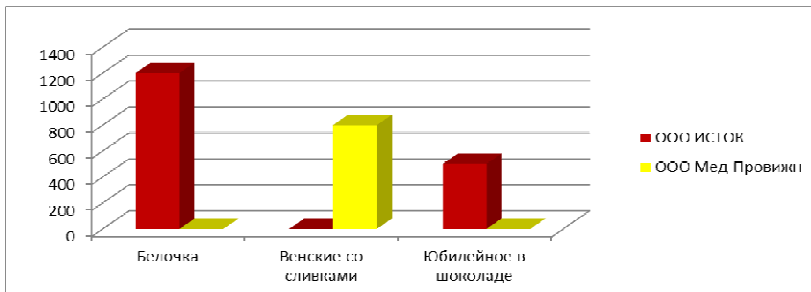


Рис. 100 – Объемная тистограмма (продажи по клиентам в разрезе номенклатуры)

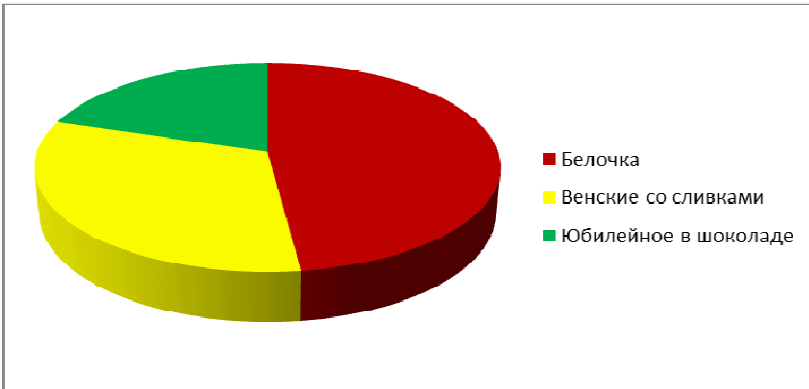


Рис. 101 – Круговая диаграмма (продажи в разрезе номенклатуры)

10. Реализуйте отчет ГрафикПродаж по примеру (рис.90). Отчет отражает график продаж по дням в разрезе клиентов.

11. Решите задачу контроля остатков товаров: при проведении документа РеализацияТоваровУслуг необходимо контролировать достаточность товара на складе.

12. Создайте отчет ОстаткиТоваровНаСкладах. Отчет отражает остатки товаров в количественном и суммовом (включая сумму НДС) выражении в разрезе складов. Пользователь должен иметь возможность указать период, на который формируется отчет.

13. Расположите в командном интерфейсе созданные отчеты в порядке их следования в содержании работы.

14. Ознакомьтесь с инструментами пользовательского режима, которые позволяют настраивать отчеты (сортировка, отборы, условное оформление, изменение типа диаграммы). В отчет занести примеры настройки форм отчетов из пользовательского режима.