

## 1. Обзор современного рынка ERP-систем

Впервые о ERP заговорили в 1990 г. Очень быстро крупные предприниматели оценили достоинства новой концепции, и уже в середине 90-х гг. был сформирован рынок по внедрению ERP-систем.

Всего за несколько лет новый метод планирования ресурсов, предназначенный изначально для промышленных предприятий, стал использоваться повсеместно. Активное развитие интернета коснулось и ERP-систем. В конце 90-х – начале 2000-х гг. почти все производители объединили ERP и Internet. На сегодняшний день ERP стали частью ИТ-сферы.

В настоящее время на рынке отечественного программного обеспечения принято выделять три группы информационных систем управления предприятием.

К первой группе обычно относят зарубежные разработки класса MRP II/ERP, представляющие собой крупные интегрированные пакеты с ориентацией на управление бизнес-процессами. Можно сказать, что эта группа содержит продукты высшей ценовой категории, такие как: SAP AG (R/3), BAAN (BAAN IV), ORACLE (OracleApplication), Navision–Microsoft и др. Специфика данной группы состоит в том, что в нее входят комплексные приложения в интегрированном формате, предназначенные для полной автоматизации деятельности различных предприятий, от средних фирм до крупных корпораций. Следует отметить тот факт, что программные продукты этой группы имеют исполнение «под ключ», однако при этом наряду с самостоятельным внедрением своих программных приложений допускается и использование «помощи» отечественных разработчиков. Общая стоимость всего комплекса является суммой трех составляющих:

- инсталляционной лицензии;
- консалтинговых услуг и обучения сотрудников;
- настройки, отладки и внедрения.

Следует отметить, что затраты последней составляющей иногда в разы могут превышать стоимость установки системы.

Вторую группу систем составляют небольшие российские пакеты прикладных программ (1С, Парус, Галактика, БОСС-Корпорация). Наши разработчики программного обеспечения выходят на рынок со своими корпоративными информационными системами, но с обязательным применением зарубежных платформ (например, система управления базой данных ORACLE). В таком случае в итоге получаются программные комплексы, превосходящие зарубежные аналоги, как минимум, в двух аспектах: цене и возможности учета уже в исходных версиях специфики отечественных фирм. Тем не менее, такие системы не в полном объеме поддерживают реализацию технологий MRPII/ERP.

В состав третьей группы информационных систем включаются локальные и малоинтегрированные пакеты российских производителей-разработчиков (БЭСТ, Инфин и т.д.).

Далее рассмотрим подробнее наиболее значимые информационные системы управления предприятием.

## **1.1 Программа SAP R/3**

SAP R/3 (разработка немецкой компании SAP AG) является самым широко используемым в мире стандартным решением класса ERP, которое служит для электронной обработки данных на основе клиент-серверной технологии. Система реализует возможность одновременного использования программы до 30 тысяч клиентов. Все составляющие системы R/3 устанавливаются с учетом на конкретного заказчика и позволяют реализовывать инсталляцию поэтапно. Системой предусматривается возможность выбора требуемой конфигурации, для этого программа располагает набором из почти 1000 уже готовых бизнес-решений и процессов.

Система состоит из ряда подсистем, созданных по принципу модульности:

- IS – отраслевые решения;
- MM – управление материальными потоками;
- QM – управление качеством;
- CO – контроллинг;

- РМ – техническое обслуживание и ремонт оборудования;
- PS – проекты;
- SD – сбыт;
- PP – планирование производства;
- WF – управление информационными потоками;
- АМ – основные средства;
- HR – управление персоналом;
- FI – финансы.

Обработка внутренних и внешних данных, а также функции поддержки принятия решений (ППР) на всех уровнях реализуются за счет наличия хранилищ бизнес-информации.

Система включает в себя следующие модули, представляющие собой основные элементы учета и отчетности:

- контроллинг(CO) – включает в себя: учет затрат по МВР (PCA), консолидацию (CS), планирование деятельности предприятия (BP), информационную систему для менеджмента (EIS);
- финансовую бухгалтерию(FI);
- управление инвестициями (IM);
- финансовый менеджмент(TR);
- систему управления материальными потоками(MM);
- систему сбыта(SD).

Следует отметить, что система предусматривает поддержку различных вариантов электронной коммерции, а также выполнение платежей при помощи кредитных карт.

Одно из основных достоинств системы заключается в устранении альтернативных каналов информации, что способствует получению адекватных и своевременных данных о состоянии процессов предприятия. К минусам же системы можно отнести относительные трудности, связанные с настройкой подсистем и жесткие требования к организации производства, а также невозможность осуществления «быстрого» реинжиниринга при появлении структурных изменений.

На текущий момент система внедрена более чем в 200 странах СНГ на различных предприятиях (н-р, Белгородэнерго, Восточно-Сибирская железная дорога, Белорусскийметаллургии-

ческий комбинат, Сургутнефтегаз, Красноярская железная дорога, Нижнетагильский металлургический комбинат и т.д.).

Стоимость внедрения программного продукта составляет примерно 6-7 тыс. долл. в расчете на 1 пользователя.

## **1.2 Программа BAAN IV**

BAAN IV (разработка одноименной компании из Нидерландов) является комплексной системой класса ERP, которая позволяет решать различные виды задач управления (BAAN-):

- моделирование предприятия;
- производство;
- финансы;
- процесс;
- транспорт;
- сбыт, снабжение, склады;
- проект;
- администратор деятельности предприятия;
- сервис.

Отличительной особенностью данной системы является ее «открытость», что позволяет пользователям модернизировать по своему усмотрению текущий функционал программного продукта при помощи создания собственных разработок, начиная с функциональных экранных форм и отчетов и заканчивая полным описанием всех бизнес-процессов. Для реализации указанных функций в системе присутствует элемент «Инструментарий», содержащий функционал для работы с различными элементами системы: меню, сеансами, экранными формами, таблицами, рапортами, библиотеками и программными скриптами.

На сегодняшний день система внедрена на следующих отечественных предприятиях: УралАЗ, «Нижфарм», КамАЗ, Челябинский ТЗ, БелАЗ, Шелеховский алюминиевый завод, Иркутское авиационное производственное предприятие и др.

### **1.3. Программа ORACLE E-BUSINESS SUITE**

ORACLE E-BUSINESS SUITE (разработка американской корпорация Oracle) является полноценным встроенным пакетом программных приложений для осуществления ведения бизнеса в электронной форме, функционирующим в корпоративных и глобальной сетях. Программный комплекс включает в себя множество приложений, требующихся любой организации или предприятию: производство, маркетинг, снабжение, продажи, бухгалтерия, обслуживание заказчиков, учет кадров и т.д.

Актуальную версию программных продуктов (ПП) Oracle E-Business Suite условно можно разделить на несколько функциональных блоков:

- Oracle ERP - модуль информационной подсистемы для решения задач планирования ресурсов предприятия, способствующих осуществлению продаж, производства, закупок и учета в процессе выполнения клиентских заказов;

- Oracle CRM - система по управлению взаимоотношениями с клиентами;

- Oracle E-Hub - система электронной коммерции.

Комплекс программных приложений Oracle для создания ERP-систем на предприятии, известный под торговой маркой Oracle Applications, имеет в своем составе набор различных приложений, предназначенных для автоматизации и оптимизации большинства хозяйственных процессов организации (производство, снабжение, финансы, управление персоналом и т.д.). Состав комплекса в данный момент насчитывает порядка 100 модулей, позволяющих организации успешно справляться с большинством задач, связанных с перемещениями финансовых и материальных потоков. К задачам такого типа относятся: управление запасами и персоналом, снабжение, расчет заработной платы, планирование производства, сотрудничество с поставщиками, финансовое планирование, управленческий учет и т.д.

Для достижения значимых успехов в ведении бизнеса компаниям следует как можно быстрее реализовывать информационный обмен данными со своими бизнес-партнерами. Для реализации такого взаимодействия в комплексе имеется специальная

высоконадежная система OracleExchange, позволяющая организациям реализовать ведение бизнеса с использованием глобальной сети Internet. Система в режиме реального времени обеспечивает эффективные механизмы взаимодействия со многими организациями, что, безусловно, способствует сервисным услугам, а также поставкам продукции и приобретению сырья высокого качества и в кратчайшие сроки.

Программа ORACLEE-BUSINESSSUITE сегодня внедрения на следующих предприятиях: Росатом, Телекомпания НТВ, ОКБ Сухого, Зеленодольский завод им. Горького и др.

## **1.4 Программы ГАЛАКТИКА и БОСС–КОРПОРАЦИЯ**

**ГАЛАКТИКА (Российская корпорация).** Система позволяет выполнять в автоматическом режиме решение возникающих на всех стадиях управленческого цикла задач планирования, прогнозирования, контроля и учета реализации продукции, коррекции планов и прогнозов, а также анализа результатов. Программа состоит из функциональных контуров, которые, в свою очередь, подразделяются на отдельные модули.

Состав функциональных контуров системы обеспечивает возможность любой организации выбора такого набора компонентов, который позволяет решать управленческие задачи, связанные с хозяйственной деятельностью, сразу в трех аспектах:

- по видам используемых ресурсов;
- по механизмам управленческой деятельности;
- по уровням управления.

Планируется, что в дальнейшем развитие системы будет связано с повышением ее технологичности, функциональности и степени интегрируемости для удовлетворения современных потребностей:

- ERP - планирование ресурсов предприятия;
- CSRP - синхронизированное с покупателем планирование ресурсов;
- SEM - стратегическое управление предприятием.

Внедрение системы: госкорпорация «Роскосмос» и корпорация «Галактика» заключили договор, регламентирующий их

сотрудничество в сфере цифровизации военно–промышленного комплекса (в частности, ракетно-космической промышленности). «Галактика» и Фонд «Сколково» заключили партнерский договор, по которому на площадях инновационного центра «Сколково» планируется создание научно-технический центра «Галактика НТЦ».

БОСС-КОРПОРАЦИЯ (российская компания «АйТи»). Система создана на базе Oracle 7 Server с целью реализации возможности автоматического управления всей деятельностью организаций различной направленности. Система состоит из главного модуля «Администратор» и ряда подсистем со следующими модулями управления:

- финансами;
- персоналом;
- производством;
- основными средствами и оборудованием;
- закупками, запасами и реализацией.

Разработчиком системы является фирма «АйТи», имеющая почти тридцатилетний опыт работы в области автоматизации управленческой деятельности. Так, уже в 1995г. компания выполнила проект по автоматизации деятельности Академии генштаба МО Российской Федерации.

Система использует в качестве аппаратной и программной платформы продукты SunMicroSystems (OC Solaris) и Oracle, что позволяет разработчиком использовать следующий инструментарий при создании проектов:

- Oracle 8 EnterpriseManager - управление и администрирование распределёнными средами данных;
- SQL\*Plus - средство выработки запросов, определения и управления данными;
- Designer - средство моделирования, генерации приложений и обратного реинжиниринга для приложений баз данных;
- ObjectDatabaseDesigner - объектное средство проектирования, создания и доступа;
- Developer - средство RAD-приложений баз данных в архитектуре «клиент-сервер» и Web.

## **1.5. Программы «1С: ERP Управление предприятием 2»**

«1С: ERP Управление предприятием 2» (компания 1С, Россия) – продукт международного класса для комплексной автоматизации средних и крупных предприятий самых различных сфер деятельности. Эта система представляется востребованной для предприятий с различными профилями, имеющими сложно-техническое, многопередельное производство, с числом рабочих мест более десяти тысяч человек.

Программа «1С:ERP» объединила в себе все лучшие практики. В ней гармонично сочетаются как проверенные временем решения (многолетний опыт разработчиков «1С»), так и инновационные тенденции (облачные, мобильные технологии).

Отличия ERP-решений «1С» от остальных систем:

- соответствие функционала отраслевой специфике, потребностям клиента;
- невысокая стоимость (лицензий, внедрения, сопровождения, владения);
- наличие персонала, имеющего компетенции;
- наличие положительных отзывов в отрасли;
- много специалистов, большой выбор партнеров, доступность сопровождения;
- компетентность исполнителя, положительный опыт работы с партнером;
- легкость внедрения и эксплуатации, гибкость;
- быстрая адаптация к изменениям законодательства;
- скорость внедрения;
- распространенность «1С:Предприятия 8», выгоды от использования совместно с другими решениями фирмы «1С» и легкость интеграции с ними;
- открытость и надежность системы;
- известность и надежность вендора;
- импортозамещение.

По результатам исследований, проведенных центром TAdviser, в нашей стране самыми популярными системами планирования ресурсов предприятия являются продукты на базе 1С. Сегмент отечественного программного обеспечения (ПО) составляет 31% по отношению к общему количеству внедрённых про-



ектов за последние 15 лет. На втором месте в рейтинге расположена система «Галактика ERP» (7,8%), третье место занимает SAP ERP (6,2%). Далее по списку следуют иностранные решения от MS (ORACLEE-BUSINESSSUITE) (6%) и BAAN(5,5%).

Из вышеизложенного можно сделать вывод, на данный момент у 1С: 8 нет достойных аналогов на российском рынке по соотношению цена/качество/масштабируемость/настраиваемость (особенно с выходом ERP 2.0). В отечественных альтернативах 1С, как правило, невозможно вносить изменения, а зарубежные достаточно дорогие.

### **Контрольные вопросы**

1. В настоящее время на российском рынке информационных систем управления предприятием можно выделить три группы. Какие?

2. Дайте краткую характеристику программы SAP R/3.

3. Дайте краткую характеристику программы BAAN.

4. Дайте краткую характеристику программы ORACLEE-BUSINESSSUITE.

5. Дайте краткую характеристику программы ГАЛАКТИКА.

6. Дайте краткую характеристику программы БОСС-КОРПОРАЦИЯ

7. Дайте краткую характеристику программы 1С: Предприятие.

8. В чем отличие программы «1С»: Предприятие от остальных аналогов?

## 2. Программа «1С: Предприятие 8»

### 2.1. Общие сведения

#### 2.1.1. Конфигурация и прикладное решение

«1С: Предприятие 8» представляет собой не просто набор программ, состоящих из постоянных файлов. Эта программа предоставляет пользователям и разработчикам инструментарий для создания собственных уникальных продуктов.

Фактически вся система может быть логически разделена на два блока, находящихся в непосредственной связи между собой: конфигурацию и среду для ее управления – платформу.

Платформой в «1С: Предприятие 8» называется среда разработки, имеющая предметно-ориентированную направленность, позволяющая разработчикам решать широкий спектр задач, касающихся в основном автоматизации деятельности предприятий. Как очевидно из самого названия, платформа является базой для развертывания и запуска различных конфигураций.

Конфигурация 1С (рис. 1) является прикладным программным решением, реализованным на базе платформы 1С, состоящим из описательной части базы данных (структура таблиц) и алгоритмической части (механизмы работы с данными).

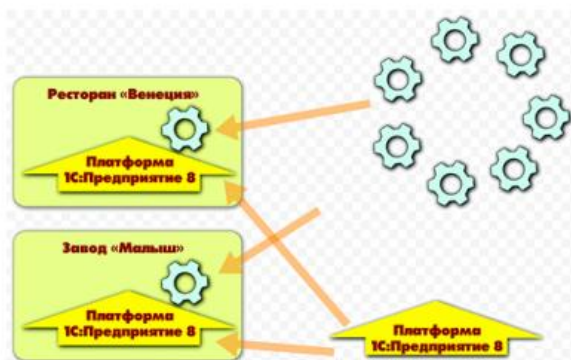


Рис. 1. Конфигурации и платформа в 1С

Таким образом, на базе единственной платформы «1С: Предприятие 8» существует множество различных конфигураций. Любое прикладное программное решение «работает» на одной платформе и с одной конфигурацией (рис. 1).

Отдельно каждый блок системы не может выполнять никаких задач: платформа отвечает только за корректность работы выбранной конфигурации, а конфигурация нуждается в наличии платформы для реализации любых проектов.

Ввиду того что спектр задач, решаемых разработчиками, может быть самым разнообразным, компания «1С» с партнерами старается представлять на рынке ПО продукты, позволяющие выполнять автоматизацию конкретных сфер деятельности человека. Так, примерами прикладных решений являются нижеперечисленные типовые пакеты:

- «1С: Предприятие 8. Управление торговлей»;
- «1С: Бухгалтерия»8;
- «1С: Зарплата и управление персоналом 8»;
- «1С: ERP Управление предприятием» и т.д.

Следует отметить тот факт, что типовые решения универсальны и по большому счету позволяют предприятиям одной отрасли использовать общее решение. Однако вместе с тем указанная универсальность приводит и к неизбежным трудностям, связанным с тем, что для какой-либо конкретной организации будут востребованы далеко не все опции прикладного решения, а каких-то специфических инструментов может и не хватать. Именно для решения этих затруднений и предусмотрено «свойство» конфигурируемости, которое позволяет подстроить типовое решение под требуемую область. Кроме того, в случае если применение типовой конфигурации не представляется возможным, система имеет функцию создания индивидуальной конфигурации на базе имеющейся платформы.

## 2.1.2. Режимы работы системы

Для того чтобы обеспечить такие возможности, система «1С: Предприятие 8» имеет различные режимы работы: «1С: Предприятие, Конфигуратор».

Режим «1С:Предприятие» является основным и служит для работы пользователей системы. В этом режиме пользователи вносят данные, обрабатывают их и получают итоговые результаты.

В режиме конфигуратора создаются необходимые для решения прикладной задачи объекты конфигурации. При этом определяются формы ввода информации, алгоритмы обработки данных, а также необходимые пользователю отчеты. В целом результатом конфигурирования является конкретная конфигурация, которая отражает модель предметной области определенного подразделения или целой организации.

Деятельность разработчика в конфигураторе может быть разделена на две составляющие. Это создание визуальных средств разработки: диалоговые окна, размещение необходимых элементов управления и задание их свойств. В этом случае практическая сторона сводится к управлению объектами с помощью мыши, а также к записи необходимой информации в поля ввода.

Вторую составляющую работы в конфигураторе представляет непосредственное программирование на встроенном языке.

При создании конфигурации, направленной на решение конкретной прикладной задачи, разработчик использует такие не заложенные в систему объекты конфигурации, как справочник, документ, форма, регистры, табличный документ и многие другие. Каждый из них характеризуется необходимым набором свойств и методов (стандартных операций по работе с объектами). На базе указанных объектов разработчик для определенной прикладной области уже создает специфические объекты, составляющие разрабатываемую конфигурацию.

В конфигураторе создаются не конкретные документы, справочники либо отчеты, а определяется лишь их вид или структура (шаблон).

### 2.1.3. Основная конфигурация и конфигурация базы данных

В каждой информационной базе существуют как минимум две конфигурации. Одна, с которой работает пользователь, а вторая конфигурация предназначена для разработчика или человека, который должен вносить изменения в конфигурацию (например, администратора базы данных). Для пользователя она «не видна».

Конфигурация, предназначенная для разработчика, называется *основная конфигурация*.

Конфигурация, с которой работают пользователи, называется *конфигурация базы данных*.

Основную конфигурацию можно редактировать. Конфигурацию базы данных редактировать нельзя, можно только произвести обновление конфигурации базы данных на основе основной конфигурации. В данном случае название «основная» дано с точки зрения разработчика. В общем случае информационная база (ИБ) «1С:Предприятия» может хранить более двух конфигураций: основную конфигурацию, конфигурацию базы данных и несколько конфигураций поставщиков. Кроме этого, вне ИБ может существовать хранилище, в котором находится конфигурация, предназначенная для групповой разработки. Вне ИБ может существовать также некоторое количество файлов конфигураций, в том числе файлы новой поставки (рис. 2).



Рис. 2. Основная конфигурация и конфигурация БД

Конфигурация поставщика, находящаяся в информационной базе, содержит предыдущее состояние поставки.

Возможна ситуация, когда конфигурация находится на поддержке одновременно у нескольких поставщиков, каждый из которых поддерживает только свою часть в виде отдельной конфигурации. В этом случае информационная база будет хранить несколько конфигураций поставщиков (состояние предыдущей поставки для каждого поставщика отдельно). Файлы новой поставки могут существовать в виде файлов конфигураций (полная поставка) и файлов обновлений (поставка обновлений). Хранилище конфигурации содержит конфигурацию, предназначенную для групповой разработки.



Рис. 3. Взаимодействие двух конфигураций

Она хранится не в виде единой конфигурации, а в виде отдельных объектов в разрезе версий конфигурации (рис. 3).

#### 2.1.4. «Толстый» и «тонкий» клиент

Самый простой и известный всем клиент 1С – это толстый клиент («обычный»). До версии 1С 8.2 кроме него никаких других вариантов предоставлено и не было. Конфигуратор 1С (в настоящее время) работает только в толстом клиенте. С файловой базой данных рекомендуется работать также с помощью толстого клиента 1С.

Толстый клиент работает под Windows. Толстым его называют по причине требовательности к ресурсам компьютера пользователя, т.к. он может запрашивать достаточно большие объемы данных по сети.

С точки зрения программиста основное отличие толстого клиента 1С в том, что большинство программ на встроенном

языке 1С он выполняет на компьютере пользователя. Запрос из базы данных будет выглядеть следующим образом:

- клиент 1С запрашивает данные у сервера 1С;
- сервер 1С запрашивает данные из базы данных;
- данные пересылаются на клиент 1С;
- клиент 1С обрабатывает данные.

Тонкий клиент 1С появился сравнительно недавно, устанавливается он по умолчанию в комплекте с другими вариантами клиентов 1С. Конфигуратор 1С не работает в тонком клиенте 1С. Он может работать с файловым вариантом базы данных, но лучше использовать клиент-серверный режим.

Тонкий клиент 1С также работает под Windows и называют его так по причине правильной организации клиент-серверной организации программы. В отличие от толстого клиента 1С запрос из базы данных будет выглядеть следующим образом:

- клиент 1С передает на сервер необходимость пользователя запросить данные у сервера 1С;
- сервер 1С запрашивает данные из базы данных;
- сервер 1С обрабатывает данные;
- результат обработки данных пересылается на клиент 1С.

Плюс – никакой требовательности к ресурсам компьютера пользователя, предполагается меньший трафик. Минус – требовательность к ресурсам сервера выше. Последний минус для больших компаний снимается тем, что сервер 1С можно масштабировать, т.е. установить систему из нескольких серверов 1С на разных компьютерах, и они будут работать в связке.

#### 2.1.5. Типы данных

Типы данных в 1С можно разбить на группы.

##### *Простые (базовые) типы данных*

Простыми они называются потому, что кроме своего значения не имеют больше ничего. К простым типам в 1С относят следующие:

- Булево- принимает всего два значения – Истина и Ложь. Применяется в основном для формирования результата логического выражения.

- Число - положительные и отрицательные числовые значения. Как целые, так и десятичные. Например, значения 1, 3.6, -2, 0, -5,54634424 будут являться значениями числового типа. Для переменной числового типа значением по-умолчанию будет являться 0 (ноль). Разделителем целой и дробной части числа в программном модуле 1С является точка «.». Над переменными этого типа можно проводить арифметические операции.

- Дата - тип данных «дата» в 1С предназначен для работы с датами (дата документа, дата события и пр.) Причем в системе «1С: Предприятие 8» тип «дата» содержит не только значение самой даты, но и значение времени в пределах этой даты (часы, минуты и секунды).

Значением «поумолчанию» для переменной типа «дата» (или, как говорят, пустой датой) является значение 01.01.0001 00:00:00 (01 января 0001 года 00 ч 00 мин 00 с.)

- Строка - тип данных, позволяющий передавать строки, т.е. любой набор символов. Строки в 1С заключаются в двойные кавычки («Это строка...»). Над строками в 1С можно производить операции сложения (по-умному это называется конкатенацией) Под сложением строк понимается получение некой результирующей строки путем «склеивания» составляющих ее подстрок. *Например:* ПолноеНаименование = «Вентилятор»+«бытовой».

В итоге ПолноеНаименование будет иметь значение «Вентилятор бытовой».

#### *Агрегатные типы данных*

Платформа оперирует объектами – это набор данных, которые не просто имеют какое-то значение как базовые типы данных, но и обладают целым набором свойств и методов, позволяющих им выполнять свои функции.

Наиболее распространенные типы, применяемые в любой конфигурации, документы и справочники. Рассмотрим агрегатный тип данных на примере «Справочника».

Набор свойств «Справочника» определяется частично платформой 1С и частично разработчиком конфигурации. Такие свойства как «наименование», «код», «родитель», «владелец» и др. существуют у справочников изначально (т.е. определены платформой) и удалить их из состава свойств невозможно.



Другие свойства, которые нужны для конкретной конфигурации (например, «ИНН» контрагента), создаются разработчиком конфигурации и могут удаляться из состава свойств совершенно свободно. Свойства также называются реквизитами объекта. В данном случае «Справочника». В свою очередь, реквизиты могут быть простого или агрегатного типа.

#### *Неопределенно и Null*

«Неопределенно» - обозначает, что существует переменная, которая может принимать значения разных типов, но в данный момент значение не задано. Значение типа «Неопределенно» имеют, как правило, реквизиты составного типа. То есть реквизиту, например, «Справочника», могут присваиваться значения разных типов (например, строка или ссылка на другой справочник). Поэтому незаполненный реквизит будет иметь тип значения «Неопределенно», т.е. в буквальном смысле тип есть, но непонятно какой конкретно, то ли ссылка, то ли строка. Конкретный тип реквизит получит после присвоения ему какого-то значения.

«Null» - обозначает отсутствие значения, т.е. в буквальном смысле пустое место. В этом и состоит его отличие от типа «Неопределенно». Используются исключительно для определения отсутствующего значения при получении данных с помощью запросов, например, при соединении таблиц.

### **Контрольные вопросы**

1. Из каких основных частей состоит программа 1С?
2. Что такое платформа и что такое конфигурация в 1С?
3. Для чего используются разные режимы запуска системы «1С: Предприятие»?
4. Какие режимы работы программы «1С: Предприятие» вы знаете?
5. Что такое конфигурируемость?
6. «Толстый» и «тонкий» клиент, в чем различие?
7. Какие типы данных вы знаете?

## 2.2 Объекты конфигурации

### 2.2.1. Подсистемы

Подсистемы – это основные элементы для построения интерфейса «1С: Предприятия». Основу разработки конфигурации составляет проектирование состава подсистем. При этом необходимо тщательно продумать состав подсистем и затем аккуратно и осмысленно привязать к подсистемам те объекты конфигурации (ОК), которые будут созданы.

Объекты конфигурации Подсистема позволяют выделить в конфигурации функциональные части, на которые логически разбивается создаваемое прикладное решение. Эти объекты располагаются в ветке объектов Общие и позволяют строить древовидную структуру, состоящую из подсистем и подчиненных подсистем (рис. 4).

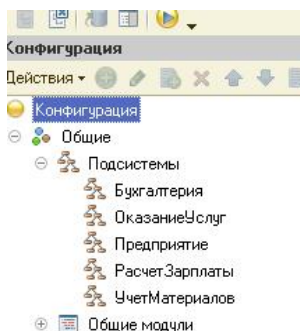


Рис. 4. Структура подсистем

Таким образом, наличие подсистем определяет структуру прикладного решения, организует весь пользовательский интерфейс, позволяет рассортировать различные документы, справочники и отчеты по логически связанным с ними разделам, в которых пользователю будет проще их найти и удобнее с ними работать. При этом каждому конкретному пользователю будут видны

лишь те разделы, то есть та функциональность прикладного решения, которые ему нужны в процессе работы.

Так, можно выделить в отдельную подсистему все, что имеет отношение к бухгалтерскому учету. Кроме этого, отдельной предметной областью является расчет зарплаты сотрудников предприятия. Всю производственную деятельность нашей фирмы можно разделить на учет материалов и оказание услуг. А кроме этого, для выполнения специальных административных функций с базой данных нам нужно иметь отдельную подсистему, в которой будет иметь доступ только администратор.

### 2.2.2. Константы, перечисления

Для работы с постоянной и условно постоянной информацией в системе используются объекты типа «Константа».

Информация, хранящаяся в константах, редко изменяется, но, как правило, часто используется в работе. Например, в константах может храниться наименование предприятия, его ИНН, фамилии директора и главного бухгалтера и другая подобная информация. В системе может быть описано неограниченное количество констант.

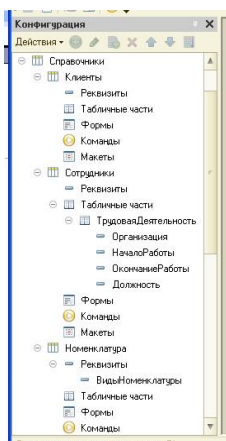
Обычно вся работа с константами ведется в специальной форме. Если пользователь имеет доступ к константам, то форма для ввода и редактирования их значений может быть вызвана через системное меню «Операции-Константы».

В пользовательском режиме работы с программой можно только корректировать значения констант в форме. Добавление новых констант, удаление ненужных констант, изменение наименования у существующих констант может выполняться только в конфигураторе.

Объект конфигурации «Перечисление» предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации.

На основе объекта конфигурации «Перечисление» платформа создает в базе данных таблицу, в которой может храниться набор некоторых постоянных значений. В реальной жизни этому

объекту может соответствовать, например, перечисление вариантов указания цены («включая НДС», «без НДС»).



Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять их, удалять или добавлять новые. Из этого следует важная особенность перечисления: значения перечисления не «обезличены» для конфигурации, на них могут опираться алгоритмы работы программы.

Рис. 5. Дерево конфигурации

### 2.2.3. Справочники

Для работы с постоянной и условно постоянной информацией с некоторым множеством значений в системе используются объекты типа «Справочник». В конкретной конфигурации создается необходимое количество справочников, для хранения данных об объектах, используемых при автоматизации данной предметной области. Например, это могут быть справочники «Клиенты», «Номенклатура», «Сотрудники» и пр. (рис. 5).

Кроме того, справочники используются для организации аналитического учета, что позволяет исключить неоднозначность при заполнении реквизитов документов или других справочников. При вводе информации в экранные формы документа минимум этой информации вводится с клавиатуры, а большая часть – выбором из открывающихся справочников.

Объект конфигурации «Справочник» предназначен для работы со списками данных. Как правило, в работе любой фирмы используются списки сотрудников, списки товаров, списки клиентов, поставщиков и т. д. Свойства и структура этих списков описываются в объектах конфигурации «Справочник», на основе кото-

рых платформа создает в базе данных таблицы для хранения информации из этих справочников.

Справочник состоит из элементов. В базе данных каждый элемент «Справочника» представляет собой отдельную запись в основной таблице, хранящей информацию из этого «Справочника». Каждый элемент «Справочника», как правило, содержит некоторую дополнительную информацию, которая подробнее описывает этот элемент. Набор такой информации является одинаковым для всех элементов «Справочника», и для описания такого набора используются реквизиты объекта конфигурации «Справочник», которые также, в свою очередь, являются объектами конфигурации.

Так как эти объекты конфигурации логически связаны с объектом «Справочник», они называются подчиненными этому объекту.

У каждого объекта конфигурации «Справочник» по умолчанию существует набор стандартных реквизитов: Код и Наименование. Причем если установить длину стандартного реквизита Код равной нулю, то у «Справочника» будет недоступен этот реквизит. То же самое относится к реквизиту Наименование. Однако как минимум либо Код, либо Наименование должны присутствовать в реквизитах «Справочника», иначе такой справочник не имеет смысла.

Таким образом, в базе данных «Справочник» хранится в виде таблицы, в строках которой расположены элементы списка, а каждому реквизиту (стандартному или созданному разработчиком) в этой таблице соответствует отдельный столбец. Соответственно в ячейках этой таблицы хранится значение конкретного реквизита для конкретного элемента «Справочника».

Причем доступность стандартных реквизитов зависит от свойств «Справочника». Например, если «Справочник» иерархический, у него будет доступен стандартный реквизит Родитель. Если «Справочник» подчинен другому объекту конфигурации, у него будет доступен реквизит Владелец.

Кроме этого, каждый элемент «Справочника» может содержать некоторый набор информации, которая одинакова по своей структуре, но различна по количеству, и предназначена

для разных элементов «Справочника». Так, например, каждый элемент «Справочника» Сотрудник может содержать информацию о составе семьи сотрудника. Для одного сотрудника это будет только супруга, а у другого семья может состоять из супруги, сына и дочери. Для описания подобной информации могут быть использованы табличные части объекта конфигурации «Справочник», являющиеся подчиненными ему объектами конфигурации. В этом случае в базе данных будут созданы дополнительные таблицы для хранения табличных частей, подчиненных конкретному элементу «Справочника».

В зависимости от того, какие действия мы хотим выполнять со справочником, нам требуется изображать «Справочник» в «разном виде». Например, для того чтобы выбрать некоторый элемент «Справочника», удобнее представить «Справочник» в виде списка, а для того чтобы изменить какой-то элемент «Справочника», удобнее представить все реквизиты этого элемента «Справочника» в одной форме.

Система может самостоятельно сгенерировать все формы, которые нужны для представления данных, содержащихся в «Справочнике». Причем система знает, какие именно формы нужно использовать в каких ситуациях.

Форма служит для визуализации данных, находящихся в базеданных. Она представляет эти данные в удобном для пользователя виде и позволяет описать алгоритмы, которые будут сопровождать работу пользователя с данными, показанными в форме.

Для отображения «Справочника» в различных ситуациях требуется максимум пять форм:

- объект - форма элемента «Справочника» используется для редактирования или создания элемента «Справочника»;
- групп - форма группы «Справочника» используется для редактирования или создания группы «Справочника».
- список - форма списка «Справочника» используется для отображения списка элементов «Справочника»;
- форма для выбора - форма выбора «Справочника» используется для того, чтобы в поле некоторой формы выбрать один из элементов «Справочника»;

- форма для выбора группы - форма выбора группы «Справочника» используется, когда в поле некоторой формы нужно выбрать не просто элемент «Справочника», а одну из его групп.

Любая форма может быть описана в конфигураторе. Для создания такого описания существует подчиненный объект конфигурации Форма, который подчинен одному из прикладных объектов (Справочник, Документ и пр.), но может существовать и самостоятельно. На основании описания, содержащегося в объекте конфигурации Форма, в нужный момент работы платформа «1С: Предприятие» создаст программный объект Форма, с которым и будет работать пользователь.

### **Контрольные вопросы**

1. Для чего используется ОК «Подсистема»?
2. Как управлять порядком вывода и отображением подсистем в конфигурации?
3. Для чего используется ОК «Константы»?
4. Для чего используется ОК «Перечисление»?
5. Для чего предназначен ОК «Справочник»?
6. Для чего используются реквизиты и табличные части «Справочника»?
7. Какие основные формы существуют у «Справочника»?
8. Что такое предопределенные элементы и как пользователь может отличить обычные элементы «Справочника» от предопределенных элементов?
9. Как создать ОК «Справочник» и описать его структуру?

## **2.3. Документы в программе «1С: Предприятие»**

### **2.3.1. Основные сведения**

Бухгалтерский учет является строго документированным. Все хозяйственные операции обязательно оформляются первичными документами, которые являются носителями информации о данной операции.

Первичный бухгалтерский документ является письменным подтверждением совершенной или подлежащей обязательному

совершению хозяйственной операций, заверенной подписями ответственных лиц. Вследствие этого он имеет правовое (юридическое) значение. Каждый первичный документ содержит показатели (реквизиты), необходимые для характеристики хозяйственной операции.

В программе «1С: Бухгалтерия» документы – это электронные аналоги обычных бумажных документов. Документы предназначены для регистрации фактов реального мира в компьютере. С помощью документов можно вводить информацию в базу данных, например, принимать нового сотрудника документом «Приказ о приеме на работу». Или можно просто создать платежное поручение в банк или ввести счет-фактуру. Свойства и структура этих «Документов» описываются в объектах конфигурации «Документ», на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов.

Логика работы документов отличается от логики работы других объектов конфигурации. Документ обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета. До тех пор пока документ не проведен, состояние учета неизменно, и документ – не более чем черновик. Как только документ будет проведен, изменения, вносимые документом данные в учет, вступят в силу, и состояние учета будет изменено.

Документы обычно формируют движения. Движения документа – это изменения в «Справочниках» и других объектах конфигурации, порожденные этим документом.

С точки зрения программиста после проведения в системе образуются проводки и движения – записи регистров в БД.

С точки зрения пользователя «провести» документ означает «пустить его в дело». С точки зрения программиста при проведении документа сработает модуль проведения документа (процедура Обработка Проведения). Именно там определяются все действия, выполняемые при проведении документа. Например, при проведении документа «Приказ о приеме на работу» сработает модуль этого документа, который в справочник Сотрудники добавит нового сотрудника.

У документа есть экранная форма – Диалог. Именно его видит пользователь на экране. Также у документа обычно есть одна



или несколько бумажных форм – Таблиц. С помощью Таблиц документ печатается на бумаге. Поведение документа определяется с помощью встроенного языка 1С. На языке записывается, что документ делает в системе, как он формирует бумажный бланк и как он себя ведет на экране.

Поскольку документ вносит изменения в состояние учета, он всегда «привязан» к конкретному моменту времени. Это позволяет отражать в базе данных фактическую последовательность событий.

Поэтому у каждого объекта конфигурации Документ по умолчанию существуют стандартные реквизиты – это Дата и Номер. Поскольку тип данных Дата содержит дату и время с точностью до секунды, этот реквизит и определяет в основном положение документа на оси времени.

Следующим важным фактом является то, что система «1С: Предприятие» имеет механизмы, позволяющие отслеживать правильность состояния учета.

Предположим, что мы изменили один из проведенных ранее документов и снова провели его задним числом. В этом случае система «1С: Предприятие» способна отследить, повлияют ли внесенные нами изменения на последующие проведенные документы, и, если это так, система способна перепровести необходимые документы. В связи с этим в программе 1С существуют следующие механизмы проведения документов.

#### 1. Оперативное проведение документа 1С:

- оперативно проводить можно только документы за сегодняшний день;

- оперативное проведение документа 1С выполняется в текущий момент времени. Даже если документ был создан сегодня, несколько часов назад, при оперативном проведении этого документа время этого документа поменяется на текущее;

- при оперативном проведении в большинстве конфигураций выполняется проверка на наличие остатков. Если, к примеру, на складе нет остатка, то товарный документ не пройдет;

- невозможно оперативно провести документ будущей датой.

#### 2. Неоперативное проведение документа 1С:

- неоперативно можно провести документ за любую дату, при этом дата и время документа не изменятся;

- при неоперативном проведении остатки могут не контролироваться. Например, документ проводится, но при этом выдает сообщение об отсутствии товара на складе.

Документы в «1С: Бухгалтерии» позволяют:

- после их заполнения автоматически формировать необходимые бухгалтерские проводки;
- готовить к печати форму первичного бухгалтерского документа (печатную форму).

### 2.3.2. Формы документа

Для визуализации документа существует несколько основных форм:

- форма документа;
- форма списка документа;
- форма выбора документа.

У всех прикладных объектов конфигурации существует некоторое количество основных форм. Они служат для отображения данных объекта в том или ином виде.

Если разработчик не назначит в качестве основных форм объекта свои собственные, система будет генерировать необходимые формы объекта самостоятельно в те моменты, когда к ним происходит обращение. Наличие такого механизма позволяет разработчику не тратить время на создание форм для тестирования своей разработки, а воспользоваться тем, что платформа создаст по умолчанию.

Создание этих форм происходит динамически, в процессе работы системы. Форма создается в тот момент, когда к ней происходит обращение. Причем неважно, интерактивное это обращение или программное.

Также примечательным фактом является то, что состав основных форм, определенных для объекта конфигурации, может не совпадать с перечнем тех форм, которые вообще можно создать для данного объекта, используя конструктор форм.

Например, для большинства регистров в конфигураторе можно задать основную форму списка. Однако если открыть конструктор форм для регистра, вы увидите, что кроме формы списка предлагается создать и форму набора записей регистра, которая

отсутствует в перечне основных форм. Дело в том, что состав основных форм определяется исходя из того, какое представление данных может понадобиться в процессе интерактивной работы пользователя. Для этих представлений разработчик может создать свои формы и указать их в качестве основных, а может использовать те формы, которые система создаст автоматически.

Конструктор форм, напротив, исходит из потребностей разработчика. Если разработчик посчитает нужным использовать для какого-либо регистра вместо обычной формы списка форму набора записей, он сможет это сделать, воспользовавшись Конструктором и определив ее в качестве основной формы регистра. Но для логики работы системы это не будет иметь принципиального значения.

У всех прикладных объектов конфигурации существует некоторое количество основных форм. Они служат для отображения данных объекта в том или ином виде. Если разработчик не назначит в качестве основных форм объекта свои собственные, система будет генерировать необходимые формы объекта самостоятельно в те моменты, когда к ним происходит обращение.

Наличие такого механизма позволяет разработчику не тратить время на создание форм для тестирования своей разработки, а воспользоваться тем, что платформа создаст по умолчанию. Создание этих форм происходит динамически, в процессе работы системы. Форма создается в тот момент, когда к ней происходит обращение.

Так, форма списка для «Справочника» Клиенты будет создана как при интерактивном выборе в меню Все функции - Справочники - Клиенты, так и при программном вызове глобального метода Получить Форму().

Состав основных форм определяется исходя из того, какое представление данных может понадобиться в процессе интерактивной работы пользователя. Для этих представлений разработчик может создать свои формы и указать их в качестве основных, а может использовать те формы, которые система создаст автоматически.

### 2.3.3. События и Обработчики событий

Любое прикладное решение «1С: Предприятия» имеет совершенно определённый жизненный цикл. Прикладное решение запускается. Открывается основной раздел. После этого прикладное решение ждёт действий пользователя. Пользователь может открыть какую-то форму. Может изменить в ней какие-то данные. Может закрыть форму. Тогда он закрывает программу. Работа прикладного решения завершается. Этот жизненный цикл фиксирован. Он определён платформой, и изменить его нельзя.

Нов в нём есть отдельные моменты, в которые вы можете вмешаться. Такие моменты, когда вы можете вмешаться в «жизнь» прикладного решения, называются события. Набор этих событий фиксирован, постоянен. Его определяет платформа. Придумать какое-то своё событие вы не можете. Но вы можете использовать любое событие из тех, которые имеются.

Например, *ПередНачаломРаботыСистемы*, *ПриНачалеРаботыСистемы* и т.д.

«1С: Предприятие» содержит большое количество событий.

При работе с событиями в платформе «1С: Предприятие 8» следует различать два типа событий: события, связанные с формой и ее элементами, и все остальные. Разница заключается в том, что обработчики событий, связанных с формой и ее элементами, – назначаемые, а обработчики всех остальных событий – фиксированные.

Фиксированный обработчик события должен иметь имя, совпадающее с именем события. Только в этом случае он будет вызываться при возникновении соответствующего события.

Назначаемый обработчик может иметь произвольное имя. Если имя процедуры совпадает с именем события формы или ее элемента, этого совсем недостаточно для вызова процедуры обработки события с таким именем. Требуется явное назначение процедуры обработчиком этого события в палитре свойств, в соответствующем событии.

Таким образом, любая процедура, расположенная в модуле формы, может быть назначена обработчиком любого события (или сразу нескольких событий) формы или ее элемента, расположенного в форме. Имя процедуры в этом случае не имеет значения. Важно лишь то, что она назначена обрабатывать какое-либо событие.

Назначение обработчика может выполняться интерактивно, при работе с формой в конфигураторе, или программно, используя методы формы и ее элементов – УстановитьДействие().

Как самому разобраться с множеством незнакомых свойств и методов объектов конфигурации, чтобы в будущем самостоятельно изучать фрагменты кода или создавать свои собственные процедуры на встроенном языке?

Синтакс-помощник – инструмент, созданный для помощи разработчику, содержащий описание всех программных объектов, которые использует система, их методов, свойств, событий и пр. Чтобы открыть синтакс-помощник, нужно нажать соответствующую кнопку на панели инструментов конфигуратора или выполнить команду главного меню Справка - Синтакс-помощник. Он представляет собой древовидную структуру, состоящую из глав, разделов, подразделов, описывающих встроенный язык 1С. Кроме Синтакс-помощника, анализ кода можно проводить и с помощью отладчика.

Пользоваться отладчиком наиболее удобно в тех случаях, когда нужно написать какой-то собственный код. Потому что в отличие от синтакс-помощника, где нужно, хорошо представлять контексты исполнения, структуру объектов и пр.(разобраться в чужом коде), с помощью отладчика ничего этого представлять не нужно. Можно просто остановиться в конкретном месте программы и посмотреть, какие же свойства здесь доступны или какие программные объекты здесь используются.

Отладчик – вспомогательный инструмент, облегчающий разработку и отладку программных модулей системы «1С: Предприятие». Отладчик предоставляет следующие возможности:

- пошаговое выполнение модуля,
- расстановка точек останова,
- прерывание и продолжение выполнения модуля,

- возможность отладки нескольких модулей одновременно,
- вычисление выражений для анализа состояния переменных,
- просмотр стека вызовов процедур и функций,
- возможность остановки по возникновению ошибки,
- возможность редактирования модуля в процессе отладки.

#### 2.3.4. Модули

Ранее был рассмотрен код обработчиков событий. Эти процедуры располагаются в модуле формы, т.е. некотором хранилище текста программы на встроенном языке. В конфигурации существуют различные виды модулей. Они могут принадлежать некоторым объектам конфигурации (например, формам), а могут существовать сами по себе (принадлежать всей конфигурации в целом). Текст программы, содержащийся в модулях, будет использоваться платформой в заранее известные моменты работы системы «1С: Предприятие» – события.

В «1С: Предприятии» существуют следующие виды модулей.

1. Модуль управляемого приложения выполняется при старте системы «1С: Предприятие» в режимах тонкого клиента и веб-клиента. В нем возможно объявление переменных, а также объявление и описание процедур и функций, которые будут доступны в любом модуле конфигурации (кроме модуля внешнего соединения). Их доступность также обеспечивается для неглобальных общих модулей с установленным свойством Клиент (управляемое приложение). В контексте модуля управляемого приложения доступны экспортируемые процедуры и функции общих модулей.

2. Общие модули. В них хранятся процедуры и функции, которые вызываются из других модулей системы. Сам по себе общий модуль не исполняется. Исполняются отдельные его процедуры/функции в момент их вызова из других модулей.

3. Модули объектов – это, например, модуль элемента «Справочника» или модуль документа. Эти модули вызываются тогда, когда либо программно создается этот объект средствами встроенного языка, например, методами СоздатьЭлемент() у менеджеров справочников или СоздатьДокумент() менеджеров до-

кументов, либо когда пользователь создает новый элемент «Справочника» или документ интерактивно. При записи измененных данных объекта в базу данных вызываются различные обработчики событий, которые располагаются в модуле объекта.

4. Модули форм. Каждая форма, определенная в конфигурации, имеет свой собственный модуль. Этот модуль исполняется при создании объекта УправляемаяФорма встроенного языка. Этот объект создается системой в режиме «1С: Предприятие» в тот момент, когда мы программно (методами ПолучитьФорму() или ОткрытьФорму()) или интерактивно открываем форму некоторого элемента.

5. Модуль сеанса – модуль, который автоматически выполняется при старте системы «1С: Предприятие 8» в момент загрузки конфигурации. Модуль сеанса предназначен для инициализации параметров сеанса и отработки действий, связанных с сеансом работы. Модуль сеанса не содержит экспортируемых процедур и функций и может использовать процедуры из общих модулей конфигурации.

6. Модуль внешнего соединения предназначен для размещения в нем текстов функций и процедур, которые могут вызываться в сессии внешнего соединения.

7. Модуль менеджеров. Для каждого прикладного объекта существует менеджер, предназначенный для управления этим объектом как объектом конфигурации. С помощью менеджера можно создавать объекты, работать с формами и макетами. Модуль менеджера позволяет расширить функциональность менеджеров, предоставляемых системой, за счет написания процедур и функций на встроенном языке. Фактически это позволяет описать собственные методы для объекта конфигурации (например, «Справочника»), которые относятся не к конкретному экземпляру объекта базы данных, а к самому объекту конфигурации.

8. Модуль команды. Как в самой конфигурации, так и у многих прикладных объектов могут существовать подчиненные объекты конфигурации – Команды. У каждой команды существует модуль команды, в котором можно написать предопределенную процедуру ОбработкаКоманды() для выполнения этой команды.

## Контрольные вопросы

1. Для чего предназначен объект конфигурации «Документ»?
2. Какими характерными особенностями обладает «Документ»?
3. Для чего предназначены реквизиты и табличные части документа ?
4. Какие существуют основные формы документа?
5. Что такое проведение документа?
6. В чем отличие оперативного проведения от неоперативного?
7. Как создать новый документ и заполнить его данными?
8. Какие виды форм документа вы знаете?
9. Что такое события и с чем они связаны?
10. «Обработчик события» для чего предназначен?
11. Для каких целей предназначены инструменты Синтакс-помощник и Отладчик?
11. Что такое модуль и для чего он нужен?
12. Какие виды модулей вы знаете?

## 2.4 Регистры в программе «1С: Предприятие»

### 2.4.1. Общие сведения

В любой конфигурации 1С.8 можно увидеть такой вид объектов, как регистры. Основное их предназначение — оптимизация получения данных для отчетов. Существует четыре вида регистров: сведений, накоплений, бухгалтерии и расчета. И хотя предназначены эти виды для решения разных задач, уже по тому, что они все называются «регистрами», можно догадаться, что они имеют и нечто общее.

1. Как объекты конфигурации они нужны для более быстрого считывания информации из базы данных, например, в запросах. Регистры можно сравнить с каталогом книжной библиотеки (раньше их составляли на бумажных карточках). То есть это не только хранение информации (данных), но и ее систематизация (создание определенной структуры), когда в конкретный регистр попадают данные (например, из документов разного вида)



и при необходимости ее можно достаточно быстро оттуда извлечь и вывести, например, в отчет или обработать иным образом. В общем случае основное использование регистров в 1С можно изобразить следующей схемой (рис. 6).

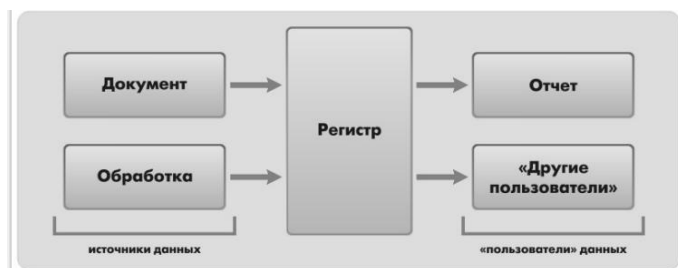


Рис. 6. Схема регистров

2. Основная проблема при проектировании регистров - это определение его структуры. Структура регистра должна быть такова, чтобы извлекать из него нужную информацию без утомительной обработки. То есть определяется, *что*(ресурс) и *в каких разрезах*(измерения) нужно учесть.

Ресурс - это количественные или суммовые данные, которые хранятся в регистре.

Разрез (измерения) - это то, в каких разрезах требуется хранение информации.

*Пример.* Регистр «Остатки товаров» должен содержать сведения о количестве и стоимости каждого товара на каждом складе. В идеологии системы 1С регистр такого вида представляет собой прямоугольную систему координат, на одной оси которой находятся склады, на другой - товары, а на пересечении конкретного склада и конкретного товара находятся цифры количества товара и стоимости товара (рис. 7).

Товар	Кол-во (шт)		Стоимость (руб)		
	Склад1	Склад2	Склад3	Склад	
Товар4	10	150			
Товар3					
Товар2					
Товар1					

Рис. 7. Пример представления регистра «Остатки товаров» в идеологии системы 1С

Регистр: Остатки товаров.

Измерения: Товар, Склад.

Ресурсы: Кол-во, Стоимость.

С помощью методов встроенного языка мы можем легко получить ответы на вопросы:

- остаток конкретного товара на конкретном складе;
- остаток конкретного товара на всех складах;
- стоимость всех товаров на конкретном складе.

Движения в регистрах

В табличном виде регистр ОстаткиТоваров представляется следующим образом:

	Товар	Склад	Кол-во	Стоимость
приход	Товар4	Склад1	15	200
расход	Товар4	Склад1	10	100
приход	Товар4	Склад1	5	50
остаток	Товар4	Склад1	10	150

Одна строка из этой таблицы называется "движение". Движения в регистрах создаются только при проведении документов. В регистре, кроме измерений ресурсов, можно задать *реквизиты*.

Реквизиты - это дополнительные сведения, сопровождающие движение. Методами встроенного языка можно отбирать движения с заданным значением реквизита.

3. Регистры имеют табличную структуру, но она отличается от структуры объектных таблиц. Так что вы не найдете таких классов, как РегистрСсылка или РегистрОбъект. Состав таблицы регистра зависит от его свойств.

4. Данные в регистры записываются в виде наборов записей. Каждый набор состоит из одной или нескольких записей. При этом на запись в наборе нельзя сослаться или обратиться к ней. А также ни набор записей, ни запись в наборе не могут иметь состояния «пометка на удаление».

5. При обращении в запросах к регистрам для получения данных существует возможность обратиться не только к физическим таблицам регистра, но и к виртуальным таблицам, которые представляют собой вложенный запрос, получающий данные по определенным параметрам. Параметры виртуальной таблицы задаются в зависимости от конкретных потребностей по получению данных из таблиц регистров.

#### 2.4.2. Регистры сведений

Регистры сведений в 1С- это структурированный набор данных с измерениями и ресурсами, предназначенный для хранения периодической информации.

##### *Периодичность*

Информация хранится в разрезе измерений и периода. Регистру сведений, можно задать периодичность:

- непериодический,
- по регистратору,
- секунда,
- день,
- неделя,
- месяц,
- квартал,
- год.

Периодичность нужна для выбора информации из регистра на определенный период времени. Если указать периодичность, записи в регистр будут производиться с периодом, когда была сделана запись. Допустим если посмотреть регистр «Цены номенклатуры», можно увидеть историю изменения цен, с какими измерениями и в какой период времени была сделана запись.

Периодичность в регистрах сведений нужны для информации, которая изменяется в течение времени, например: курсы валют, цены номенклатуры, скидки и наценки номенклатуры и т.д.

### *Регистраторы*

Если делать запись в регистр сведений с помощью документа, нужно установить режим записи: «Подчинение регистратору» и выбрать документ, которым будет делаться запись в регистр. Тогда в регистре появится поле «Регистратор», где будет храниться информация, каким документом была сделана запись. Также регистратор можно использовать как период, для этого укажите в поле "Периодичность" - "По регистратору". Подчинение регистратору делают, когда требуется жестко привязать регистр к документу и изменение записей в регистре в ручном режиме становится недоступным.

Документов, которые будут в роли регистраторов, может быть несколько. Для того чтобы добавить регистратор, нужно зайти в свойства нужного регистра сведений, перейти на вкладку «Регистраторы» и установить галочки напротив документов, которые будут выполнять роль регистратора.

Посмотреть движения, которые делает регистратор можно из документа. Для этого нужно зайти в интересующий документ, нажать: Перейти – Движения документа по регистратору.

В свойствах регистра добавить права, их можно назначить на вкладке «Права». Затем в списке ролей нужно выбрать роль, которой требуется добавить права на регистр, и в списке прав установить права для выбранной роли.

### *Уникальность записей*

Уникальность записи зависит от периода и измерений. Например, если вы хотите записать в регистр «Цены номенклатуры» запись с одинаковыми измерениями, в один и тот же день, то у вас это не получится и программа вызовет ошибку, так как пе-

риодичность регистра в пределах дня. Если периодичность задана регистратором, он так же участвует в уникальности записи. Для непериодических и независимых регистров уникальность зависит от комбинации измерений.

### *Формы*

Для просмотра записей используйте форму списка, в ней можно установить отбор, по интересующим полям, посмотреть историю записей и изменить их через форму записи. Посмотреть записи регистра можно следующим образом: в верхнем меню нажать кнопку «Операции» - «Регистры сведений». В открывшемся окне выберете нужный регистр. После этого откроется форма списка в виде таблицы, где каждая строка является уникальной записью.

Для редактирования / создания используйте форму записи, если запись подчинена регистратору, тогда поле будет недоступно и форму создать не получится.

Добавлять формы нужно в конфигураторе, зайдя в регистр сведений, во вкладке «Формы» и нажать на «лупу» у нужного вида формы. Далее откроется окно, где можно настроить поля будущей формы (расположение, названия и прописать функционал).

### *Измерения, ресурсы и реквизиты*

Измерения предназначены для формирования уникальности записи, по ним в дальнейшем можно осуществлять отбор и делать срез по конкретному измерению. Сочетание измерений и формирует ключ записи. Лучше не создавать большое количество измерений, чтобы таблица не разрасталась и в процессе работы с ней не тормозила.

У измерений есть галочка «Ведущее», если она установлена, то запись будет храниться в базе данных, пока существует данное измерение. Ведущих измерений можно сделать несколько. Например, в регистре сведений «Цены номенклатуры» ведущим измерением является номенклатура, если удалить номенклатуру, которая участвует в записи, тогда автоматически удалится запись в регистре сведений по этой номенклатуре.

Ресурсы предназначены для хранения суммовой информации: количество, цена и т.д. Ресурсы в дальнейшем будем полу-

чать на определенный период времени (если регистр периодический) по измерениям.

Реквизиты в большинстве случаев предназначены для хранения дополнительной информации, они не принимают участие в уникальности записи. Например, в реквизиты можно заносить информацию такую, как автор, комментариев и т.д.

С регистром сведений можно производить следующие действия:

- запись в регистр сведений 1С;
- отбор в регистре сведений 1С;
- удаление записи в регистре сведений 1С.

#### *Особенности*

- Уникальность записей по набору измерений: каждая запись в регистре сведений - это новое значение ресурса.

- Записи регистра сведений могут быть как периодические, так и нет.

- Регистр сведений может быть зависимым и независимым от регистратора.

- Есть возможность сделать срез первых и последних записей на нужную дату. Это реализовано виртуальными таблицами: «СрезПервых» и «СрезПоследних». Для использования этих таблиц можно воспользоваться как отбором, так и запросом (в конструкторе запросов вы уведите эти виртуальные таблицы и сможете сделать по ним запрос). Эти таблицы будут доступны, если регистр сведений периодический.

Пример. Регистр «Цены номенклатуры» - периодический регистр сведений, записи производиться по регистратору. Периодичность установлена в пределах дня. Значит, цену можно изменить один раз в день по уникальным в течение дня измерениям.

Регистр подчинен документу «Установка цен номенклатуры». Значит, запись в регистр происходит из этого документа. Движения по конкретному документу можно посмотреть из формы документа «Установка цен номенклатуры».

Регистр предназначен для хранения информации по цене номенклатуры с измерениями "Тип цен", "Номенклатура" и "Характеристика номенклатуры". Ведущими измерениями являются

все три поля измерения, по ним можно будет делать отбор при выборке. Имеет особое свойство, не используемое в других видах регистров - периодичность.

Может не иметь регистратора, то есть быть независимым, в этом случае записи производятся непосредственно в регистр, минуя регистрирующий документ (то самое исключение из общей схемы использования регистров в 1С). Тогда как остальные виды регистров должны иметь хотя бы один документ-регистратор.

Кроме того, данный вид регистра имеет автоматический контроль уникальности записей по периоду (периодичность, указанная в свойствах регистра) и измерениям. То есть среди записей регистра не может быть более одной записи с одинаковыми показателями период+измерение+регистратор (если он есть). Уникальность записей в других видах регистров осуществляется по регистратору.

Платформа формирует виртуальные таблицы только для периодических регистров сведений. Доступны следующие типы: СрезПервых, СрезПоследних.

### 2.4.3. Регистры накоплений

Регистр накопления - объект конфигурации, предназначен для описания структуры накопления данных. На основе объекта конфигурации Регистр накопления платформа создает в базе данных таблицы, в которых будут накапливаться данные, поставляемые различными объектами базы данных. Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую заданную в конфигураторе структуру.

На основании таблицы движений регистра накопления система рассчитывает таблицу итогов регистра, которая хранит в базе данных итоги на момент времени последнего движения (актуальные итоги).

Основным назначением регистра накопления является накопление числовой информации в разрезе нескольких измерений, которые описываются разработчиком в соответствующем объекте

конфигурации Регистр накопления и являются подчиненными объектами конфигурации (рис. 8).



Рис. 8. Структура хранения регистра накопления

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем. Разработчик может при необходимости предоставить пользователю возможность редактировать регистр накопления. Но предназначение регистра накопления заключается в том, чтобы его модификация производилась на основе алгоритмов работы других объектов базы данных, а не в результате непосредственных действий пользователя.

Регистр накопления делится на два подвида Остатки и Обороты. Данные регистра накопления хранятся в БД в виде двух таблиц — таблица движений и таблица итогов. Обращение напрямую возможно только к таблице движений.

#### *Регистры остатков*

Рассмотрим в качестве примера отслеживание взаиморасчетов с покупателями товаров, которые производит или продает предприятие. Для того чтобы оперативно получать информацию о взаимной задолженности предприятия и покупателя, потребуется регистр «Взаиморасчеты», в котором для каждого покупателя будет храниться сумма задолженности. При совершении хозяйственной операции состояние регистра будет соответствующим образом изменяться, каждый раз отражая текущее состояние взаиморасчетов. Регистр «Взаиморасчеты» — это регистр остатков.



### *Регистры оборотов*

Легко получить информацию об объеме закупок, совершенных данным покупателем за какой-либо период времени, из регистра «Взаиморасчеты» нельзя. Можно проанализировать все движения, имеющие отношение к данному покупателю, и вычислить общую сумму закупок. Но когда необходимо получать эти сведения оперативно (например, по условиям договора при достижении определенного объема закупок покупателю должна предоставляться скидка), такой способне подходит.

В этом случае решением проблемы может быть использование регистра оборотов. В таком регистре — назовем его «Объем закупок» — в разрезе покупателей будет храниться информация об объеме закупок (об обороте покупателя). При создании регистра оборотов можно указывать, с какой периодичностью будет накапливаться информация: день, неделя, месяц и так далее.

Теперь при совершении хозяйственных операций необходимо будет изменять не только состояние регистра «Взаиморасчеты», но и регистр «Объем закупок». В этот регистр при совершении клиентом каждой покупки будет заноситься информация о сумме покупки. В результате в регистре «Объем закупок» будет постоянно накапливаться информация об общем объеме закупок клиента.

При чтении данных платформа 1С обращается к таблицам базы данных. Но для регистров платформа 1С на основе реальных таблиц умеет формировать виртуальные таблицы, которые физически в базе данных не хранятся. Это позволяет разработчику вместо того, чтобы делать сложный запрос к реальной таблице, сразу получить данные из виртуальной простейшим запросом. А также исключает возможные ошибки.

Для регистров накопления набор предоставляемых виртуальных таблиц также зависит от вида регистра. Как известно, есть два вида регистров накопления: Остатки и Обороты.

В регистре накопления остатков доступны следующие таблицы: Остатки, Обороты, Остатки и Обороты.

В регистре накопления оборотов доступна только одна виртуальная таблица – Обороты.

#### 2.4.4. Регистры бухгалтерии

Регистр бухгалтерии похож на регистр накопления, но предназначен для систематизации данных о бухгалтерских проводках. Регистр бухгалтерии может использоваться не только для бухгалтерского, но и для любого другого вида учета.

Его основная особенность заключается в возможности учета данных методом двойной записи по принципу Дебет-Кредит. Для реализации возможности формирования проводок Регистр бухгалтерии должен быть связан со специальным объектом - План счетов.

Самым большим набором виртуальных таблиц обладают регистры бухгалтерии:

- Остатки,
- Обороты,
- ОборотыДтКт,
- Остатки и Обороты,
- Субконто,
- Движения с Субконто.

Но скорость обращения к регистрам бухгалтерии самая низкая. Поэтому если есть возможность получить те же остатки или обороты с использованием регистров накопления, то их и надо использовать.

#### 2.4.5. Регистры расчета

Этот вид регистра предназначен не только для хранения, накопления и систематизации данных, но и для реализации сложных механизмов периодических расчетов. Для этого в свойствах регистра расчета необходимо определить еще один объект 1С-план видов расчета. То есть работа регистра этого вида невозможна без определения для него конкретного плана видов расчета.

Можно сказать, что регистр расчета используется и для хранения информации о видах расчета, и для хранения результатов расчетов, и для промежуточных значений расчетов. Основное его предназначение в конфигурациях 1С- это расчеты начислений, например, заработной платы и других выплат сотрудникам.

И для реализации этих задач при определении параметров регистра расчета в нем возможно указать связь с графиком времени, что позволяет производить расчеты в зависимости от того времени, которое задано в этом графике. Сам график времени должен быть определен с помощью соответствующего регистра сведений.

Таким образом, можно сказать, что регистр расчета имеет в итоге самую сложную структуру по сравнению с другими видами регистров в 1С. Здесь также в зависимости от настроек доступны следующие виртуальные таблицы:

ДанныеГрафика,  
ФактическийПериодДействия,  
База<ИмяРегистра>.

#### 2.4.6. Виртуальные таблицы

У некоторых объектов метаданных помимо основной таблицы в базе данных присутствуют виртуальные таблицы. Они облегчают доступ к некоторой информации, содержащейся в основной таблице. Использовать данные виртуальных таблиц можно при помощи запросов, либо при помощи специальных методов встроенного языка 1С 8.

Рассмотрим основные виды виртуальных таблиц 1С для различных объектов метаданных:

СрезПоследних. Метаданные: периодические регистры сведений. Данная таблица позволяет получить последние актуальные данные на указанную дату, такие данные называются срезом последних. Возможно получить срез как в целом по регистру, так и по одному либо нескольким измерениям;

СрезПервых. Метаданные: периодические регистры сведений. Данная таблица позволяет получить первые актуальные данные, появившиеся в регистре в указанную дату либо после нее. Такие данные называются срезом первых. Возможно получить срез как в целом по регистру, так и по одному либо нескольким измерениям;

Остатки. Метаданные: регистры бухгалтерии, регистры накопления с видом регистра остатки. Данная таблица позволяет получить остатки по ресурсам регистра на указанную дату. Воз-

можно получать остатки как в целом по регистру, так и в разрезе определенных измерений, для регистров бухгалтерии в измерения также включаются счет и субконто, относящиеся к данному счету;

**Обороты.** Метаданные: регистры бухгалтерии, регистры накопления. Данная таблица позволяет получить обороты по ресурсам регистра за указанный период. Возможно получать обороты как в целом по регистру, так и в разрезе определенных измерений, для регистров бухгалтерии в измерения также включаются счет, субконто, кор. счет, кор. субконто;

**Остатки и Обороты.** Метаданные: регистры бухгалтерии, регистры накопления с видом регистра остатки. Данная таблица позволяет получать остатки по ресурсам на начало и конец указанного периода, и обороты по ресурсам за указанный период. Возможно получать остатки и обороты как в целом по регистру, так и в разрезе определенных измерений, для регистров бухгалтерии в измерения также включаются счет и субконто, относящиеся к данному счету;

**ОборотыДтКт.** Метаданные: регистры бухгалтерии. Данная таблица позволяет получить обороты по ресурсам регистра за указанный период. Возможно получать обороты как в целом по регистру, так и в разрезе определенных измерений, в измерения также включаются дебетовая и кредитовая части проводки: счет дебета, счет кредита, субконто дебета, субконто кредита и т.д.;

**Движения с Субконто.** Метаданные: регистры бухгалтерии. Данная таблица позволяет получить записи регистра бухгалтерии вместе со значениями субконто.

**Конструктор запросов** позволяет работать с виртуальными таблицами регистров. Если у регистра есть виртуальные таблицы, то они будут находиться в разделе База данных на вкладке Таблицы и поля после основной таблицы регистра;

**Условие Счета.** В данном поле задается условие на поле Счет виртуальной таблицы. В отличие поля Условия для создания условий доступно только измерение Счет;

**Субконто.** В данном поле задается параметр, содержащий массив видов субконто, которые должны анализироваться в данной виртуальной таблице. Например, если вы хотите получить

остатки по 41 счету бухгалтерского учета, но вам не нужен разрез по складам, то в данный параметр можно передать массив, состоящий из одного элемента: Номенклатура из плана видов характеристик Виды Субконто Хозрасчетные.

*Обороты в регистре накопления*

Начало Периода. Параметр, в котором хранится дата начала периода, за который будут браться обороты.

Конец Периода. Параметр, в котором хранится дата окончания периода, за который будут браться обороты.

Периодичность. Определяет группировку по периоду в виртуальной таблице. Если, например, указана периодичность Месяц, то данные в таблице будут сгруппированы по всем измерениям и месяцу, в полях таблицы появится полеПериод, в котором будет храниться первый день месяца, в котором было сделано движение регистра. Если оставить полеПериодичностьпустым, то периода в полях виртуальной таблицы не будет. Помимо периодичности связанной с временными промежутками есть еще несколько ее типов.

Запись. Данные будут выбираться по отдельным записям регистра, точно также, как в полной таблице. В полях появляются Период и Регистратор (документ, который сделал данное движение).

Регистратор. Данные будут сгруппированы по документу, сделавшему движения в регистре. Данная группировка удобна как раз тогда, когда вам необходимо иметь разрез по документам. В полях появляются Период иРегистратор.

Период. Данные группируются по измерениям регистра за весь период оборота.

Авто. Данные группируются до секунды, в полях появляются Регистратор, Период Секунда, ... . Период Год.

*Обороты в регистре бухгалтерии*

Условие Кор Счета. В данном поле задается условие на поле корреспондирующий счет проводки. В отличие поляУсловия для создания условий доступно только измерение КорСчет.

Кор Субконто. В данном поле задается параметр, содержащий массив видов субконто корреспондирующего счета, работает аналогично параметру Субконто.

### *Остатки и обороты в регистре накопления*

Метод Дополнения. Метод дополнения периодов определяет, за какие периоды будут получены движения регистра. Если указано значение Движения, то будут получены периоды, в которых были движения (обороты не равны нулю), если Движения и Границы Периода, то за периоды, по которым были движения, плюс начальный и конечный периоды, если на их начало был остаток. Данный параметр имеет смысл только, если Периодичность не равна Период. Движения и Границы Периода является значением по умолчанию, поэтому если нужен данный метод дополнения, поле можно оставить пустым.

### *Остатки и обороты в регистре бухгалтерии*

Все параметры, используемые в данной таблице, были описаны в предыдущих пунктах.

### *Обороты Дебет Кредит*

Условие СчетаДт. В данном поле задается условие на поле СчетДт виртуальной таблицы, это счет левой части бухгалтерской проводки.

Условие СчетаКт. В данном поле задается условие на поле СчетКт виртуальной таблицы, это счет правой части бухгалтерской проводки.

СубконтоДт. В данном поле задается параметр, содержащий массив видов субконто, которые должны анализироваться для левой (дебетовой) части проводки.

СубконтоКт. В данном поле задается параметр, содержащий массив видов субконто, которые должны анализироваться для правой (кредитовой) части проводки.

### *Движения с субконто*

Упорядочивание. В данном параметре можно задать поля сортировки записей виртуальной таблицы. Упорядочивать записи можно только в том случае, если заполнен параметр Первые.

Первые. В данном параметре указывается число первых записей, которые будут взяты из регистра в виртуальную таблицу. Данный параметр необходим для упорядочивания элементов виртуальной таблицы 1С.

#### 2.4.7. Планы видов характеристик

Это прикладные объекты конфигурации, предназначенные для хранения информации о характеристиках различных объектов. С их помощью пользователь может создавать всевозможные характеристики, описывать тип этих характеристик и задавать их значения.

Например, для того чтобы описывать товары произвольным количеством произвольных характеристик (цвет, размер, запах и т.д.).

План видов характеристик имеет структуру – справочник (описание структуры приведено ранее в п. 2.2.3). Отличие в том, что план видов имеет узкую «специализацию». Это, главным образом, проявление в хранении информации, т.е. хранится информация о том, какими видами характеристик может описываться какой-либо объект базы данных. В структуру информации входят два основных поля: название вида характеристики и ее тип характеристики. В эти поля разработчик или пользователь, особенно пользователь, могут внести любое требуемое количество видов характеристик.

При необходимости в рамках некоторого проекта создания «нового» набора из имеющихся и возможных типов значений, у объекта конфигурации План видов характеристик существует свойство Тип значения характеристик. Это свойство определяет составной тип данных, куда входят все типы, которые могут понадобиться при указании типа значения характеристики.

Рассмотрим ситуацию, когда пользователю недостаточно уже существующих типов данных конкретной системы или проекта. Например, требуется вести учет пациентов в лечебном учреждении по диагнозу, которого в справочнике Заболевание в конфигурации отсутствует. В этом случае необходимо временно обратиться к разработчику, который создаст и укажет в качестве свойства объекта конфигурации План видов характеристик – дополнительное значение характеристик (одну или несколько). И после этого пользователь сможет работать с этим вспомогательным справочником.

Создав новый вид характеристики Заболевание, пользователь сможет дать необходимые значения в справочнике дополни-

тельных значений характеристик (здесь ключевым является понятие дополнительных). То есть этот справочник для плана видов характеристик является подчиненным. При необходимости создать новый вид характеристик Симптомы и его значения, разработчик будет создавать их в том же самом справочнике дополнительных характеристик. И они не будут «смешиваться» со значениями диагноза.

План видов характеристик не имеет внутренних предопределенных механизмов привязки вида характеристики к тому объекту, который он должен описывать. Он лишь предоставляет возможность разработчику и пользователю описать некий набор характеристик и задать их тип. Каким образом хранить соответствие конкретного вида характеристик или значения характеристик конкретному объекту базы данных, решает сам разработчик в зависимости от создаваемого прикладного решения.

### **Контрольные вопросы**

1. Опишите объект конфигурации «Регистры». Для чего он предназначен.
2. В чем необходимость использования регистров? Каким образом структурируется в них информация, которая уже содержится и в других объектах?
3. Для чего нужны измерения регистра, ресурсы и реквизиты?
4. Что такое «Движения регистра» и что такое «Регистратор»?
5. Для чего предназначен объект конфигурации «Регистр сведений»?
6. Какими особенностями обладает объект конфигурации «Регистр накоплений»?
7. На какие подвиды делится «Регистр накопления»?
8. В чем главные отличия «Регистра сведений» от «Регистра накоплений»?
9. Что такое «Регистр бухгалтерии»?
10. Что такое «Регистр расчетов»?
11. В чем отличие виртуальных таблиц от обычных?
12. Для чего предназначен объект конфигурации «Планы видов характеристик»?



## 2.5. Отчеты и запросы в «1С: Предприятии»

### 2.5.1. Отчеты

Для получения бухгалтерских итогов, а также другой сводной или детальной информации в «1С: Бухгалтерии» используются отчеты. Количество и состав отчетов, которые могут быть получены при использовании системы, целиком определяется при конфигурировании.

В списке База данных представлены таблицы для создания запроса. На основе их данных мы имеем возможность построить отчет.

Объект конфигурации Отчет предназначен для описания алгоритмов, при помощи которых пользователь сможет получать необходимые ему выходные данные. Алгоритм формирования выходных данных описывается при помощи визуальных средств или с использованием встроенного языка. В реальной жизни объектам конфигурации Отчет соответствуют всевозможные таблицы выходных данных, сводных данных, диаграммы и пр.

1. Макет. Объект конфигурации Макет предназначен для хранения различных форм представления данных, различных данных, которые могут потребоваться каким-либо объектам конфигурации или всему прикладному решению в целом.

Макет может содержать табличный или текстовый документ, двоичные данные, HTML-документ или ActiveDocument, графическую или географическую схему, схему компоновки данных или макет оформления схемы компоновки данных.

Макеты могут существовать как сами по себе (общие макеты), так и быть подчинены какому-либо объекту конфигурации.

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.

Так как у отчета, который мы создаем, еще не существует схемы компоновки данных, платформа предложит создать новую

схему. Схема компоновки данных с точки зрения конфигурации является макетом, поэтому будет открыт конструктор макета, предлагающий выбрать единственный тип макета – Схема компоновки данных.

2. Схема компоновки данных. Платформа создаст новый макет, содержащий схему компоновки данных, и сразу же откроет конструктор схемы компоновки данных. Конструктор обладает большим количеством возможностей для визуального проектирования отчетов, но мы сейчас воспользуемся только самыми простыми его возможностями и определим те данные, которые хотим видеть в результате работы нашего отчета.

3. Набор данных. Добавим новый набор данных – запрос. Для этого нажмем кнопку Добавить или вызовем контекстное меню ветки Наборы данных.

4. Конструктор запроса – инструмент, созданный для помощи разработчику, позволяющий визуально конструировать запрос. Даже пользователь, не знакомый с языком запросов, может с помощью конструктора создать синтаксически правильный запрос.

## 2.5.2. Система компоновки данных

Одним из важным инструментом платформы «1С: Предприятие» является система компоновки данных (СКД). СКД – мощный механизм платформы 1С для разработки отчетов.

Отчеты в 1С – это специальный объект метаданных, предназначенный для формирования удобного для пользователя представления данных из таблиц базы данных 1С и для вывода на печать. Отчеты 1С создаются, используя язык запросов.

Запросы – самый мощный и эффективный инструмент получения данных. Язык запросов позволяет в удобном виде получать данные из базы данных.

Любой отчет, как правило, подразумевает получение сложной выборки данных, сгруппированных и отсортированных определенным образом.

Чаще всего исходные данные, необходимые для отчета, находятся в базе данных. Для того чтобы указать системе компо-

новки данных, какая информация и откуда должна быть получена, используется язык запросов системы «1С: Предприятие 8».

На этапе разработки отчета можно задать стандартные настройки отчета для того, чтобы пользователь мог сразу же запустить отчет на выполнение. В то же время пользователь может самостоятельно изменить настройки отчета и выполнить его. При этом система компоновки данных сгенерирует другой запрос и другим образом представит конечные данные – в соответствии с новыми настройками, заданными пользователем.

#### *Способы доступа к данным*

Система «1С: Предприятие 8» поддерживает два способа доступа к данным, хранящимся в базе данных:

- объектный (для чтения и записи);
- табличный (для чтения).

Объектный способ доступа к данным реализован посредством использования объектов встроенного языка. Важной особенностью объектного способа доступа к данным является то, что, обращаясь к какому-либо объекту встроенного языка, мы обращаемся к некоторой совокупности данных, находящихся в базе данных, как к единому целому.

Табличная техника предназначена для получения информации из базы данных по некоторым условиям (отбор, группировка, сортировка, объединение нескольких выборок, расчет итогов и т. д.). Табличная техника оптимизирована для обработки больших объемов информации, расположенной в базе данных, и получения данных, отвечающих заданным критериям.

#### *Работа с запросами*

Для работы с запросами используется объект встроенного языка Запрос. Он позволяет получать информацию, хранящуюся в полях базы данных, в виде выборки, сформированной по заданным правилам.

#### *Источники данных запросов*

Исходную информацию запрос получает из набора таблиц. Эти таблицы представляют разработчику данные реальных таблиц базы данных в удобном для анализа виде. Все таблицы, которыми оперирует язык запросов, можно разделить на две большие группы: реальные таблицы и виртуальные таблицы (рис. 9).

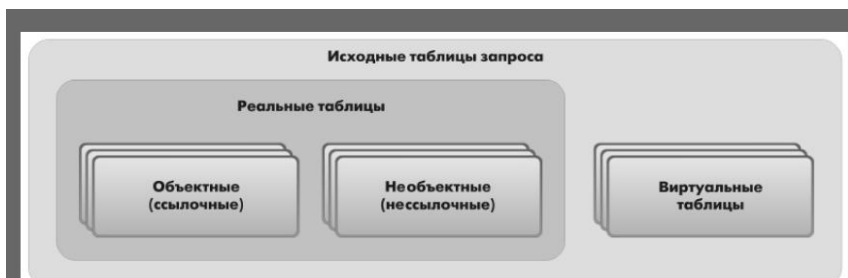


Рис. 9. Таблицы запросов

Посмотреть состав таблиц, доступных для запроса и их описание можно в синтакс-помощнике в разделе Работа с запросами - Таблицы запросов. Отличительной особенностью реальных таблиц является то, что они содержат данные какой-либо одной реальной таблицы, хранящейся в базе данных.

Однако общим для всех виртуальных таблиц является то, что им можно задать ряд параметров, определяющих, какие данные будут включены в эти виртуальные таблицы. Набор таких параметров может быть различным для разных виртуальных таблиц и определяется данными, хранящимися в исходных таблицах базы данных.

Реальные таблицы подразделяются на объектные (ссылочные) и необъектные (нессылочные). В объектных (ссылочных) таблицах представлена информация ссылочных типов данных (справочники, документы, планы видов характеристик и т. д.). А в необъектных (нессылочных) – всех остальных типов данных (константы, регистры и т. д.).

Отличительной особенностью объектных (ссылочных) таблиц является то, что они включают в себя поле Ссылка, содержащее ссылку на текущую запись. Кроме этого, для таких таблиц возможно получение пользовательского представления объекта. Эти таблицы могут быть иерархическими, и поля таких таблиц могут содержать вложенные таблицы (табличные части).

### *Язык запросов*

Алгоритм, по которому данные будут выбраны из исходных таблиц запроса, описывается на специальном языке – языке запросов. Текст запроса может состоять из нескольких частей:

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

Обязательной частью запроса является только первая – описание запроса. Все остальные присутствуют по необходимости. Описание запроса определяет источники данных, поля выборки, группировки и т. д. Объединение запросов определяет, как будут объединены результаты выполнения нескольких запросов.

Упорядочивание результатов определяет условия упорядочивания строк результата запроса. Автоупорядочивание позволяет включить режим автоматического упорядочивания строк результата запроса. Описание итогов определяет, какие итоги необходимо рассчитывать в запросе и каким образом группировать результат.

Следует заметить, что в случае, когда язык запросов используется для описания источников данных в системе компоновки данных, секция описания итогов языка запросов не используется. Это связано с тем, что система компоновки данных самостоятельно рассчитывает итоги на основании тех настроек, которые сделаны разработчиком или пользователем.

Применение различных синтаксических конструкций языка запросов подробно описано во встроенной справке в режиме Конфигуратор: Справка - Содержание справки- «1С: Предприятие» - Встроенный язык - Работа с запросами.

### *Система компоновки данных*

Система компоновки данных предназначена для создания произвольных отчетов в системе «1С:Предприятие» и состоит из нескольких основных частей.

Исходные данные для компоновки отчета содержит в себе схема компоновки данных. Это наборы данных и методы работы с ними (рис. 10).

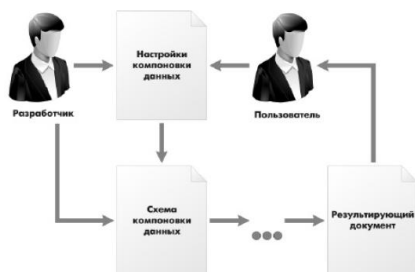


Рис. 10. Общая схема работы с системой компоновки данных

Разработчик создает схему компоновки данных, в которой описывает текст запроса, наборы данных, связи между ними, доступные поля, параметры получения данных, и задает первоначальные настройки компоновки – структуру отчета, макет оформления данных и др.

Отчет системы компоновки данных, который получит пользователь, представляет собой не просто таблицу записей, удовлетворяющих условиям запроса. Отчет системы компоновки имеет сложную иерархическую структуру и может состоять из различных элементов, таких как группировки, таблицы и диаграммы.

При этом пользователь может изменить существующую структуру отчета или вообще создать совершенно новую структуру отчета. Может настроить необходимый ему отбор, оформление элементов структуры отчета, получить расшифровку по каждому элементу и т. д.

На основе схемы компоновки и имеющихся настроек компоновщик макета создает макет. Это этап подготовки к исполнению отчета. Макет компоновки данных является уже готовым заданием для выполнения процессором компоновки. Он содержит необходимые запросы, макеты областей отчета и др.

Процессор компоновки данных выбирает данные из информационной базы согласно макету компоновки, агрегирует и оформляет эти данные.

Результат компоновки обрабатывается процессором вывода, и в итоге пользователь получает результирующий табличный

документ. Эту последовательность работы можно представить в виде следующей схемы (рис. 11).

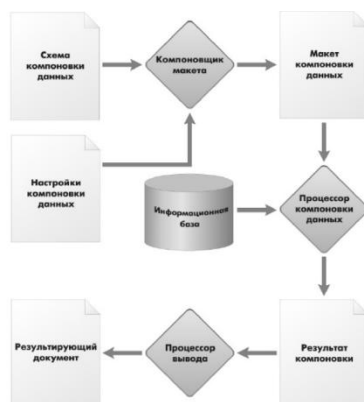


Рис. 11. Схема работы системы компоновки

Алгоритм взаимодействия этих объектов выглядит следующим образом. Разработчик создает схему компоновки данных и настройки по умолчанию. В общем случае на основе одной схемы компоновки данных может быть создано большое количество различных отчетов. Настройки компоновки данных, создаваемые разработчиком или изменяемые пользователем, определяют, какой именно отчет будет получен в конкретном случае.

При создании запроса платформа предоставляет нам в качестве источников данных некоторое количество виртуальных таблиц. Название «виртуальные» полностью соответствует их сути, поскольку эти таблицы, в свою очередь, также являются результатом запроса, который система формирует в момент выполнения соответствующего участка кода.

### 2.5.3. Запросы и язык запросов

Запросы – это мощный инструмент, служащий для быстрого получения и обработки данных, содержащихся в различных объектах информационной базы 1С.

Запрос создается как отдельный объект, который имеет обязательный атрибут Текст, куда собственно и помещается сам запрос. Кроме этого, в запрос могут быть переданы различные параметры, необходимые для его выполнения. После того, как текст и параметры запроса заполнены, запрос необходимо выполнить и поместить результат выполнения в выборку или таблицу значений.

Для создания запросов с 1С есть очень удобный встроенный механизм – конструктор запросов. Он содержит следующие основные вкладки:

- «Таблицы и поля» - содержит поля, которые необходимо выбрать и их источники;
- «Связи» - описывает условий для конструкции СОЕДИНЕНИЕ;
- «Группировка» - содержит описание конструкций группировок и суммируемых полей по ним;
- «Условия» - отвечает за отборы данных в запросе;
- «Дополнительно» - дополнительные параметры запроса, такие как ключевые слова команды «ВЫБРАТЬ» и пр.;
- «Объединения/Псевдонимы» - указываются возможности объединения таблиц и задаются псевдонимы (конструкция «КАК»);
- «Порядок» - отвечает за сортировку результата запросов;
- «Итоги» - аналогична вкладке «Группировка», но применяется для конструкции «ИТОГИ».

Простейшие и наиболее часто применяемые запросы служат для получения данных из какого-то источника. Источником могут являться практически все объекты, содержащие какие-либо данные: справочники, документы, регистры, константы, перечисления, планы видов характеристик и т.д. Из этих объектов с помощью запроса можно получать значения реквизитов, табличных частей, реквизитов табличных частей, изменений, ресурсов и т.д.

Для получения текста запроса часто бывает удобно пользоваться Конструктором запросов на языке запросов.

Язык запросов 1С – один из важнейших пунктов в изучении 1С-программирования. Он является одним из основополагающих механизмов 1С 8 для разработчиков. При помощи



запросов можно быстро получить любые данные, хранящиеся в базе. Его синтаксис очень похож на SQL, но есть и отличия.

Основные достоинства языка запросов 1С 8 перед SQL:

- разыменование ссылочных полей (обращение через одну или несколько точек к реквизитам объектов);
- работа с итогами очень удобная;
- возможность создавать виртуальные таблицы;
- запрос можно писать как на английском, так и на русском языках;

• возможность блокировать данные для исключения взаимных блокировок.

Недостатки языка запросов в 1С:

- в отличие от SQL, в 1С запросы не позволяют изменять данные;
- отсутствие хранимых процедур;
- невозможность преобразования строки в число.

### **Контрольные вопросы**

1. Что такое система компоновки данных и для чего она предназначена?

2. Охарактеризуйте объект языка запросов «1С Отчет» и для чего он предназначен?

3. Что из себя представляет объект встроенного языка «Запрос», для чего предназначен?

4. Дайте понятие опции «настройка компонентов данных». Для чего она предназначена?

5. Проведите сравнительный анализ реальных и виртуальных таблиц. В чем отличия между ними?

6. Что является источником 1С объекта запрос?

7. Что такое параметры виртуальной таблицы?

8. Конструктор запроса. Основное назначение опции. Как ее использовать?

9. Составьте алгоритм выбора данных для отчета за определенный период.

10. Можно ли использовать в отчете данные нескольких таблиц? Если да, то как?

11. Приведите пример группировки в структуре отчета. Представьте ответ в виде последовательности этапов выполнения группировки.

12. Ресурсы в системе компоновки данных. Основное назначение опции. Как используется?

13. Опишите последовательность действий при выводе данных в виде таблицы.

14. Как сделать отчет универсальным?

## **2.6 Встроенный язык программирования «1С: Предприятие»**

### **2.6.1. Общие сведения**

Встроенный язык 1С относится к объектно-ориентированным языкам. Он является важной частью технологической платформы «1С: Предприятие 8». Основным достоинством языка является то, что он дает возможность для описания собственных алгоритмов использовать естественный язык и при этом позволяет реализовывать логические алгоритмы при сравнительно невысокой вычислительной сложности. Тем самым, язык 1С делает прикладные решения понятными не только разработчикам, но и пользователям и дает возможность быстрого изменения алгоритмов при их функционировании.

Встроенный язык 1С имеет много общих черт с другими языками, такими как Pascal, JavaScript, Basic, что облегчает его освоение начинающими разработчиками. Однако язык 1С не является прямым аналогом какого-либо из перечисленных языков.

Вот лишь некоторые наиболее значимые особенности встроенного языка:

- предварительная компиляция – перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код;

- кэширование скомпилированных модулей в памяти;

- мягкая типизация – тип переменной определяется типом значения, которое она содержит, и может изменяться в процессе работы;

- отсутствие программного описания объектов конфигурации – разработчик может использовать либо встроенные в платформу объекты, либо объекты, созданные системой в результате визуального конструирования прикладного решения.

Назначение встроенного языка в системе «1С: Предприятие» определяется идеологией создания прикладных решений. Прикладные решения в «1С: Предприятии» не кодируются целиком. Большая часть прикладного решения создается разработчиком путем визуального конструирования – создания новых объектов конфигурации, задания их свойств, форм представления, взаимосвязей и пр. Встроенный язык используется лишь для того, чтобы определить поведение объектов прикладного решения, отличное от типового, и создать собственные алгоритмы обработки данных.

Встроенный язык поддерживает работу с большим количеством разнообразных объектов. Безусловно, основную группу объектов составляют прикладные объекты, позволяющие описывать алгоритмы функционирования бизнес-логики.

Однако не менее важной группой являются объекты, предназначенные для хранения временных наборов данных в течение сеанса работы пользователя, которые можно увидеть в синтаксис-помощнике (рис. 12). Как правило, они служат для вспомогательного сбора, группировки, анализа и обработки информации.

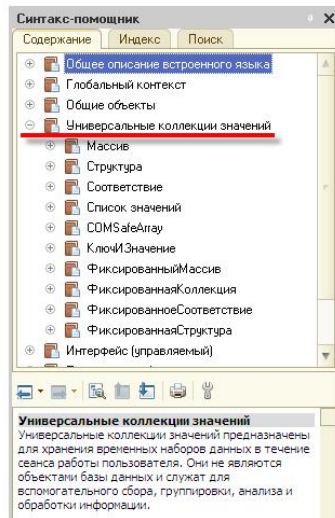


Рис. 12. Синтаксис-помощник

Перечислим кратко их возможности.

### *Массив*

Представляет собой пронумерованную коллекцию значений произвольного типа. К элементу массива можно обращаться по его индексу. В качестве элементов массива могут выступать, в частности, другие массивы. Это позволяет создавать многомерные массивы.

### *Структура*

Представляет собой поименованную коллекцию, состоящую из пар ключ-значение. Ключ может быть только строковым, значение – произвольного типа. К элементу структуры можно обращаться по значению его ключа, т. е. по имени. Обычно используется для хранения небольшого количества значений, каждое из которых имеет некоторое уникальное имя.

### *Соответствие*

Также как и структура, представляет собой коллекцию пар ключ-значение. Однако (в отличие от структуры) ключ может быть практически любого типа.

### *Список значений*

Используется, как правило, для решения интерфейсных задач. Позволяет строить динамические наборы значений и манипулировать ими (добавлять, редактировать, удалять элементы, сортировать). Он может содержать значения любого типа, кроме того, в одном списке типы хранимых значений могут быть разными. Например, список значений может использоваться для выбора конкретного документа из списка возможных документов, сформированного по сложному алгоритму.

### *Таблица значений*

Позволяет строить динамические наборы значений и манипулировать ими. Она может быть наполнена значениями любого типа, и в одной таблице типы хранимых значений могут быть разными. Одним из примеров использования таблицы значений может служить организация представления в форме списка элементов «Справочника», отображенных по сложному алгоритму.

### *Дерево значений*

Представляет собой динамически формируемый набор значений любого типа, похожий на таблицу значений. В отличие от таблицы значений строки дерева значений могут образовывать иерархические структуры: каждая строка дерева может иметь набор подчиненных строк, каждая из подчиненных строк, в свою очередь, также может иметь набор подчиненных строк и так далее. При этом поиск значений, сортировка, получение итогов могут осуществляться либо по текущему уровню иерархии, либо, включая все подчиненные.

### *COMSafeArray*

Представляет собой объектную оболочку над многомерным массивом `SAFEARRAY` из `COM`. Позволяет создавать и использовать `SAFEARRAY` для обмена данными между `COM`-объектами.

### *Фиксированный массив*

Неизменяемый массив. Массив заполняется системой при инициализации объектов данного типа или разработчиком, с помощью конструктора.

### *Редактор текстов и модулей*

Для создания и изменения текстов на встроенном языке разработчик может использовать редактор текста имодуля, обладающий удобными средствами создания, редактирования и синтаксической проверки модулей.

## 2.6.2. Конструкции языка

### *Значение, тип и представление*

Любая программа занята исключительно тем, что выполняет какие-то действия со значениями. Эти действия делятся на три большие группы. Сначала программа получает какие-то значения с диска, с клавиатуры и пр. Затем она что-то делает со значениями, которые она получила. Например, вы ввели время начала занятий и время окончания занятий. Тогда программа может из одного значения вычесть другое и получить третье значение — продолжительность ваших занятий. В концов программа завершает свою работу тем, что выводит эти значения, т.е. показывает их на экране, записывает на диск, печатает на принтере или передаёт другому компьютеру. В программе существует много самых разных значений. Чтобы программа могла что-то делать с этими значениями, она должна знать их тип. С типами мы уже знакомы в п. 2.1.5.

Представление— это форма демонстрации значения на экране понятной для пользователя. Каждый тип имеет собственное представление. Например, представление типа Число — это последовательность цифр 346,45 или 58,2. Представление слов, которые написаны в модуле, бывают разного цвета. Синим цветом, платформа выделяет то, что ей известно. То, что она знает и умеет выполнять. Красным цветом платформа выделяет то, что является обязательным. То, что обязательно должно быть в этой инструкции. То, без чего инструкция будет неправильной. Чёрным цветом платформа выделяет значения, записанные прямо в тексте программы. Такие значения называются литералами.

### *Инструкции*

Во встроенном языке существует 5 основных инструкций, которые применяются наиболее часто. Это:

- инструкция присваивания;
- инструкция Если;
- инструкция вызова процедуры или функции;
- инструкция Цикл;
- инструкция Попытка.

Самая простая инструкция во встроенном языке – это инструкция присваивания. Выглядит она следующим образом (рис. 13). Слева находится имя переменной, а справа — некоторое выражение.

---

```
<Имя переменной> = <Выражение>;
```

---

Рис. 13. Пример инструкции присваивания

Иногда инструкция присваивания используется тогда, когда значение, которое вы получите или вычислите, нужно прямо в этом месте (рис. 14).

---

```
<Имя свойства> = <Выражение>;
```

---

Рис. 14. Пример инструкции присваивания

Слева находится имя свойства, а справа – снова выражение. Переменная – это специальное место в памяти компьютера, в котором сохраняется одно значение на то время, пока выполняется ваша программа.

Выражение – это математическая, или логическая, или строковая формула, по которой вычисляется значение.

Например, КоличествоЗначений = 6 \* 5.

Выражение обычно состоит из одной или нескольких операций. В этом примере значение – это 30, то есть литерал типа Число и снимиможновыполнять арифметические операции: сложение, вычитание, умножение и деление. В выражениях можно использовать не только литералы, но и переменные.