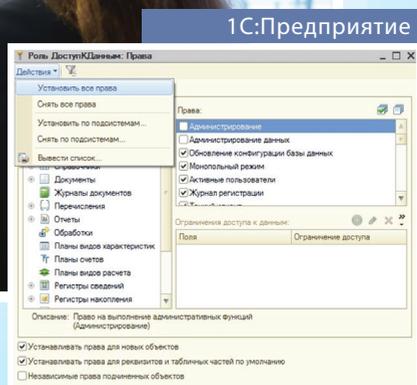


Министерство науки и высшего образования
Российской Федерации
Тольяттинский государственный университет

Н.Н. Казаченок

РАЗРАБОТКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

Электронное учебно-методическое пособие



© Казаченок Н.Н., 2025

© ФГБОУ ВО «Тольяттинский государственный университет», 2025

ISBN 978-5-8259-1682-8

УДК 004.428.4(075.8)

ББК 32.973.432.2я73

Рецензенты:

канд. пед. наук, доцент, доцент кафедры «Вычислительная техника»
Самарского государственного технического университета

Е.В. Панюкова;

канд. пед. наук, доцент кафедры «Прикладная математика
и информатика» Тольяттинского государственного университета

О.Ю. Копша.

Казаченок, Н.Н. Разработка предметно-ориентированных систем : электронное учебно-методическое пособие / Н.Н. Казаченок. – Тольятти : Издательство ТГУ, 2025. – 1 оптический диск. – ISBN 978-5-8259-1682-8.

Учебно-методическое пособие «Разработка предметно-ориентированных систем» разработано в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 09.03.03 «Прикладная информатика».

Составлено в соответствии с государственными требованиями к минимуму содержания и уровня подготовки бакалавра.

Может быть полезно преподавателям, студентам, обучающимся по направлению подготовки 09.03.03 «Прикладная информатика», и всем, кто интересуется данной тематикой.

Текстовое электронное издание.

Рекомендовано к изданию научно-методическим советом Тольяттинского государственного университета.

Минимальные системные требования: IBM PC-совместимый компьютер: Windows XP/Vista/7/8/10; PIII 500 МГц или эквивалент; 128 Мб ОЗУ; SVGA; CD-ROM; Adobe Acrobat Reader; интернет-браузер.

© Казаченок Н.Н., 2025

© ФГБОУ ВО «Тольяттинский государственный университет», 2025

Учебное издание

Казаченок Надежда Николаевна

РАЗРАБОТКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

Редактор *Т.М. Воропанова*

Технический редактор *Н.П. Крюкова*

Компьютерная верстка: *Л.В. Сызганцева*

Художественное оформление,

компьютерное проектирование: *И.И. Шишкина*

В оформлении пособия использованы изображения от DC Studio и pinnacleanimates на сайте ru.freepik.com

Дата подписания к использованию 10.02.2025.

Объем издания 5,4 Мб.

Комплектация издания: компакт-диск, первичная упаковка.

Тираж 50 экз. Заказ № 1-45-23.

Издательство Тольяттинского государственного университета

445020, г. Тольятти, ул. Белорусская, 14,

тел. 8 (8482) 44-91-47, www.tltsu.ru

Содержание

ВВЕДЕНИЕ	6
Модуль 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ	8
Тема 1.1. Предназначение и общие задачи, решаемые предметно-ориентированными информационными системами	8
Модуль 2. ОСНОВНЫЕ МЕТОДЫ РАЗРАБОТКИ И СОПРОВОЖДЕНИЯ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ПЛАТФОРМЕ ФИРМЫ «1С»	14
Тема 2.1. Введение в 1С. Назначение и основные понятия системы «1С:Предприятие» как платформы для построения предметно-ориентированных информационных систем	14
Тема 2.2. Объектная модель данных в «1С:Предприятие»	18
Тема 2.3. Построение модели взаимодействия пользователя с системой в «1С:Предприятие»	27
Тема 2.4. Типы данных в «1С:Предприятие»	35
Тема 2.5. Основы встроенного языка «1С:Предприятие»	41
Тема 2.6. Виды клиентов в «1С:Предприятие»	48
Тема 2.7. Запросы в «1С:Предприятие»	55
ПРАКТИЧЕСКИЕ РАБОТЫ	66
Практическая работа 1. Создание информационной базы. Подсистемы. Справочники	66
Практическая работа 2. Управляемые формы в «1С:Предприятие»	69
Практическая работа 3. Документы в «1С:Предприятие»	75
Практическая работа 4. Механизм Ввод на основании	80
Практическая работа 5. Регистры накопления	85
Практическая работа 6. Регистры сведений	93

Практическая работа 7. Создание произвольного макета отчета в «1С:Предприятие»	97
Практическая работа 8. Объект <i>План</i> видов характеристик в «1С:Предприятие»	108
Практическая работа 9. Настройка прав. Роли и интерфейсы в «1С:Предприятие»	115
Практическая работа 10. Запросы в «1С:Предприятие»	121
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	133
ГЛОССАРИЙ	135

ВВЕДЕНИЕ

Учебно-методическое пособие «Разработка предметно-ориентированных систем» предназначено для студентов направления подготовки 09.03.03 «Прикладная информатика». Пособие может также использоваться студентами других направлений и специальностей при изучении вопросов, связанных с разработкой учетных систем на платформе «1С:Предприятие».

В пособии рассмотрена платформа «1С:Предприятие». Данное средство представляет набор инструментов для разработки приложений по автоматизации учетных систем предприятий различных областей.

Система «1С:Предприятие» является предметно-ориентированной средой, то есть она создавалась как средство решения большого круга задач по автоматизации. Любая организация ведет хозяйственную деятельность. Следовательно, архитектура системы, типы данных, иерархия объектов, методы, применимые к объектам, есть не что иное, как отражение экономических реалий деятельности предприятия.

Целью освоения дисциплины «Разработка предметно-ориентированных систем» является формирование у студентов знаний о предметно-ориентированных информационных системах, в том числе разработанных на основе технологических платформ, и практических навыков их разработки для решения задач автоматизации различных предметных областей.

Задачи изучения дисциплины

1. Рассмотреть понятия и ввести основные определения предметно-ориентированных информационных систем.
2. Ознакомиться с основными принципами построения предметно-ориентированных информационных систем.
3. Сформировать практические навыки работы с основными программными средствами автоматизации в социальной сфере.
4. Расширить профессиональный кругозор студентов и подготовить их к эффективной работе в современной организации.

Изучив данное пособие, студент должен:

- *знать* виды обеспечения информационных систем, специфику информационных объектов и ресурсов, информационных потребностей в предметной области;
- *уметь* обосновывать выбор обеспечения при разработке ИС, адаптировать предметно-ориентированные информационные системы к решению практических задач;
- *владеть* навыками разработки предметно-ориентированных информационных систем.

Учебно-методическое пособие состоит из двух частей: теоретической части, которая представляет собой теоретический материал по темам дисциплины, и комплекта практических работ, предназначенных для применения механизмов платформы «1С:Предприятие» в процессе разработки предметно-ориентированных систем.

В соответствии с рабочей программой в рамках дисциплины текущий контроль успеваемости осуществляется в форме проверки и оценки выполнения практических работ на учебном занятии. Выполненная практическая работа демонстрируется преподавателю. Оформляется отчет и проводится защита: студент отвечает на вопросы по технологии выполнения заданий практической работы. Если работа выполнена, ее защита прошла в установленный срок, студент получает максимальные 5 баллов.

Итоговый контроль успеваемости – зачет, который проводится в соответствии с графиком учебного процесса. Оценка «зачтено» выставляется, если учащийся глубоко усвоил весь программный материал, свободно справляется с практическими заданиями. Оценка «не зачтено» – если учащийся допускает принципиальные ошибки при применении теоретических знаний и практических умений и навыков в ходе выполнения заданий.

Модуль 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Тема 1.1. Предназначение и общие задачи, решаемые предметно-ориентированными информационными системами

Форма проведения занятия – лекция.

Вопросы для обсуждения

1. Подходы при построении предметно-ориентированных информационных систем.
2. Основные черты предметно-ориентированных информационных систем.

При освоении темы необходимо:

1. Освоить учебный материал по теме 1.1.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 1.1.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Информатизация общества представляет собой сложный процесс применения информационных технологий в профессиональной деятельности человека. Уровень взаимодействия информационной службы и остальных подразделений организации влияет на показатели эффективности информатизации.

Согласно ФЗ «Об информации, информатизации и защите информации», принятому в феврале 1995 г., информационная система – это организационно-упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы (процессы сбора, обработки, накопления, хранения, поиска и распространения информации).

Предметная область представляет собой определенную сферу деятельности человека с его реальными практическими задачами, которые автоматизирует информационная система.

Определение границ предметной области влияет на функционал разрабатываемой информационной системы. Если границы предметной области определены широко, то функционал системы будет сложным в использовании. Эффективность работы с такой системой будет снижена. Если границы предметной области определены недостаточно, то функционал информационной системы будет, наоборот, неполноценным, многие решения многих задач система не будет поддерживать.

«Существенными с точки зрения организации и осуществления управления информационными процессами любой сферы деятельности человека являются особенности, связанные с «человеческим фактором». Из всех видов технологий информационная технология предъявляет самые высокие требования к «человеческому фактору», оказывая принципиальное влияние на квалификацию, содержание труда, физическую и умственную нагрузку, профессиональные перспективы и уровень социальных отношений в коллективе. Эффект от внедрения компьютерных технологий зависит от активности пользователей, уровня их обучения, применения новых программ, интеграции в сети и т. д.

Недостаточно создать информационную систему и накопить необходимые информационные ресурсы. Чтобы их эффективно использовать, кроме управления информационными процессами, поддержания работоспособности системы необходимо управлять пользователями, мотивировать их к изучению и активному использованию информационных технологий, отчуждению формируемой ими информации, поиску новых возможностей использования накопленных информационных ресурсов, обеспечению конфиденциальности и сохранности информации.

Для предметно-ориентированных видов деятельности, к которым относится любая сфера деятельности человека, современные информационные технологии предоставляют принципиально новые возможности, в том числе:

- возможность быстрой организации рабочих групп для решения профессиональных задач; появляются новые формы организации рабочих процессов, которые обеспечиваются и поддерживаются взаимодействием через сеть Интернет (в том числе дистанционно);
- удаленный доступ пользователей к «центральным» информационным и вычислительным ресурсам информационной системы предприятия;
- возможность мониторинга и управления рабочими процессами в распределенной системе, появляются возможности для учета, контроля, воздействия на исполнителей, что является одним из существенных факторов эффективности менеджмента» [10].

Реализация этих возможностей связана с необходимостью постоянного анализа, а при необходимости перестройки рабочих процессов и всей организационной структуры предприятия. Особенную роль необходимо уделять также обеспечению защиты потоков информации от несанкционированного доступа.

«Сущность любого производства (экономического объекта) состоит в назначении удовлетворять определенные потребности общества. Все такие объекты состоят из некоторого количества элементов (подсистем), которые взаимодействуют с внешней средой, имеют свою систему управления, все это обеспечивает их существование и выполнение функций.

При взаимодействии организации с внешней средой в ней происходят различные изменения, направленные на достижение поставленной цели. Наличие всех этих изменений определяет систему управления.

Организация – это стабильная формальная социальная структура, которая получает ресурсы из окружающего мира и перерабатывает их в продукты своей деятельности. Всем организациям присущи как индивидуальные, так и общие черты.

Результатом взаимодействия организации со средой являются происходящие в ней различного рода изменения, которые вызывают необходимость управления – такого целенаправленного воздействия на организацию, которое обеспечит достижение поставленных целей.

Управление позволяет в зависимости от особенностей конкретных организаций и целей управления стабилизировать их, сохранить их качественную определенность, поддержать динамическое равновесие со средой, обеспечить совершенствование организации и достижение того или иного полезного эффекта» [11].

«Система управления предприятием функционирует на базе информации о состоянии объекта, его входов и выходов в соответствии с поставленной целью. Управление осуществляется выдачей управленческих воздействий с учетом обратной связи и внешней среды, рынка и вышестоящих органов управления. Назначение управляющей системы – формировать такие воздействия на управляемую систему, которые побуждали бы её принять состояние, определяемое целью управления» [12].

Предметно-ориентированные информационные системы управляют экономическим объектом и имеют следующие характеристики:

- структура информационной системы достаточно сложная, состоящая из нескольких разных иерархических структур, с их сложным функциональным взаимодействием;
- система состоит из элементов или подсистем, которые выделены по определенному признаку, отвечающему конкретным целям и задачам;
- поддержание целостности систем через функционирование элементов системы, подчиненность их единой цели;
- функциональная специфичность и автономность системы;
- наличие многочисленных связей и отношений между подсистемами внутри системы, выстроенными по уровням иерархии;
- развитие системы во времени;
- учет данных особенностей и круг задач объясняют разный состав и различные структуры предметно-ориентированных информационных систем.

Информационные технологии, поддерживающие основные черты предметно-ориентированных информационных систем, следующие:

- построение предметно-ориентированных систем на базе сетевых технологий обработки информации;

- реализация функциональных операций по управлению системой в непрерывном режиме и в реальном времени;
- применение облачных технологий в работе предметно-ориентированных систем с полной функциональной реализацией в них операций по управлению;
- мониторинг решения отдельных задач и системы в целом;
- интеллектуальная поддержка работы системы;
- автоматизация отчетной документации;
- внедрение гибкого пользовательского интерфейса;
- внедрение и использование при разработке специализированных платформ для построения предметно-ориентированных систем.

Совершенствование перечисленных требований позволит качественно повысить уровень использования предметно-ориентированных систем в профессиональной деятельности человека и общества в целом.

Контрольные вопросы

1. Что такое экономический объект?
2. Каковы принципиально новые возможности использования современных информационных технологий в любой сфере деятельности человека?
3. Каковы составляющие информационной системы организации?
4. Какие подходы относят к задачам оперативного уровня управления?
5. Чем отличаются задачи тактического и стратегического уровня управления?
6. Каковы особенности стратегического уровня управления?
7. Каковы основные характеристики предметно-ориентированных систем?
8. Какие требования предъявляются к предметно-ориентированным информационным системам при их разработке?
9. Какие недостатки социальной эффективности внедрения информационных технологий в предметную область можно выявить?
10. Каковы основные черты предметно-ориентированных информационных систем?

Рекомендуемая литература

1. Трофимова, М. В. Предметно-ориентированные информационные системы : учеб. пособие / М. В. Трофимова. — Ставрополь : Издательство Северо-Кавказского федерального университета, 2014. — 187 с.
2. Вдовин, В. М. Предметно-ориентированные экономические информационные системы : учеб. пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. — 3-е изд. — Москва : Дашков и К, 2016. — 385 с. — ISBN 978-5-394-02262-3.

Модуль 2. ОСНОВНЫЕ МЕТОДЫ РАЗРАБОТКИ И СОПРОВОЖДЕНИЯ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ПЛАТФОРМЕ ФИРМЫ «1С»

Тема 2.1. Введение в 1С. Назначение и основные понятия системы «1С:Предприятие» как платформы для построения предметно-ориентированных информационных систем

Форма проведения занятия — лекция.

Вопросы для обсуждения

1. Основные понятия системы «1С:Предприятие».
2. Вопросы организации работы с системой «1С:Предприятие».

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.1.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.1.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Система программ «1С:Предприятие» предназначена для автоматизации управления и учета всех видов деятельности на предприятиях различных отраслей. «1С:Предприятие» включает в себя решения для комплексной автоматизации производственных, торговых и сервисных предприятий, продукты для управления финансами холдингов и отдельных предприятий.

Наиболее широко распространены тиражные решения системы «1С:Предприятие», предназначенные для автоматизации различных учетных задач ведения бухгалтерского учета, расчета зарплаты, управления кадрами, а также другие разнообразные отраслевые и специализированные решения, разработанные ООО «1С», ее партнерами и независимыми организациями.

Основные принципы, применяемые фирмой «1С» в своих разработках, следующие:

1. Прикладные решения, разработанные по требованиям заказчика, должны обеспечивать высокий уровень адаптируемости.
2. Любое типовое прикладное решение может быть доработано в соответствии с требованиями пользователей.
3. Технологии 1С обеспечивают стандартизацию разработки.
4. Использование технологий 1С не требует специальных знаний от разработчика.

Возможности системы «1С:Предприятие», которые позволяют создавать специализированные бизнес-приложения, предоставляет технологическая платформа. Платформа «1С:Предприятие» является фундаментом, основой для построения прикладных решений разной направленности.

Под технологической платформой будем понимать совокупность взаимосвязанных составных частей и специальных механизмов, которые обеспечивают целостное функционирование системы «1С:Предприятие» и решают учетные задачи в определенной предметной области.

Конфигурируемость — это главная особенность технологической платформы «1С:Предприятие».

Под конфигурацией будем понимать совокупность взаимосвязанных объектов с определенной структурой информационных масивов и набор описанных алгоритмов обработки данных.

Различают следующие виды конфигураций:

- типовые конфигурации, разрабатываемые фирмой «1С», предоставляют готовый набор стандартизованных методологических решений, используемых в качестве основы, шаблона при разработке других конфигураций;
- конфигурации, разрабатываемые партнерами «1С», автоматизирующие сравнительно узкие участки учета.

Платформа «1С:Предприятие» с конфигурацией выступает как готовая к использованию программа, ориентированная на определенные классы задач, решаемых на предприятии.

Типовые конфигурации фирмы «1С», ориентированные на наиболее распространенные типы учетных операций в организациях:

«1С:Бухгалтерия предприятия», «1С:Зарплата и управление персоналом», «1С:Управление торговлей», «1С:Управление небольшой фирмой». Существенная часть пользователей применяет типовые конфигурации без каких-либо изменений и доработок.

Готовое решение «1С:Предприятие» является открытым, то есть пользователь может вносить изменения, настраивать приложение под свои собственные нужды как самостоятельно, так и привлекая сторонних специалистов. Исключения составляют решения, которые выпускаются в так называемой «базовой версии».

Технологическая платформа включает в свой состав все средства разработки, необходимые для модификации прикладных решений.

Другими словами, платформа выполняет как функции среды исполнения, так и функции среды разработки, тем самым адаптируя используемое прикладное решение под требования заказчика.

Платформа «1С:Предприятие», в частности входящий в ее состав модуль *Конфигуратор*, позволяет создать собственную уникальную конфигурацию. Конфигурация может быть выполнена самостоятельно любым пользователем. Конфигурация также может быть выполнена на базе типовой конфигурации, которая входит в состав продуктов фирмы «1С».

Разработка уникальных конфигураций позволяет автоматизировать с помощью средств платформы «1С:Предприятие» самые разнообразные задачи по экономической деятельности организации, для выполнения необходимых учетных задач объектов предметной области.

В настоящее время хорошо развита система абонентного обслуживания пользователей системы «1С:Предприятие».

На основании того, что в настоящее время существует объективная тенденция изменения учетного законодательства в Российской Федерации, фирма «1С» постоянно разрабатывает различные методические рекомендации по организации учета, информационному и технологическому сопровождению.

Согласно статье «Информационно-технологическое сопровождение программ «1С:Предприятие», «ежемесячно фирма «1С» выпускает компакт-диск информационно-технологического сопровождения, который содержит:

- методические материалы для пользователей, технологическую информацию по сопровождению систем;
- актуальные типовые формы бухгалтерской отчетности, ежеквартально выпускаемые фирмой «1С» в соответствии с инструкциями Министерства по налогам и сборам и Минфина РФ;
- новые релизы (в пределах той же версии) компонент системы «1С:Предприятие»;
- методики, консультации, руководства, практические рекомендации;
- учебные и методические материалы по эксплуатации, настройке и конфигурированию системы, подготовленные сотрудниками фирмы «1С» и специалистами Финансовой академии при Правительстве РФ;
- систему правовой поддержки, интегрированную с программами семейства «1С:Предприятие». Раздел правовой поддержки составляется компанией «Гарант-Сервис», ежемесячно обновляется и содержит документы, действительно необходимые в работе нашим пользователям: нормативную базу данных по бухучету и налогообложению; бизнес-справки, включая налоговый календарь, справочник бухгалтера, курсы валют и т. д. Особое внимание уделяется удобству работы с информацией: интеграция системы «1С:Гарант» и программ семейства «1С:Предприятие» позволяет подобрать нормативные документы по тому счету или виду начисления зарплаты, с которым в данный момент работает бухгалтер» [13].

Контрольные вопросы

1. Каково основное предназначение платформы «1С:Предприятие»?
2. Что такое типовая конфигурация?
3. Что относят к основным компонентам платформы «1С:Предприятие»?
4. Что такое технологическая платформа «1С:Предприятие»?
5. Что понимают под конфигурацией?
6. Какие различают виды конфигураций?
7. Какие задачи выполняет режим *Конфигуратор* в системе «1С:Предприятие»?

8. Какие задачи выполняет режим «1С:Предприятие» в системе «1С:Предприятие»?
9. Как создать информационную базу в системе «1С:Предприятие»?
10. Как осуществляется информационно-технологическое сопровождение пользователей «1С:Предприятие»?

Рекомендуемая литература

1. Радченко, М. Г. Архитектура и работа с данными «1С:Предприятия 8.2» / М. Г. Радченко, Е. Ю. Хрусталева. — Москва : 1С-Паблишинг, 2011. — 268 с. — (1С. Профессиональная разработка). — ISBN 978-5-9677-1555-6.

Тема 2.2. Объектная модель данных в «1С:Предприятие»

Форма проведения занятия — лекция.

Вопросы для обсуждения

1. Основные понятия объектной модели данных «1С:Предприятие».
2. Сравнительные характеристики объектов конфигурации: справочник, документ, регистр, отчет.

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.2.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.2.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Объектная модель данных представляет собой описание предметной области, ее состав и структуру. Задача объектной модели данных — моделирование общего документооборота и учетных процессов деятельности в целом. Основные объекты, которые входят в состав объектной модели данных, — справочник, документ, регистр, отчет.

Объект конфигурации — не абстрактная конструкция. Каждый объект конфигурации представляет собой аналог реального объекта

из реальной деятельности организации. В этом заключается их прикладная направленность.

Конфигурация системы «1С:Предприятие» – структура взаимосвязанных данных и представляет собой описание объектной модели данных. Разработка конфигурации включает описание различных алгоритмов обработки данных и содержит информацию о представлении данных в интерфейсе пользователя. На основании данного описания система «1С:Предприятие» сформирует базу данных, соответствующую требованиям пользователя.

Платформа «1С:Предприятие» при разработке прикладных решений использует технологию метаданных. Технология метаданных поддерживает визуальное конструирование прикладного решения.

Конфигурация состоит из отдельных логических единиц – объектов конфигурации, называемых прототипами. В конфигурации можно создать только определенные виды объектов, причем их количество не ограничено. Такие объекты располагаются внутри дерева объектов конфигурации.

Древовидная структура, каждая ее ветвь описывает определенную составляющую конфигурации. Состав каждого объекта конфигурации также представляется в виде древовидной структуры, содержащей подчиненные объекты конфигурации.

При добавлении в конфигурацию новый объект конфигурации наследует базовую реализацию прототипа. Платформа в этом случае определяет таблицы в хранилище данных, формирует набор прав для данного объекта, определяет стандартные действия системы для данного объекта конфигурации.

Все добавленные объекты конфигурации представляют собой определенную структуру данных, а также формируют поведение подчиненных объектов. К подчиненным объектам относят реквизиты, табличные части, формы, макеты.

В конфигурации все прикладные объекты обладают свойствами. Эти свойства отражаются в модели предметной области при помощи реквизитов.

Табличные части – это таблицы реквизитов. Табличные части моделируют сложные свойства объектов, которые не могут быть представлены одним-единственным параметром.

Общие объекты обеспечивают моделирование свойств конфигурации в целом и являются вспомогательными.

Разработчик при помощи основных, общих объектов, а также настройки подчиненных объектов формирует структуру и состояние предметной области.

Платформа «1С:Предприятие» на основе объектов конфигурации создает в базе данных таблицы информационной базы, в которых будут храниться данные. В зависимости от состава и значений свойств объекта меняется состав полей таблицы базы данных. Как правило, объект конфигурации и соответствующий ему набор таблиц базы данных принято называть одинаково.

Объекты, сохраняемые в базе данных и содержащие данные конфигурации, можно разбить на четыре блока: блок условно-постоянной информации, блок различных документов, блок обработки данных и вывода информации, блок регистров.

Если в базе данных требуется хранить информацию редко изменяющуюся, то для хранения удобны объекты из блока условно-постоянной информации. К этому блоку относят справочники, перечисления, константы, планы видов характеристик, планы счетов, планы видов расчетов, используемые при решении любых практических задач.

Если требуется хранить информацию о происходящих в организации событиях и различных выполняемых операциях, то есть информация привязана ко времени, то удобно использовать документы. К такой группе также относятся объекты, которые используются для решения вопросов дополнительной функциональности системы:

- журналы – как инструмент визуального группирования информации разных документов;
- последовательности – как инструмент логического группирования информации разных документов;
- нумераторы – как инструмент группирования разных документов для ведения единой нумерации.

Если требуется хранить информацию о состоянии показателей, учитываемых в системе, то для решения таких задач наиболее удобны объекты из группы регистров. Причем такие показатели могут иметь или не иметь привязку ко времени, быть специфичными

или общими, предназначенными к использованию в специальных моделях учета.

Внешние обработки и отчеты не входят в состав прикладного решения и хранятся в отдельных файлах. Обработки и отчеты можно использовать в различных прикладных решениях, их можно проектировать и отлаживать в процессе работы «1С:Предприятие» без необходимости сохранения каждый раз конфигурации прикладного решения.

Рассмотрим более подробно объекты конфигурации.

Константа — объект метаданных, который нужен для хранения постоянной и условно-постоянной информации. Это та информация, которая изменяется крайне редко, например, название организации.

Добавлять новые константы или удалять старые можно только в режиме *Конфигуратор*, а заполнение и модификация значений обычно выполняется пользователями в режиме «1С:Предприятие». При решении прикладных задач посредством констант обычно запоминаются общие для всей информационной базы значения.

Справочник — объект, предназначенный для хранения постоянной информации, которая может иметь множество значений. Заранее значения справочников неизвестны, их вводит пользователь. По сути, справочники необходимы для хранения объектов аналитики учета. Хотя пользователь имеет возможность редактировать содержимое справочников, происходит это сравнительно нечасто.

Система «1С:Предприятие» позволяет создать любое количество справочников разных типов — иерархические или с табличной частью.

Справочники могут быть подчиненными. То есть если справочник подчинен какому-то другому справочнику, то элемент этого справочника не может существовать сам по себе. У этого справочника должен быть Родитель.

Также справочники могут иметь предопределенный элемент, то есть элемент, который был создан разработчиком на этапе конфигурирования этого программного обеспечения.

При этом в режиме *Конфигуратор* формируется структура справочника, а данными он заполняется уже в режиме «1С:Предприятие».

Постоянный объект закреплённых в конфигурации метаданных, сформированный списком, называется перечислением в «1С:Предприятие».

Примерами общеизвестных перечислений, остающихся неизменными, являются: пол — мужской, женский, число — единственное, множественное, месяцы года и так далее. Перечисления не подлежат корректировке в процессе всей работы конфигурации.

Документ — прикладной объект конфигурации, который вводится в систему для фиксации каких-то событий, которые происходят в организации, например, закупка материалов, продажа товара. Так же, как и справочники, документы состоят из реквизитов и табличных частей. Любой документ в системе «1С:Предприятие» обладает уникальным свойством — возможностью проведения. Данное событие влияет на состояние учета внутри организации, формирует проводки, тем самым совершает движения по определенным регистрам (сведений, накопления, бухгалтерии, расчета). Факт проведения означает, что документ обработан полностью.

Если документ не проведен, то он ничего не записал в регистры системы «1С:Предприятие».

Еще одна важная особенность документа — его привязка к шкале времени. Это позволяет отслеживать последовательность событий.

Регистр — объект, предназначенный для хранения информации о состояниях и количествах объектов базы данных.

В состав регистра как объекта конфигурации включаются измерения, ресурсы и реквизиты.

Регистр содержит измерения, то есть свойства, которые обеспечивают анализ в требуемых разрезах накапливаемых данных. Регистр содержит ресурсы, то есть виды накапливаемых данных.

Ресурсы хранят данные регистра, то есть ту информацию, ради хранения которой регистр, собственно, создавался и с которой работает функциональность регистра.

Кроме того, для ситуаций, когда необходимо кроме данных хранить дополнительную информацию о собственно записях регистра, возможно использование реквизитов.

Регистр, как правило, связан с документом и обеспечивает сохранение данных, изменяемых или создаваемых при проведении документа.

Документы записывают свои результаты в таблицы регистра «1С:Предприятие». Документ, который записал движения в регистр, называют регистратором. Дата и время движения почти всегда равны дате документа. Дату движения называют периодом. Сведения из регистров обычно используются для формирования отчетов.

Записи в регистре пользователь не может пометить на удаление, так как они заносятся в него с помощью документа-регистратора при проведении документа. Регистры «1С:Предприятие» — это внутренний, программный объект. Обычный пользователь с ним не работает, только смотрит его результаты при формировании отчетов. Поэтому регистры «1С:Предприятие» в интерфейс пользователя не выводятся и открыть их через меню нельзя.

Регистры «1С:Предприятие» бывают разных видов — регистры накопления, регистры сведений, регистры бухучета, регистры расчета. Цель использования регистра зависит от его вида.

Регистр накопления — это таблица, в которую каждый документ записывает одну или несколько строк своих движений со знаком плюс или минус. Это означает, что итоговое количественное значение регистра изменяется на соответствующую цифру, чтобы в дальнейшем, например, отчет просто отобразил заранее посчитанные итоги.

Регистры накопления бывают двух видов. Все зависит от их предназначения. Выделяют регистры остатков и регистры оборотов.

Регистр остатков — таблица движения документов, которая автоматически вычисляет итоговое значение на начало периода и итоговое значение на конец периода. Регистр остатков используют в том случае, когда у справочника логически может быть остаток.

Регистр оборотов — это таблица движения документов, которая не считает итоговые значения.

Структура таблиц регистра оборотов, хранимых в базе данных, ничем не отличается от регистра остатков. Отличие только в том, что для оборотных регистров не существует понятия «вид движения», такие как приход или расход, а также не применяется понятие «текущие итоги».

Регистр оборотов используют в том случае, когда у справочника логически не может быть остатка. Регистр оборотов не рассчитывает автоматически остатки.

Для регистра оборотов в таблице итогов хранятся посчитанные суммарные обороты ресурсов, в рамках которого были зарегистрированы движения в данном регистре. Обороты хранятся с периодичностью месяц. Итоги формируются на каждую дату, когда было движение. Таблица оборотов использует информацию и из таблиц итогов, и из таблиц движений.

Регистры накопления разрабатываются при необходимости.

Регистр сведений — это таблица, не хранящая движений, таких как приходы и расходы. Регистр сведений обычно используется для хранения информации необъектных данных, развернутой в некоторых разрезах.

Одним из основных факторов для использования регистров сведений для хранения данных является то, что регистры сведений обеспечивают уникальность хранимой информации для конкретных комбинаций значений разрезов. То есть набор значений разрезов является уникальным ключом каждой записи регистра. Сама информация в регистре хранится в виде значений ресурсов. А разрезы хранения информации реализуются посредством измерений регистра.

Одной из возможностей регистра сведений является хранение данных не только в разрезе указанных измерений, но и в хронологическом разрезе. Такие регистры называются периодическими. В результате в структуру хранения данных регистра будет добавлено поле Период. Причем оно будет включено в состав ключа записей регистра.

Таким образом, в регистре сведений можно хранить не просто статические данные, но и историю изменений этих данных.

При такой организации хранения информации появляются следующие возможности:

- получение актуальной информации на момент времени;
- получение информации о значениях данных, заданных после некоторого временного момента;
- получение динамики изменения данных за некоторый период.

На официальном сайте фирмы «1С» регистры бухгалтерии определены как прикладные объекты конфигурации. Они используются в механизме бухгалтерского учета и позволяют вести многоуровневый и многомерный аналитический учет, учет по нескольким пла-

нам счетов, опциональное ведение количественного, суммового и валютного учета по отдельным разрезам аналитики и т. д.

Регистр бухгалтерии обязательно связан с одним из планов счетов, который своими настройками влияет на хранение итогов. На структуру хранения записей регистра бухгалтерии в основном влияют измерения, ресурсы, реквизиты, а также настройки используемых счетов, устанавливающие признаки учета счетов, субконто, вид счета.

Объект «1С:Предприятие» регистр расчета – это прикладной объект конфигурации, служащий для хранения записей о тех или иных видах расчета, которые необходимо выполнить и используемые в механизме сложных периодических расчетов.

Регистр расчета обязательно взаимосвязан с одним из планов видов расчета прикладного решения. Данная связь определяет у каждой записи регистра поле Вид расчета, и механизмы регистра могут отслеживать взаимное влияние записей расчета друг на друга.

Регистр расчета хранит данные не только в разрезе созданных измерений, но и в разрезе времени. Это является причиной существования у каждой записи регистра расчета еще одного обязательного поля – Период действия. При создании регистра расчета разработчик может указать минимальную периодичность, с которой записи будут заноситься в регистр.

Отчет – объект, предназначенный для представления алгоритмов и инструментов получения выходных данных. Отчеты обобщают, формируют и выводят в заранее заданной форме учетную информацию.

Внешние обработки и отчеты представляют собой обработки и отчеты, которые не входят в состав прикладного решения и хранятся в отдельных файлах.

В режиме «1С:Предприятие» внешнюю обработку или отчет можно запустить на выполнение, открыв ее как любой другой файл, хранящийся на диске. Такая обработка или отчет будут работать точно так же, как если бы они входили в состав прикладного решения.

Любой отчет или обработка могут быть преобразованы во внешние, а также наоборот. Кроме этого, внешние обработки и отчеты могут быть добавлены в качестве новых объектов в структуру прикладного решения.

Также в конфигураторе поддерживается возможность сравнения и объединения существующей в конфигурации обработки или отчета с внешней обработкой и отчетом, а также сравнение и объединение двух внешних отчетов или обработок.

Контрольные вопросы

1. Что такое объектная модель данных в «1С:Предприятие»?
2. В чем суть технологии метаданных?
3. Почему объекты конфигурации имеют прикладную направленность?
4. Какие составляющие имеет простая модель данных конфигурации?
5. Какие объекты конфигурации входят в состав условно-постоянного блока данных?
6. Какие отличительные особенности имеет объект Документ?
7. Какие отличительные особенности имеет объект Регистр?
8. Чем объект Регистр накопления отличается от объекта Регистр сведений?
9. Как можно охарактеризовать периодический регистр сведений?
10. Какие отличительные особенности имеет объект Отчет?

Рекомендуемая литература

1. Гладких, Т. В. Разработка прикладных решений для информационной системы 1С: Предприятие 8.2 : учеб. пособие / Т. В. Гладких, Е. В. Воронова. — Воронеж : Воронежский государственный университет инженерных технологий, 2016. — 55 с. — URL: www.iprbookshop.ru/50639.html (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-00032-182-9.
2. Скороход, С. В. Программирование на платформе 1С: Предприятие 8.3 : учеб. пособие / С. В. Скороход. — Ростов-на-Дону [и др.] : Издательство Южного федерального университета, 2019. — 133, [1]с. — URL: znanium.com/catalog/document?id=357443 (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-9275-3315-2.

Тема 2.3. Построение модели взаимодействия пользователя с системой в «1С:Предприятие»

Форма проведения занятия – лекция.

Вопросы для обсуждения

1. Интерфейсные механизмы в «1С:Предприятие».
2. Ролевая модель в «1С:Предприятие».

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.3.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.3.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Существует множество интерфейсных механизмов, предоставляемых платформой «1С:Предприятие»: командный интерфейс, определение доступности команд по функциональным опциям и ограничение доступа к данным. Рассмотрим их.

Составной частью пользовательского интерфейса является командный интерфейс, обеспечивающий доступ пользователей к функциональности компьютерной программы.

Введем определение объекта конфигурации *Подсистема* с точки зрения разработчика.

Подсистема – объект, предназначение которого в формировании пользовательского интерфейса путем группировки объектов конфигурации по определенным правилам.

Подсистемы формируют структуру командного интерфейса. Это объекты конфигурации, с которых начинается разработка нового решения.

Подсистемы бывают верхнего уровня и подчиненные, то есть у подсистемы верхнего уровня может быть несколько подчиненных подсистем.

Для пользователей *Подсистемы* представляются как разделы пользовательского интерфейса. Любой объект конфигурации может принадлежать как одной подсистеме, так и нескольким подсистемам

одновременно. Каждый раздел или подсистема определяет состав определенных команд. Состав команд, включенных в командный интерфейс, определяется правами пользователя, значениями функциональных опций и видимостью команд по ролям пользователей.

На основании прав пользователя определяются строгие границы для каждого пользователя, тем самым это влияет на его возможности по выполнению тех или иных действий. Механизм видимости по ролям помогает для каждого отдельно взятого пользователя настроить только ту часть функциональности системы, которая ему необходима для выполнения задач.

Со стороны пользователя на платформе «1С:Предприятие» реализован эргономичный пользовательский интерфейс, который рассчитан на комфортную работу неподготовленного человека.

Интуитивно понятный пользовательский интерфейс, удобный и гибкий в настройках, представляет собой совокупность сервисов, позволяющих пользователю, используя инструментарий платформы, решать сложные практические задачи.

Диалоговое взаимодействие, различные элементы управления, переход между разделами системы, дизайн экранов, осуществление обратной связи — все перечисленное является элементами пользовательского интерфейса.

Пользовательский интерфейс системы «1С:Предприятие» позволяет в каждом отдельном окне прикладного решения, как основном, так и в вспомогательном, решать конкретные задачи.

В главном окне прикладного решения пользователь осуществляет вызов различных команд, перемещается по приложению.

Дополнительное окно необходимо для работы с объектами информационной базы, например, со справочником и его элементами, для построения отчета, запуска внешних обработок.

В главном окне приложения пользователю предоставляется рабочий стол и структура прикладного решения в виде панели разделов.

Рабочий стол открывается по умолчанию при запуске приложения и отображает наиболее важную информацию для пользователя и часто используемые команды. *Рабочий стол* не связан с подсистемами конфигурации.

Состав выводимой на рабочий стол информации и список доступных команд для пользователя формируются при разработке прикладного решения вручную.

Панель разделов представляет собой перераспределение в интерфейсе пользователя всего функционала прикладного решения. Панель разделов соответствует уровню подсистем конфигурации.

Разделы в интерфейсе пользователя представлены в виде гиперссылок. При активизации раздела вся функциональность соответствующей подсистемы, включая вложенные подсистемы, представляется пользователю в виде команд на панели навигации и панели действий.

Интерфейс разделов состоит из панели навигации и панели действий.

Команды панели навигации определяют навигацию, то есть пользователь перемещается по конфигурации.

Панель действий определяет конкретное действие, например, создать документ или сформировать отчет.

На панели навигации и на панели действий выделяют группы.

Для панели навигации — это группа *Важное*, группа *Обычное*, группа *См. также*, которые просто влияют на отображение команд на панели.

Для панели действий — это группа *Создать*, группа *Отчеты*, группа *Сервис*, причем каждая в своем составе имеет команды по назначению.

Для того чтобы настроить командный интерфейс, для каждой подсистемы необходимо в контекстном меню выделенной подсистемы выполнить команду Открыть командный интерфейс.

Все команды в «1С:Предприятие» можно классифицировать следующим образом: системные, стандартные и произвольные.

Состав системных команд и выполняемые действия предопределены на уровне платформы.

Состав стандартных команд зависит от объектов конфигурации, от их свойств, то есть они предопределены на уровне объекта конфигурации.

Выполняемые действия тоже считаются предопределенными, но всегда есть возможность повлиять на них, используя механизм событий.

Создаются разработчиком только произвольные команды. Для этого в конфигурации есть объект *Команда*.

В рамках прикладного решения для реализации используются также общие команды. Общие команды необходимы для реализации специфического функционала для определенного объекта, причем здесь используются команды, подчиненные данному объекту.

По области видимости команды классифицируют на глобальные и локальные команды формы.

Глобальные команды в рамках прикладного решения формируют глобальный командный интерфейс и вводятся для выбора пользователем той или иной функциональности. К данной группе можно отнести: системные команды, стандартные команды объектов конфигурации, а также общие и подчиненные команды.

Локальные команды формы, а также часть глобальных команд формируют командный интерфейс форм и используются для выполнения действий в форме. К данной группе относят стандартные команды формы, расширений формы, расширений элементов формы *Таблица или произвольные команды формы*.

Если рассматривать прикладное решение в целом, то интерфейс приложения можно классифицировать следующим образом: обычный интерфейс и управляемый интерфейс.

К характеристикам обычного интерфейса можно отнести:

- наличие главного меню;
- главное меню неизменно для всех пользователей, независимо от их прав доступа и каких-либо настроек;
- возможность создавать разные интерфейсы для разных пользователей.

Обычный интерфейс предусматривает работу с обычными формами. Обычные формы работают только в толстом клиенте и не работают на каналах связи. Такие формы разрабатываются интерактивно, то есть разработчик вручную определяет ее состав из элементов управления (текстовые поля, кнопки, переключатели, флажки), самостоятельно размещает их на форме. Также необходимо помнить при работе с обычными формами: программный код всегда исполняется на клиенте.

В настоящее время в основном используется управляемый интерфейс, который состоит из команд и окон, является динамическим, то есть доступность тех или иных команд зависит от прав пользователей, настроек, выполненных в конфигурации.

При запуске прикладного приложения в управляемом режиме в пользовательском интерфейсе открывается рабочий стол, на котором пользователь видит актуальную информацию.

Рабочий стол имеет возможности настройки, не прибегая к средствам конфигурирования.

Главной особенностью управляемого интерфейса является возможность работы в веб-клиенте через веб-браузер, при этом на компьютер платформа «1С:Предприятие» не устанавливается.

Управляемая форма – новый объект платформы «1С:Предприятие» генерируется платформой автоматически.

Структура управляемой формы является четкой, так как разработчик не может своевольно изменять положение элементов управления на ней. Разработчик только описывает элементы формы и может изменять взаимное расположение элементов только согласно определенной структуре. Модуль управляемой формы может компилироваться как на клиенте, так и на сервере.

Форма помогает визуализировать информацию, которая хранится в базе данных в удобном для пользователя виде, а также описывает алгоритмы работы пользователя с этой информацией.

У каждого прототипа объекта может быть свой определенный набор форм. Этот набор форм привязан именно к прототипу объекта. От вида формы также зависит ее функциональность.

Любая форма для любого объекта может быть сконструирована в конфигураторе. Для ее создания в конфигурации присутствует объект конфигурации Форма, причем объект подчинен прикладному объекту, но в целом может быть и самостоятельным объектом.

Для представления прикладного объекта в разных ситуациях можно использовать пять форм, причем в разных местах конфигуратора одинаковые формы могут называться по-разному.

Классификация форм следующая:

– форма элемента используется для работы с одним элементом прикладного объекта;

- форма группы используется для работы с группой прикладного объекта;
- форма списка отображает список элементов прикладного объекта;
- форма выбора используется для выбора одного элемента из списка элементов прикладного объекта;
- форма выбора группы используется для выбора группы прикладного объекта.

В редакторе форм все элементы формы представляются в виде иерархической структуры, корнем которой является сама форма. Это позволяет быстро перемещаться к нужному элементу формы.

Элементы формы – это те объекты, посредством которых осуществляется взаимодействие с формой. Именно они отвечают за ввод данных, различные кнопки и прочее. Элементы формы существуют только на форме и доступны, только когда форма открыта. С элементами формы можно работать интерактивно и программно, то есть посредством кода можно обращаться к свойствам элемента формы и изменять их. Элементам можно задавать различные настройки: изменять их порядок, группировать, изменять расположение на форме.

Также у формы есть реквизиты. Реквизиты – это определенные объекты, которые создаются на форме и существуют только на определенной форме. Такие объекты существуют, когда форма открыта. Если форма закрыта, они перестают существовать. В основном эти объекты, их значения используются для вспомогательных вычислений, для определенных действий на форме. Также на форме имеется основной реквизит формы, который задает ее функциональность. В основном у каждой формы, например, формы элемента, основным реквизитом является реквизит *Объект*. Посредством этого основного реквизита можно обращаться ко всем реквизитам объекта. Этот реквизит заполняется во время создания формы на сервере, имеет определенный тип. Потом этот тип преобразуется в тип, пригодный для работы на клиенте (на тонком клиенте или веб-клиенте). Если рассматривать, например, форму списка, то на ней основным реквизитом будет являться *Список*.

Команды формы. Они позволяют выполнить определенные действия, когда данная форма открыта. Эти действия могут быть

как над элементами формы, реквизитами формы, так и над объектами, которые связаны и выведены на этой форме. Код команды прописан в модуле формы. Также у команды может существовать клиентский и серверный обработчик, то есть команда может выполнять действия как в клиентском контексте, так и в серверном контексте.

Параметры формы. Они необходимы для передачи данных с одной формы на другую. При создании параметров формы они доступны только в серверном контексте, кроме ключевых.

Добавлять элемент на форму можно с помощью команды <Добавить> в дереве формы или просто претащить определенный реквизит формы в дерево ее элементов.

Правило визуализации элемента на форме следующее: если элементу, например, кнопке, не назначен обработчик, то элемент не отображается. Если элементу формы не назначены данные (реквизиты), то элемент также не отображается. То есть элемент управления формы должен быть привязан либо к реквизиту формы, либо к команде.

Следовательно, необходимо придерживаться следующего алгоритма при добавлении объекта на форму: создать реквизит (команду), затем перетащить его в дерево элементов управления и создать элемент управления.

Построение модели взаимодействия пользователя с системой «1С:Предприятие» дает возможность обеспечить комфортную работу и безопасность доступа к данным. Для этого требуется дополнительная настройка прикладного решения как на уровне разработчика, так и на пользовательском уровне.

Также для обеспечения безопасности доступа к данным и удобства работы с ними в системе «1С:Предприятие» можно построить ролевую модель системы.

Роли в системе «1С:Предприятие» определяют полномочия пользователей для работы с данными информационной базы.

В конфигураторе можно создать нужное число ролей, с помощью которых разграничить права пользователей на доступ к данным. Роли могут быть заданы от возможности только просмотра ограниченного числа видов документов до полного набора прав по вводу, просмотру, коррективке и удалению любых видов данных.

Все права, которые заложены в системе «1С:Предприятие», можно разделить на две группы: группа основных прав и группа интерактивных прав.

Основные права дают пользователю право на выполнение действий с элементами данных системы или над всей системой в целом. Такие права всегда проверяются, независимо от способа обращения к объектам информационной базы.

Интерактивные права дают пользователю право на выполнение действий, которые могут быть выполнены интерактивно. Такие права проверяются только при выполнении интерактивных операций, например, редактирование формы.

Алгоритм предоставления доступа по каждому объекту и виду права доступа следующий:

- доступ открыт, если хотя бы в одной роли есть разрешение;
- доступ закрыт, если во всех ролях есть запрещение.

Если пользователь делает попытку выполнить неразрешенное действие, то система выдаст окно предупреждения.

Установленные права зависят друг от друга. Такую взаимосвязь система «1С:Предприятие» отслеживает между правами автоматически. Если разрешение на какое-либо право снимается, то система автоматически снимает разрешения на все права, зависящие от него. И наоборот, при установке какого-либо права система устанавливает все права, от которых это право зависит.

Контрольные вопросы

1. Что такое командный интерфейс в «1С:Предприятие»?
2. Какие объекты конфигурации используют при разработке пользовательского интерфейса?
3. Что такое подсистема в «1С:Предприятие»?
4. Что представляет собой панель навигации?
5. Что представляет собой панель действий?
6. В чем отличие между системными и стандартными командами?
7. Для чего предназначены общие команды?
8. Что такое управляемая форма?
9. Какие виды форм можно создать в «1С:Предприятие»?
10. Для чего предназначены роли в «1С:Предприятие»?

Рекомендуемая литература

1. Разработка управляемого интерфейса / В. А. Ажеронок, А. В. Островерх, М. Г. Радченко, Е. Ю. Хрусталева. — Москва : ИС-Паблишинг, 2010. — 723 с. — ISBN 978-5-9677-1148-0.
2. Заика, А. А. Разработка прикладных решений для платформы 1С:Предприятие 8.2 в режиме «Управляемое приложение» : учеб. пособие / А. А. Заика. — 3-е изд.(электрон.). — Москва : Интернет-Университет Информационных Технологий [и др.], 2021. — 238 с. — URL: www.iprbookshop.ru/73721.html (дата обращения: 05.11.2023). — Режим доступа: по подписке.

Тема 2.4. Типы данных в «1С:Предприятие»

Форма проведения занятия — лекция.

Вопросы для обсуждения

1. Основные типы данных.
2. Понятие типобразующего объекта.

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.4.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.4.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Типы данных в системе «1С:Предприятие» — одно из основных понятий, которое определяет информацию, содержащуюся в переменной или в любом объекте конфигурации.

Прикладные решения могут оперировать с достаточно большим набором типов данных, который позволяет правильно организовать работу по обработке данных, их представлению пользователю.

В системе «1С:Предприятие» выделяют группы данных: примитивные типы, типы-объекты и их подчиненные, прикладные типы, интерфейсные типы, типы общего назначения.

Примитивные типы данных входят в группу базовых типов, которые предоставляет система «1С:Предприятие». Такие типы данных используются и в других программных средах.

К примитивным типам данных в системе «1С:Предприятие» относятся:

- Число – десятичное число.
- Строка – строковое значение с установкой ее длины.
- Дата – значениями данного типа является дата и время.
- Булево – значениями данного типа является либо *Истина*,

либо *Ложь*.

– Тип – значения данного типа используются для идентификации типов значений, что применяется для определения и сравнения типов. Данный тип не имеет литералов и возвращается функциями встроеного языка *ТипЗнч* и *Тип*.

– Неопределенно – значение данного типа применяется, когда необходимо использовать пустое значение, не принадлежащее ни к одному другому типу.

– Null – значение данного типа используется исключительно для определения отсутствующего значения при работе с базой данных.

Следующая группа типов данных – типы-объекты. К этим типам данных относят универсальные коллекции значений.

«Универсальные коллекции значений – это объекты, предназначенные для хранения временной информации. Они не разрабатываются в конфигурации и не хранятся непосредственно в базе данных, а создаются только на этапе работы программы и уничтожаются при закрытии сеанса программы. Коллекции не являются объектами информационной базы и служат для вспомогательного сбора, группировки, анализа и обработки информации» [14].

К универсальным коллекциям значений относят массив, структуру, таблицу значений и другие.

Все универсальные коллекции значений создаются с помощью метода *Новый*. Их можно «обойти» как по индексу, так и с помощью цикла *Для Каждого Из*.

Массив – это объект, который представляет собой совокупность значений произвольного типа, включая и массивы. Это позволяет

создавать многомерные массивы, например, двумерные. Пример кода показан на рис. 1.

```
Массив = Новый Массив;  
Массив.Добавить ("Один");  
Массив.Добавить (2);  
Массив.Добавить ("три");  
Массив.Добавить (4);
```

Рис. 1. Фрагмент кода с использованием массива

Структура — это коллекция значений, позволяющая строить динамический набор данных, каждый элемент которой формируется по правилу <ключ — значение>. «Ключи структуры уникальны, и поэтому ими можно идентифицировать значения. Ключ структуры должен быть строковым и отвечать требованиям к именам переменных. К значениям структуры можно обратиться как к свойствам объекта, при этом ключ используется как имя свойства» [14]. Пример кода показан на рис. 2.

```
Структура = Новый Структура;  
Структура.Вставить ("Товар", Товар);  
Структура.Вставить ("Контрагент", Контрагент);
```

Рис. 2. Фрагмент кода с использованием структуры

Таблица значений — объект, который позволяет строить динамические наборы значений и манипулировать ими. Таблица значений может работать со значениями разных типов. Таблица значений может иметь любое количество колонок и быть связанной с элементом *Табличное поле* [14]. Пример кода показан на рис. 3.

```
ТабЗнач = Новый ТаблицаЗначений;  
ТабЗнач.Колонки.Добавить ("Количество", "Количество товара");  
СтрокаТаблицыЗначений=ТабЗнач.Добавить ();  
СтрокаТаблицыЗначений.Количество=100;
```

Рис. 3. Применение коллекции «Таблица значений»

Рассмотрим следующую группу типов данных — это прикладные типы.

Прикладные типы представляют собой типы данных, зависящие от метаданных, то есть появление этих типов данных в прикладном

решении связано с использованием прикладных объектов конфигурации. Различные объекты конфигурации определяют разное количество зависимых типов данных.

Введем понятие типобразующего объекта. В системе «1С:Предприятие» этот тип данных становится доступным, как только разработчик добавляет в конфигурацию новый прикладной объект.

Состав такого типа данных определяется принадлежностью объекта к тому или иному классу прикладных объектов.

Например, после создания нового справочника *Комплектующие* становятся доступны следующие типы данных:

- СправочникМенеджер. Комплектующие;
- СправочникСсылка. Комплектующие;
- СправочникОбъект. Комплектующие;
- СправочникВыборка. Комплектующие;
- СправочникСписок. Комплектующие.

Необходимо понимать, что для двух разных справочников (также и других объектов) такие типы данных будут различными. Например, в конфигурации существуют два справочника – *Комплектующие* и *Цены*. Несмотря на тот факт, что оба эти справочника унаследовали функциональность класса *Справочники* и для них в прикладном решении был создан один и тот же состав типов данных.

Например, СправочникОбъект.Комплектующие и СправочникОбъект.Цены – это разные типы данных.

Обратите внимание, что после создания нового регистра накопления, например, Заказы, состав новых типов данных изменится:

- РегистрНакопленияМенеджер. Заказы;
- РегистрНакопленияВыборка. Заказы;
- РегистрНакопленияСписок. Заказы;
- РегистрНакопленияНаборЗаписей. Заказы;
- РегистрНакопленияЗапись. Заказы;
- РегистрНакопленияКлючЗаписи. Заказы.

Прикладной тип данных изначально в системе «1С:Предприятие» не предустановлен и существует только в конкретном прикладном решении.

Следующая группа типов данных – интерфейсные типы.

Интерфейсные типы позволяют организовывать визуальное взаимодействие прикладного решения с пользователем. В основном это типы, связанные с работой форм и их элементов (полей, надписей, табличных данных и др.).

Следующая группа типов данных — типы общего назначения. К ним относятся типы данных, реализующие определенную функциональность прикладного решения: *Текстовый Документ*, *Табличный Документ*, *Построитель Запроса* и другие.

В системе «1С:Предприятие» (версия 8.3) в ветке Общие дерева конфигурации присутствует объект *Определяемые типы*. Этот объект позволяет создать новый тип данных на базе уже существующих в системе, например, примитивных или более сложных типов, например, ссылочных.

Новый объект *Определяемые типы* добавляется в конфигурацию, выполнив в контекстном меню команду **Добавить**. Набор типов данных для объекта *Определяемые типы* задается в палитре свойств в окне редактирования типов данных.

Определяемые типы обычно применяются, когда в конфигурации часто используется примитивный тип с одинаковыми квалификаторами, например, тип Число с длиной 15, точностью 3. Если разные объекты имеют реквизиты с таким типом, то таким реквизитам можно присвоить новый определяемый тип. При необходимости можно быстро изменить его настройки.

В системе «1С:Предприятие» существует понятие *Составной тип данных*. Тип данных считается составным, если при выборе типа объекта метаданных (реквизита, измерения, ресурса) в окне выбора типа данных выбрано более одного типа данных или выбраны ссылки на объекты разного вида.

Составной тип данных можно применять при решении различных задач, например, при работе с объектом *План видов характеристик*.

Следует отметить, что определяемый тип данных может также быть и составным. Для этого в палитре свойств *Определяемого типа* в окне выбора типа установить флажок *Составной тип*.

Контрольные вопросы

1. Для каких целей определяются типы данных в системе?
2. Что представляют собой примитивные типы данных?
3. Что представляет собой тип Булево?
4. Что представляет собой тип Неопределенно?
5. Как можно охарактеризовать типы-объекты?
6. Что такое универсальные коллекции значений?
7. Как можно охарактеризовать типобразующие объекты в «1С:Предприятие»?
8. Что представляют собой интерфейсные типы в «1С:Предприятие»?
9. Что представляют собой типы общего назначения в «1С:Предприятие»?
10. В чем отличия составного и определяемого типа данных?

Рекомендуемая литература

1. Скороход, С. В. Программирование на платформе 1С: Предприятие 8.3 : учеб. пособие / С. В. Скороход. — Ростов-на-Дону [и др.] : Издательство Южного федерального университета, 2019. — 133, [1]с. — URL: znanium.com/catalog/document?id=357443 (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-9275-3315-2.
2. Основы конфигурирования в системе «1С:Предприятие 8.0»: учеб. пособие. — 3-е изд. (электрон.). — Москва : Интернет-Университет Информационных Технологий [и др.], 2021. — 222 с. — URL: www.iprbookshop.ru/102027.html (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-4497-0876-2.

Тема 2.5. Основы встроенного языка «1С:Предприятие»

Форма проведения занятия – лекция.

Вопросы для обсуждения

1. Модули платформы «1С:Предприятие».
2. Встроенный язык – основной механизм технологической платформы «1С:Предприятие».

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.5.
2. Обратит внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.5.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Программирование представляет собой процесс написания программы, то есть написание алгоритмов на некотором специализированном языке, который понятен вычислительной машине.

Встроенный язык является основным механизмом технологической платформы «1С:Предприятие», который позволяет описывать алгоритмы функционирования прикладного решения.

Основная часть разрабатываемого приложения в «1С:Предприятие» создается с помощью визуального конструирования, то есть через создание объектов конфигурации, задания их свойств, форм представления, установки взаимосвязей. Причем каждому объекту конфигурации присуще типовое поведение. Чтобы задать объекту приложения собственное поведение, отличное от типового, используется встроенный язык.

В системе «1С:Предприятие» для написания программ на встроенном языке необходимо использовать команды, конструкции языка, переменные.

Для написания текста кода используются различные модули платформы «1С:Предприятие». Всего существует восемь модулей: модуль приложения, модуль сеанса, модуль внешнего соединения, общий модуль, модуль менеджера объекта, модуль объекта, модуль формы, модуль команды.

Модуль приложения содержит все методы и события, связанные с началом и концом работы «1С:Предприятие».

Модуль сеанса необходим для установки параметра сеанса. Параметр сеанса – глобальные переменные, доступные в любом месте программы. Их можно установить в самом начале сеанса.

В *модуле внешнего соединения* располагаются все методы и события, связанные с началом и концом работы в режиме СОМ-соединения.

Общий модуль необходим для описания процедур и функций, которые будут доступны из любого места конфигууратора. Обычно такие процедуры и функции связаны между собой по определенной логике.

Модуль менеджера объекта хранит процедуры и функции, которые не связаны с экземпляром объекта, то есть если мы работаем с каким-то справочником, то в модуле менеджера объекта мы пишем процедуру в целом для работы с этим справочником. Обычно через них можно получать какие-то выборки и подобное.

Модуль объекта связан с экземпляром конкретного объекта. То есть когда мы создаем элемент справочника или документ, то в модуле объекта описываются процедуры и функции, которые непосредственно нужны для работы с данным конкретным экземпляром объекта.

Модуль формы содержит процедуры и функции, необходимые для работы с формой.

Модуль команды (команды бывают как общие, так и подчиненные какому-то объекту) необходим для написания действий, которые будут отрабатываться при выполнении этой команды.

Именно в модулях содержится описание процедур и функций, но в некоторых случаях можно обращаться к операторам напрямую, например, при работе с модулем приложения, модулем формы или модулем объекта.

Например, в модуле приложения можно просто написать сообщение, например, как показано на рис. 4.

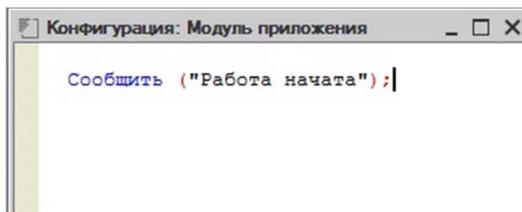


Рис. 4. Работа в модуле приложения

Программирование — это в первую очередь работа с данными.

Данные где-то нужно хранить до обработки, после обработки и главное — во время обработки. Для того чтобы данные можно было хранить во время обработки, были придуманы переменные.

Переменная — это временная сущность, необходимая для временного хранения информации.

Это может быть примитивная информация, например, число, строка, дата. Также это может быть более сложная информация, например, в переменной может храниться массив, таблица значений или ссылка на какой-либо справочник или документ.

Переменные задаются двумя способами:

- задание при помощи ключевого слова ПЕРЕМ;
- задание напрямую, путем обращения к переменной и присваивания ей значения.

Процедуры и функции — это основные операторы, при помощи которых осуществляется программирование в «1С:Предприятие».

Процедура — это фрагмент кода с определенным набором действий.

Функция — это тоже фрагмент кода с определенным набором действий, но функция возвращает определенное значение.

Для создания процедуры используется ключевое слово *Процедура* и указывается имя процедуры, например, как показано на рис. 5.

```
Процедура ИмяПроцедуры()  
    [Операторы]  
КонiecПроцедуры
```

Рис. 5. Конструкция процедуры

Название процедуры должно быть уникальным в рамках модуля.

Для создания функции используется ключевое слово *Функция* и указывается имя функции. Функция всегда возвращает определенное значение, для этого вводится ключевое слово *Возврат*, например, как показано на рис. 6.

```
□ функция ИмяФункции()  
  [Операторы]  
  Возврат ИмяПеременной;  
  КонецФункции
```

Рис. 6. Конструкция функции

Название функции также должно быть уникальным.

Во встроенном языке «1С:Предприятие» существуют и другие конструкции, которые часто используются в программном коде при решении задач.

Основным оператором условий во встроенном языке «1С:Предприятие», посредством которого осуществляется обработка условий, является конструкция *Если...Тогда*.

Конструкция *Если...Тогда* имеет следующий синтаксис, как показано на рис. 7.

```
Если <ЛогическоеУсловие> Тогда  
  [Операторы]  
КонецЕсли
```

Рис. 7. Конструкция *Если...Тогда*

Логическое условие представляет собой выражение, которое должно вернуть либо *Истину*, либо *Ложь*.

Если логическое условие принимает значение *Истина*, то выполняются операторы после ключевого слова *Тогда*. Если же это условие не выполняется, то выполняются операторы после ключевого слова *КонецЕсли*.

Представленную конструкцию можно расширить, добавив ключевое слово *Иначе*, как показано на рис. 8.

```
Если <ЛогическоеУсловие> Тогда
    [Операторы1]
Иначе
    [Операторы2]
КонецЕсли
```

Рис. 8. Конструкция *Если...Тогда...Иначе*

В этом случае если логическое условие принимает значение *Истина*, то выполнится первая группа операторов, если же это условие не выполняется, то выполнится вторая группа операторов.

Могут возникать еще более сложные ситуации, и нужно отработать три условия. В таких случаях конструкция еще более усложняется, как показано на рис. 9.

```
Если <ЛогическоеУсловие1> Тогда
    [Операторы1]
ИначеЕсли <ЛогическоеУсловие2> Тогда
    [Операторы2]
...
ИначеЕсли <ЛогическоеУсловиеN> Тогда
    [ОператорыN]
Иначе
    [ОператорыИначе]
КонецЕсли
```

Рис. 9. Отработка трех и более условий

В этом случае используется ключевое слово *ИначеЕсли*. Логическое условие, которое установлено после ключевого слова *ИначеЕсли*, будет проверяться тогда, когда условия после ключевого слова *Если* и после предыдущих ключевых слов *ИначеЕсли* (если они присутствуют) не выполняются и возвращают *Ложь*.

Причем в этой конструкции ключевое условие *Иначе* не обязательно к использованию.

Циклы необходимы для многократного перебора команд, инструкций кода.

Во встроенном языке «1С:Предприятие» используются три вида циклов: *Для Цикл*, *Пока Цикл* и *Для каждого Цикл*.

Циклы *Для Цикл* и *Пока Цикл* используются для циклического повторения произвольных операторов. Цикл *Для каждого Цикл* применяется в ситуациях, когда требуется выполнить обход коллекций значений (массивов, таблиц значений).

Конструкция *Для Цикл*, как показано на рис. 10, достаточно проста в исполнении.

```
Для ИмяПерем = Начало по Конец Цикл  
  <Операторы>  
КонецЦикла;
```

Рис. 10. Конструкция *Для Цикл*

В конструкции видно, что в самом начале переменной *ИмяПерем* присваивается выражение *Начало*, которое имеет числовое значение. В каждой итерации цикла переменная *ИмяПерем* увеличивается на 1. Цикл будет выполняться, пока переменная *ИмяПерем* меньше или равно переменной *Конец*. Причем *Начало* и *Конец* могут быть как числами, так и переменными с типом число.

Такой цикл позволяет выполнять линейные обходы.

Конструкция *Пока Цикл*, как показано на рис. 11, позволяет выполнять любые обходы.

```
Пока ЗначениеВыражения Цикл  
  <Операторы>;  
КонецЦикла;
```

Рис. 11. Конструкция *Пока Цикл*

ЗначениеВыражения представляет собой некоторое логическое выражение. Пока это выражение *Истина*, то операторы цикла выполняются. Если *ЗначениеВыражения* принимает *Ложь*, то цикл прекращается.

Конструкция *Для каждого Цикл* (рис. 12) используется для выполнения обхода различных коллекций, например, таблиц значений, массивов, табличных частей объектов.

```
Для Каждого ЭлементКоллекции Из ИмяКоллекции Цикл  
  <Операторы>  
КонецЦикла
```

Рис. 12. Конструкция *Для каждого Цикл*

Переменной *ЭлементКоллекции* присваивается очередное значение элемента коллекции. Обход будет продолжаться до тех пор, пока все элементы коллекции не будут перебраны.

1С – это событийное программирование, то есть это означает, что постоянно происходят те или иные события, например, открыл пользователь систему «1С:Предприятие», открыл документ или закрыл форму документа. С каждым этим действием связано событие, и с каждым этим событием связан определенный фрагмент кода, который будет выполняться.

Процедуры и функции являются неотъемлемой частью механизма событий. Когда происходит какое-то событие в «1С:Предприятие», например, запускается система или закрывается система, открываются, создаются или записываются документы, система генерирует событие. К этим событиям привязаны процедуры, то есть событие произошло – система тут же вызывает связанную с ним процедуру. А в эту процедуру можно включить код, который необходимо использовать; например, при проведении документов нужно создать записи в регистрах.

Рассмотрим механизм обработки событий.

Платформа «1С:Предприятие» – объектно-ориентированная платформа.

Документы, справочники – это объекты, массивы – это объекты, константы – это объекты. В системе «1С:Предприятие» все является объектами, кроме переменных.

Объект представляет собой совокупность данных и методов (процедуры и функции) для работы с этими данными.

В «1С:Предприятие» используется понятие Обработчик события.

Обработчик события – это процедура, которая реагирует на определенное событие.

Обработчик события можно создать в любом модуле, кроме общего.

Примерами обработчиков событий в системе «1С:Предприятие» могут быть *ПриНачалеРаботыСистемы()*, *ПриОткрытии()*, *ПередЗаписью* и многие другие. Для каждого модуля набор обработчиков событий изменяется. Чтобы создать обработчик события, необходимо выполнить в меню *Текст* команду *Процедуры и функции*.

Контрольные вопросы

1. Каково назначение встроенного языка «1С:Предприятие»?
2. Что такое модуль «1С:Предприятие»?
3. Какие модули существуют в системе «1С:Предприятие»?
4. Для чего предназначен модуль объекта?
5. Для чего предназначен общий модуль?
6. Каково предназначение модуля форм в системе «1С:Предприятие»?
7. Что такое процедура в «1С:Предприятие»?
8. Что такое метод в «1С:Предприятие»?
9. Какие виды циклов используются в «1С:Предприятие»?
10. Что такое обработчик события в «1С:Предприятие»?

Рекомендуемая литература

1. Скороход, С. В. Программирование на платформе 1С: Предприятие 8.3 : учеб. пособие / С. В. Скороход. — Ростов-на-Дону [и др.] : Издательство Южного федерального университета, 2019. — 133, [1]с. — URL: znanium.com/catalog/document?id=357443 (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-9275-3315-2.

Тема 2.6. Виды клиентов в «1С:Предприятие»

Форма проведения занятия — лекция.

Вопросы для обсуждения

1. Архитектура «1С:Предприятие».
2. Особенности видов клиентов в «1С:Предприятие».
3. Директивы компиляции в «1С:Предприятие».

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.6.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.6.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

В системе «1С:Предприятие» существуют два варианта хранения данных: файловый вариант и клиент-серверный вариант.

Файловый вариант — это значит, что вся информация хранится в одном файле. И «1С:Предприятие» работает с этим файлом, извлекает информацию из этого файла и отображает ее в клиентском приложении, наоборот, оно записывает информацию в этот файл. Следовательно, осуществляется двусторонний обмен. В этом файле хранится вся структура метаданных, а также те данные, которые ввели пользователи при работе с этой программой.

Такой вариант хранения данных применим для работы небольших систем, где немного пользователей и рабочих мест.

В других случаях используется клиент-серверный вариант хранения данных.

В клиент-серверном варианте хранения данных (или его еще называют клиент-серверная архитектура) существуют три звена.

Первое звено — локальная машина, то есть та машина, где работает пользователь.

Второе звено — кластер серверов «1С:Предприятие», то есть это отдельная машина, на которой установлено и работает специальное программное обеспечение или сервер «1С:Предприятие».

Третье звено — система управления базой данных (СУБД), которая представляет собой отдельную машину (хотя может находиться вместе с кластером серверов «1С:Предприятие»), на которую установлена база данных и хранится вся информация, то есть в кластере серверов ничего не хранится, а все хранится в СУБД.

Причем если кластер серверов — программное обеспечение фирмы «1С», то СУБД, как правило, стороннее программное обеспечение.

И файловый вариант хранения данных, и клиент-серверный вариант хранения данных работали и под обычным приложением, которое использовалось в более ранних версиях «1С:Предприятие».

С приходом управляемого приложения добавились виды клиентских приложений.

Ранее на платформах 8.0 и 8.1 существовало всего два вида клиента или два вида клиентских приложений, то есть приложе-

ний, под которыми работал клиент. Это конфигуратор и обычное приложение (толстый клиент). Связь между кластером серверов «1С:Предприятие» и клиентскими машинами осуществлялась только по локальной сети. Это была работа под обычным приложением. С приходом платформы 8.2 появилась возможность работы через Интернет, то есть связь между кластерами серверов «1С:Предприятие» и локальными машинами смогла осуществляться через сеть Интернет. Это стало возможным благодаря управляемому приложению, что не было возможным под обычным приложением.

Управляемое приложение существенно изменило правила программирования в «1С:Предприятие». Если в обычном приложении пользователю было неважно, где выполняется код — на локальной машине или на серверной, то под управляемым приложением это стало важным. То есть под управляемым приложением стало критично, как осуществляется связь между клиентской локальной машиной и кластером серверов «1С:Предприятие». Для локальных машин появились разные виды клиентских приложений.

Клиентское приложение (или клиент) — это программа, работающая на устройстве пользователя, которая обеспечивает интерактивное взаимодействие с «1С:Предприятие».

На платформе 8.2, а потом с развитием и на платформе 8.3 появились следующие виды клиентов: толстый клиент, тонкий клиент, веб-клиент, мобильный клиент, конфигуратор.

Толстый клиент — это клиентское приложение, которое устанавливается на компьютер и которое осуществляет связь с кластером серверов «1С:Предприятие» только посредством локальной сети.

Тонкий клиент — это клиентское приложение, которое устанавливается на компьютер, но посредством этого программного обеспечения связь локальной машины с кластером серверов «1С:Предприятие» может осуществляться как по локальной сети, так и через Интернет.

Веб-клиент — это клиентское приложение, которое не устанавливается на компьютер пользователя и вся работа с базой данных осуществляется через веб-браузер, то есть ничего отдельно устанавливать не надо.

Мобильный клиент — это клиентское приложение (данный вид появился в платформе 8.3), которое устанавливает связь с кластером серверов «1С:Предприятие» посредством сети Интернет.

Конфигуратор — это та среда, где ведется непосредственно разработка, в этом случае связь с кластером серверов «1С:Предприятие» осуществляется по локальной сети.

Также отдельным видом приложения, которое устанавливается, как правило, на отдельный компьютер, является *сервер*. Сервер не является клиентским приложением, то есть пользователь с ним напрямую работать не может.

Поскольку работа в управляемом приложении эмулируется под трехзвенную архитектуру, необходимо иметь представление обо всех ее составляющих и понимать их взаимодействие.

Так как в управляемом приложении возможна связь через Интернет, то стало критично, где выполняется код. Мы не можем через Интернет передавать большие объемы данных, все расчеты и вычисления должны выполняться на стороне кластера серверов «1С:Предприятие», а на стороне клиентского приложения должны прорисовываться форма и выполняться различные обработчики на форме, например, нажатие на кнопку. А действия достаточно сложные, например, проведение документа или запрос должны выполняться на стороне сервера, на стороне кластера серверов «1С:Предприятие».

Поскольку в управляемом приложении стало важно, где выполняется код — на клиентской машине или на сервере, появляются два контекста выполнения кода — клиентский контекст и серверный контекст.

Клиентский контекст — код выполняется на локальной машине клиента. *Серверный контекст* — код выполняется на машине кластера серверов «1С:Предприятие».

Также выполнение контекста кода связано с видами клиентов.

Код, который работает в толстом клиенте, тонком клиенте, веб-клиенте, мобильном клиенте, — это код, который работает в клиентском контексте. И, соответственно, код, который работает на сервере, — это код, который работает в серверном контексте.

Рассмотрим директивы компиляции.

При помощи директив компиляции можно показывать, где выполняется код в серверном контексте или клиентском контексте. Каждая процедура или функция в модулях форм или в модулях команд должна прописываться с указанием директивы компиляции. В общих модулях тоже можно указывать директивы компиляции.

Всего директив компиляции пять: *&НаКлиенте*, *&НаСервере*, *&НаСервереБезКонтекста*, *&НаКлиентеНаСервереБезКонтекста*, *&НаКлиентеНаСервере*.

Если код метода предварен директивой *&НаКлиенте*, то код выполняется в клиентском контексте, то есть на компьютере клиента.

Если код метода предварен директивой *&НаСервере*, то код выполняется в серверном контексте, то есть на компьютере сервера, где установлен кластер серверов «1С:Предприятие». Но если эта директива предваряет метод в модуле формы, то в методе будет доступен контекст формы.

В случае применения директивы *&НаСервереБезКонтекста* код выполняется тоже в серверном контексте на компьютере кластера серверов «1С:Предприятие», но если эта директива предваряет моды в модуле формы, то в этом методе не будет доступен контекст формы.

Когда в клиентском контексте нельзя выполнить определенные методы, например, создать объект или выполнить запрос, то эту задачу надо передать на сервер и сделать это в серверном контексте. Эта передача с клиентской машины на сервер называется — вызов сервера.

Рассмотрим на примере работу с контекстом в общем модуле.

В палитре свойств общего модуля есть настройки работы с клиентом, сервером, внешним соединением и др. Это говорит о том, что общий модуль поддерживает методы работы с клиентским контекстом, а также серверным контекстом.

Если необходимо использовать общий модуль в серверном контексте, то обязательно устанавливаем флажок *Сервер* и флажок *Серверный вызов*.

Если необходимо использовать общий модуль в клиентском контексте, то обязательно устанавливаем флажок *Клиент*, дополнительных указателей ставить не нужно.

Контекст формы. Если пользователь создает управляемую форму с помощью доступного ему способа — стандартного с помощью механизмов платформы и с помощью команд, то форма сначала создается в серверном контексте, то есть на стороне сервера. И только потом она передается в клиентский контекст. Другими словами, управляемая форма одновременно существует и на сервере, и на клиенте. Следовательно, при программировании на форме, используя соответствующие директивы, необходимо определять, какой метод выполняется на клиенте, а какой на сервере. Так же, как в случае общих модулей, можно из клиентского метода вызвать серверный метод. Но есть особенность, о которой надо помнить: контекстный вызов и внеконтекстный вызов.

В случае контекстного вызова данные управляемой формы передаются на сервер, то есть форма на сервере обновляется.

В случае внеконтекстного вызова данные формы не передаются на сервер, и поэтому в серверном контексте нет данных этой управляемой формы, из которой произошел внеконтекстный вызов.

Контекстный вызов осуществляет тогда, когда процедура или функция выполняется под директивой *&НаСервере*. Внеконтекстный вызов будет выполняться тогда, когда процедура или функция выполняется под директивой *&НаСервереБезКонтекста*. Получается, внеконтекстный вызов более быстрый, чем контекстный, так как данные не передаются на сервер, а происходит только его вызов и передача управления на сервер. В случае контекстного вызова данные формы передаются и канал связи будет более загруженным, но так как происходит кэширование данных, то есть не все данные передаются на сервер, а только измененные. Так как управлять кэшированием нет возможности, контекстный вызов все равно проигрывает внеконтекстному вызову. Но в процедуре, которая выполняется под директивой *&НаСервереБезКонтекста*, данные формы не будут доступны, то есть не будет возможности обратиться ни к реквизитам, ни к параметрам и прочему.

Директива *&НаКлиентеНаСервереБезКонтекста* используется только в модулях форм и применяется, если обращение к процедуре происходит как с клиента, так и с сервера. В остальном эта директива похожа на директиву *&НаСервереБезКонтекста*.

Директива *&НаКлиентеНаСервере* используется только в модуле команды. Процедура или функция, которые работают под такой директивой, могут быть использованы как на клиентской, так и на серверной стороне в модуле команды.

Контрольные вопросы

1. Какие варианты хранения данных поддерживает «1С:Предприятие»?
2. Какими характеристиками обладает файловый вариант хранения данных «1С:Предприятие»?
3. Как можно охарактеризовать клиент-серверную архитектуру «1С:Предприятие»?
4. Что понимают под кластером серверов «1С:Предприятие»?
5. Какие виды клиентских приложений используются в «1С:Предприятие»?
6. Что такое толстый клиент?
7. В чем различия между толстым и тонким клиентом?
8. Что такое серверный контекст?
9. Каковы характеристики директивы компиляции *&НаКлиенте*?
10. Каковы характеристики директивы компиляции *&НаСервере-БезКонтекста*?

Рекомендуемая литература

1. Радченко, М. Г. Архитектура и работа с данными «1С:Предприятия 8.2» / М. Г. Радченко, Е. Ю. Хрусталева. — Москва : 1С-Паблишинг, 2011. — 268 с. — (1С. Профессиональная разработка). — ISBN 978-5-9677-1555-6.

Тема 2.7. Запросы в «1С:Предприятие»

Форма проведения занятия – лекция.

Вопросы для обсуждения

1. Подходы при построении предметно-ориентированных информационных систем.
2. Основные черты предметно-ориентированных информационных систем.

При освоении темы необходимо:

1. Освоить учебный материал по теме 2.7.
2. Обратить внимание на основные понятия и определения.
3. Ответить на контрольные вопросы по теме 2.7.

Оборудование к занятию: проектор, ноутбук.

Теоретические сведения

Системная архитектура состоит из двух частей – архитектура данных и архитектура бизнес-приложений. Архитектура бизнес-приложений, по сути, представляет собой логику получения данных из базы данных или от пользователя, логику обработки данных, полученных от пользователя или уже имеющихся в СУБД, и логику визуализации данных и сохранения данных. Простейший инструмент визуализации – форма.

Другими словами, архитектура данных отвечает на вопросы: что за информация, какова ее структура, как эти данные обрабатываются.

В системе «1С:Предприятие» есть две модели обращения к данным – объектная (через объекты конфигурации) и табличная (использование запросов).

Основной способ получения данных – запрос.

Поскольку все объекты в «1С:Предприятие» представляют собой таблицы, любой справочник, документ, регистр, то так или иначе запрос делается всегда.

Необходимо помнить, что язык запросов манипулировать данными не может. Нельзя с помощью языка запросов осуществить изменение базы данных, можно только получить данные и их визу-

ализировать. Все изменения осуществляются с помощью объектной модели.

Язык запросов «1С:Предприятие» применяется в двух случаях: для вывода отчета через систему компоновки данных и для выборки данных из информационной базы, чтобы в дальнейшем произвести с ними манипуляции через объектную модель.

Текст запроса состоит из нескольких секций:

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

Из перечисленных секций обязательной является только описание запроса. Раздел *Описание запроса* содержит сведения о полях выбора, используемых таблицах и соединениях, условиях и группировках.

Секции состоят из предложений, которые, в свою очередь, содержат ключевые слова, например, ВЫБРАТЬ, ИЗ, ГДЕ, обозначающие определенное действие, которое нужно выполнить с базой данных.

Ключевое слово, с которого начинается предложение, обычно дает название предложению языка запросов.

Запрос формируется и выполняется разработчиком из встроенного языка.

Общая схема работы с запросом состоит из нескольких последовательных этапов:

- создание объекта *Запрос* и установка текста запроса;
- установка параметров запроса;
- выполнение запроса и получение результата;
- обход результата запроса и обработка полученных данных.

Для формирования и выполнения запросов в «1С:Предприятие» используется объект *Запрос*, который создается вызовом конструкции *Новый Запрос* (строка 1 на рис. 13).

```

1  Запрос = Новый Запрос;
   Запрос.Текст =
2
   "ВЫБРАТЬ Заказ.Регистратор
   |
   |ИЗ РегистрНакопления.Заказы КАК Заказ
   |ГДЕ Сумма<=МаксимальнаяСумма ";
3
   Запрос.УстановитьПараметр ("МаксимальнаяСумма", ЭтаФорма.МаксимальнаяСумма);
4
   Вывод = Запрос.Выполнить().Выбрать();

   Сообщение = Новый СообщениеПользователю;
5
   Пока Вывод.Следующий() Цикл
     Сообщение.Текст = Вывод.Регистратор;
     Сообщение.Сообщить();
   КонiecЦикла;

```

Рис. 13. Пример использования запроса

Текст запроса представляет собой инструкцию для выполнения запроса (строка 2 на рис. 13).

В тексте запроса прописаны источники данных запроса, какие поля необходимо получить, при необходимости установить режимы группировок, сортировок и прочее.

Инструкция составляется на встроенном языке и состоит из секций с определенным назначением, предложений, содержащих ключевые слова, например, ВЫБРАТЬ, ИЗ, ГДЕ, обозначающих определенное действие, выполняемое с базой данных.

Метод *Запрос.Выполнить* запускает запрос на выполнение, при необходимости данные можно ввести через параметр (строка 4 и строка 3 соответственно на рис. 13).

После выполнения запроса «1С:Предприятие» возвращает результат, который необходимо обработать.

Существует два способа обработки результата запроса: *выгрузить в таблицу значений* и *обойти по строчкам списка значений* (строка 5 на рис. 13).

Запросы в «1С:Предприятие» позволяют получать данные, поэтому любой запрос начинается с ключевого слова ВЫБРАТЬ. Далее после этой команды указываются наименования полей, из которых необходимо получить данные. Если указать «*», то будут выбраны все доступные поля. После ключевого слова ИЗ необходимо указать наименования документов, регистров, справочников и прочее, откуда будут выбираться данные. На слайде эта часть запроса выделена рамкой.

Инструкции расширяют запрос. Для ключевого слова ВЫБРАТЬ можно добавить одну или несколько инструкций:

РАЗЛИЧНЫЕ — позволяет отобразить только отличающиеся хотя бы по одному полю строки, без дублирования записей;

ПЕРВЫЕ n, где n — количество строк, которые необходимо отобразить с начала результата. Например, когда нужно отобразить определенное количество последних по дате документов. Чаще всего такая инструкция используется вместе с сортировкой **УПОРЯДОЧИТЬ ПО**;

РАЗРЕШЕННЫЕ — позволяет выбирать из базы данных только те записи, которые доступны пользователю. Если у пользователя нет доступа к данным, то он увидит сообщение об ошибке.

Эти ключевые слова могут использоваться как все вместе, так и по отдельности.

В реальных задачах запросы могут быть сложными, и источников в них может быть несколько. Поэтому, следуя правилам оформления запросов, можно вводить псевдонимы источников и затем использовать их в других частях запроса. Это становится обязательным, когда из разных таблиц выбираются поля с одинаковыми именами.

Псевдонимы задаются с помощью ключевого слова **КАК**, после которого следует имя псевдонима.

Конструкции формирования условий необходимы для наложения какого-либо отбора на выгружаемые данные.

В запросах существует два типа условий:

- условие накладывается на записи, выбираемые из базы данных с использованием ключевого слова **ГДЕ**;
- условие накладывается на выбранные и сгруппированные записи с использованием секции **ИМЕЮЩИЕ**.

При формировании условий отбора в предложениях **ГДЕ**, **ИМЕЮЩИЕ** применяются логические операторы **И**, **ИЛИ**, **НЕ**.

Логические операторы имеют приоритет:

- самый высокий приоритет имеет логический оператор **НЕ**;
- следующим по приоритету является оператор **И**;
- самый низкий приоритет у оператора **ИЛИ**;
- для того чтобы обеспечить другой порядок вычислений, можно использовать круглые скобки.

При выборе секции отбора необходимо учитывать следующее. Условия в секции ГДЕ отсеивают записи на этапе выборки из таблицы, условия в секции ИМЕЮЩИЕ действуют на уже выбранные записи.

Инструкция ИМЕЮЩИЕ обычно используется в предложении СГРУППИРОВАТЬ ПО.

Группировка в запросах позволяет сворачивать группировочные поля или строки таблицы по определенному признаку – группировочным полям.

Правила использования группировки следующие:

- все поля запроса должны делиться на группировочные и поля агрегатных функций;
- когда в предложении СГРУППИРОВАТЬ ПО содержится некоторое поле, по всем полям, полученным от него через точку, тоже выполняется группировка;
- при выполнении группировок количество итоговых записей в результате запроса будет меньше исходных данных или равно им;
- для отбора результатов, полученных после выполнения группировок, предназначена конструкция ИМЕЮЩИЕ, как показано на рис. 14.

```
ВЫБРАТЬ Заказ.Регистратор
|
| ИЗ РегистрНакопления.Заказы КАК Заказ
| СГРУППИРОВАТЬ ПО Заказ.Регистратор
| ИМЕЮЩИЕ
| СУММА (Сумма) <=&Сумма
```

Рис. 14. Текст запроса с использованием ИМЕЮЩИЕ

При формировании запросов можно использовать агрегатные функции.

СУММА или СРЕДНЕЕ – вычисляет соответственно арифметическую сумму или среднее арифметическое значение всех попавших в выборку значений поля. В качестве параметра используются числовые значения.

МИНИМУМ или МАКСИМУМ – вычисляет соответственно минимальное или максимальное значение из всех попавших в выборку значений поля. В качестве параметра используются значения любого типа.

КОЛИЧЕСТВО — подсчитывает количество значений параметра, попавших в выборку. В отличие от других агрегатных функций допускает три способа использования:

- подсчитать количество значений указанного поля, не равных NULL;
- подсчитать количество различных значений указанного поля, не равных NULL. Для этого перед спецификацией поля надо указать ключевое слово РАЗЛИЧНЫЕ;
- подсчитать количество строк в результате запроса. Для этого в качестве параметра функции нужно указать звездочку «*».

В качестве параметра функции можно указывать ссылки на поля, содержащие значения любого типа.

Инструкция ДЛЯ ИЗМЕНЕНИЯ предназначена для указания необходимости блокировки считываемых данных. Для файлового варианта блокируются указанные таблицы, а для клиент-серверного варианта — только выбранные записи.

«Контроль остатков» при проведении документа — нормальная операция для учетной системы. Однако у неопытного разработчика может возникнуть ситуация, когда система не сможет верно оценить количество товаров на складе, что приведет к ошибкам в учете.

Пример возможной ошибки списания в минус.

Рассмотрим хронологическую последовательность:

- на складе 10 карандашей;
- пользователь № 1 начинает проводить документ и спрашивает у системы: есть ли в наличии 8 карандашей? Система отвечает — да;
- пользователь № 2 начинает проводить документ и спрашивает у системы: есть ли в наличии 7 единиц товара? Система отвечает — да;
- так как пользователь № 1 получил одобрение от системы, документ списывает со склада 8 карандашей. На складе остаётся 2 штуки;
- пользователь № 2 тоже получил одобрение от системы, и документ, ничего не подозревая, списывает со склада еще 7 единиц;
- на складе возникает отрицательный остаток — минус 5.

Инструкция ДЛЯ ИЗМЕНЕНИЯ предотвращает отрицательные остатки.

Инструкция ДЛЯ ИЗМЕНЕНИЯ в тексте запроса «1С:Предприятие» устанавливает блокировку на читаемые данные до конца транзакции, например, обработку проведения документа. Следовательно, в рассмотренном примере пользователь № 2 встанет в очередь, пока пользователь № 1 не закончит операцию с ресурсом.

Инструкцию ДЛЯ ИЗМЕНЕНИЯ необходимо использовать для данных, которые планируется изменять в рамках текущей транзакции» [15].

В языке запросов можно объединять несколько запросов. При этом записи каждого запроса будут объединены в один результат запроса, причем полностью одинаковые строки в результате запроса из разных запросов заменяются одной. Чтобы эта установка не применялась, нужно использовать ключевое слово ВСЕ.

При объединении запросов обязательно должно выполняться правило – количество полей запросов должно быть одинаковым, также их порядок должен сохраняться.

Альтернативой объединению является соединение.

Соединения бывают нескольких видов. Ключевые слова ЛЕВОЕ, ПРАВОЕ и ПОЛНОЕ уточняют характер соединения. Слова ВНУТРЕННЕЕ или ВНЕШНЕЕ можно не указывать вообще, они повышают наглядность и читаемость текста запроса.

Соединения связывают две таблицы по определенному условию связи.

Суть ЛЕВОГО СОЕДИНЕНИЯ заключается в том, что полностью берется первая указанная таблица, называемая главной, и к ней по условию связи привязывается вторая. Если записей, соответствующих первой таблице, во второй не нашлось, то в качестве их значений подставляется NULL.

Например, ЛЕВОЕ СОЕДИНЕНИЕ работает примерно как цикл в цикле – берется первая запись из левой таблицы, и пробегают все записи из правой на предмет удовлетворения условию связи. Затем берется вторая запись из левой таблицы и так далее. Если вдруг условию связи удовлетворяют несколько записей из правой таблицы, то в результирующую таблицу будет добавлено несколько строк, по количеству удачных связей.

Суть ПРАВОГО СОЕДИНЕНИЯ – все с точностью до наоборот.

ПОЛНОЕ СОЕДИНЕНИЕ отличается от предыдущих тем, что в результате будут возвращены все записи — как первой таблицы, так и второй. Если по заданному условию связи в первой или второй таблице не найдено записей, вместо них будет возвращено значение NULL.

Отличием ВНУТРЕННЕГО СОЕДИНЕНИЯ от ПОЛНОГО является то, что если хотя бы в одной из таблиц не найдена запись, то запрос не выведет ее вообще.

В запросе всегда важно указывать, как упорядочить записи. Предложение УПОРЯДОЧИТЬ ПО позволяет определить условия упорядочивания строк в результате запроса, как показано на рис. 15.

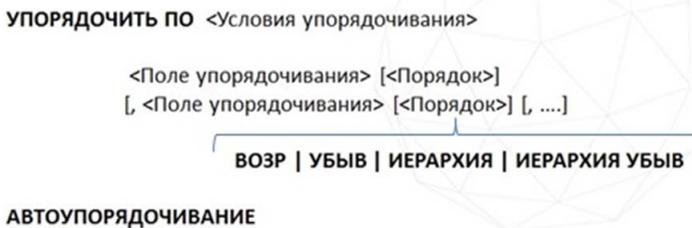


Рис. 15. Конструкция УПОРЯДОЧИТЬ ПО

В результате запроса предложение УПОРЯДОЧИТЬ ПО позволяет сортировать строки либо по убыванию, либо по возрастанию. Если метод сортировки не указан, то сортировка происходит по возрастанию.

Очень часто возникает ситуация, когда необходимо получать определенные итоги, используя данные базы данных.

Предложение ИТОГИ в запросах производят над данными выборки математические или статистические операции. К полученному результату выборки будет добавлена отдельная строка со значением итогов.

Секция ИТОГОВ в запросах начинается с ключевого слова ИТОГИ и содержит итоговые поля с применяемой к ним агрегатной функцией. Дополнительно после ключевого слова ПО нужно прописать группировочные поля, в разрезе которых будет подсчитываться результат по итоговым полям. Также при применении клю-

чего слова ОБЩИЕ можно подсчитать итоги по всей выборке, как показано на рис. 16.

```
ВЫБРАТЬ
| Заказ.Комплектующие, Заказ.ВнесеннаяСумма
| ИЗ Документ.Заказ.Оплата КАК Заказ
| ИТОГИ СУММА (ВнесеннаяСумма) ПО Общие
```

Рис. 16. Текст запроса с выводом итогов

Для формирования запросов любой сложности можно использовать конструктор запросов.

Конструктор запросов – механизм «1С:Предприятие», который представляет определенную форму с определенными вкладками, каждая из которых имеет конкретный функционал.

Стартовая вкладка *Таблицы и поля* отражает информацию о полях и таблицах в запросе. На этой вкладке закладывается основа секции ВЫБРАТЬ ИЗ. Ориентируясь на дерево конфигурации, можно определить источники информации для выборки и конкретные поля объектов.

Вкладка *Связи* содержит информацию о выбранных таблицах, а также поля, по которым они соединены. В тексте запроса информация из этой вкладки формируется в конструкцию СОЕДИНЕНИЕ. Нередко «1С:Предприятие» предлагает собственные варианты связей, основываясь на похожих типах данных, но разработчики предпочитают их изменять.

На вкладке *Группировка* строки группируются по определенным полям, а с числовыми реквизитами проводятся простые арифметические операции. В тексте запроса данные расположены в разделе СГРУППИРОВАТЬ ПО.

На вкладке *Условия* устанавливаются дополнительные условия на данные результата выборки. Эти условия в тексте запроса присутствуют после оператора ГДЕ.

Вкладка *Дополнительно* содержит несколько стандартных ограничений или возможностей выборки, которые в тексте определяются как: ПЕРВЫЕ N, РАЗЛИЧНЫЕ, РАЗРЕШЕННЫЕ. На вкладке присутствует возможность определения типа запроса – обычная выборка, формирование или ликвидация временной таблицы.

Возможность блокировки данных предназначена для включения автоматического режима блокировок в целях сохранения целостности информации на период от чтения до записи.

Вкладка *Объединения/Псевдонимы* содержит настройки изменения имен реквизитов или совмещения данных из группы запросов. Эти настройки находятся в виде текста по операторам КАК и ОБЪЕДИНИТЬ.

Все перечисленные вкладки конструктора запросов необходимы для правильной настройки запроса на выборку данных для решения любой практической задачи. Но в отдельных случаях пользователь вынужден обращаться и к дополнительным настройкам конструктора. Чаще всего из этого списка задействована опция формирования пакетных запросов.

Пакет запросов необходим в случаях, когда возникает необходимость получения нескольких выборок данных за одно обращение к базе или во время работы с временными таблицами, которые конструируются платформой в период выполнения запроса.

Пользователь может самостоятельно создавать временные таблицы с помощью конструктора на вкладке *Таблицы и поля* с помощью инструмента *Создать описание временной таблицы*. Это часто используется, если необходимо добавить в запрос таблицы извне. Здесь же можно описать вложенный запрос для дальнейшего его использования при обращении к базе данных.

У конструктора запроса есть еще одна опция, которая используется достаточно редко. Это вывод итогов. Вкладка *Итоги* позволяет пользователю выполнить необходимые арифметические расчеты с числовыми полями. В текстовом виде эта настройка отображается в виде оператора ИТОГИ ПО.

И, наконец, вкладка конструктора запроса *Построитель*. Здесь представлены настройки, которые позволят запросу быть выполненным интерактивно или без участия человека. Можно изменить текст запроса с учетом данных, которые были введены изначально. В текстовом виде выполненные настройки *Построителя* заключаются в фигурные скобки.

Чтобы контролировать процесс формирования запроса, его текстовый вид, в конструкторе можно активировать окно с текстом запроса к базе, щелкнув по кнопке <Запрос>. В этом окне можно отредактировать запрос, а конструктор в свою очередь добавит изменения во вкладки при закрытии.

Контрольные вопросы

1. Что такое логика приложения?
2. Как можно охарактеризовать объектную модель обращения к данным в «1С:Предприятие»?
3. Как можно охарактеризовать табличную модель обращения к данным в «1С:Предприятие»?
4. Что такое запрос?
5. Из каких секций состоит текст запроса?
6. Какова общая схема работы с запросом?
7. Для чего применяется метод УстановитьПараметр()?
8. Как применить группировку?
9. Какие агрегатные функции можно использовать в запросе?
10. Как объединить запросы?

Рекомендуемая литература

1. Хрусталева, Е. Ю. Язык запросов «1С:Предприятия 8» / Е. Ю. Хрусталева. – Москва : 1С-Публишинг, 2013. – 369 с. – ISBN 978-5-9677-1987-5.

ПРАКТИЧЕСКИЕ РАБОТЫ

Практическая работа 1 Создание информационной базы. Подсистемы. Справочники

Цель работы – формирование практических навыков создания информационной базы и основных объектов конфигурации «1С:Предприятие».

Общее задание (сквозная задача)

Условно основное направление деятельности предприятия – ремонт компьютерной техники. Для автоматизации некоторых участков работы сотрудников, таких как учет комплектующих на складе, выдача их сотруднику для ремонта, инвентаризация склада, работа с клиентами средствами технологий «1С:Предприятие», разработайте конфигурацию информационной системы «Учет компьютерных комплектующих».

Порядок выполнения работы

1. Создайте новую информационную базу.

Запустите систему «1С:Предприятие». Для создания новой информационной базы щелкните по кнопке <Добавить>. Далее необходимо выполнить последовательность действий с диалоговыми окнами: выберите *Создание новой информационной базы*, определите способ создания информационной базы, определите наименование информационной базы, выберите место хранения информационной базы. Щелкните по кнопке <Готово>. В системе «1С:Предприятие» есть два режима работы с информационной базой – режим *Конфигуратор* и режим *Предприятие*. Режим *Конфигуратор* предназначен для разработки конфигурации, режим *Предприятие* – пользовательский режим.

2. Откройте информационную базу в режиме *Конфигуратор*. Для этого выберите наименование информационной базы в списке и щелкните по кнопке <Конфигуратор>.

3. Создайте объекты *Подсистемы* с именами *Справочники* и *Предприятие*.

Подсистемы как общие объекты конфигурации позволяют определить интерфейс пользователя. В дальнейшем, по мере выполнения конфигурации объектами, все объекты (справочники, документы, отчеты и пр.) будут распределены по подсистемам для удобства работы с ними в пользовательском режиме *Предприятие*. Для того чтобы добавить подсистему, выделите объект Подсистема в ветке *Общие* и выполните команду *Добавить* в контекстном меню. В диалоговом окне в поле *Имя* задайте наименование подсистемы.

4. Для каждой введенной подсистемы установите пиктограмму. Для этого в свойствах для каждой подсистемы задайте определенный рисунок.

5. Чтобы посмотреть полученный результат, запустите отладку, выполнив команду **Отладка – Начать отладку**. Система «1С:Предприятие» предложит обновить информационную базу и переключится в пользовательский режим.

6. Создайте справочник *Склад*, который должен быть доступен в подсистеме *Справочники*. В справочник *Склад* добавьте predetermined элемент *Основной склад*.

Справочники используются в ситуациях, когда необходимо исключить неоднозначно трактуемый ввод информации. Аналогом справочника является список данных, который может корректироваться и дополняться в процессе работы. Например, это могут быть списки номенклатуры, организаций или сотрудников. Система «1С:Предприятие» позволяет создавать любое количество справочников. При этом в конфигураторе формируется структура справочника, а данными он заполняется в режиме *Предприятие*. Справочники бывают нескольких видов – обычные, иерархические, справочники с табличной частью. Вид справочника определяется на стадии разработки объекта в режиме *Конфигуратор*. Для того чтобы добавить объект *Справочник*, необходимо в дереве конфигурации выделить объект *Справочники* и в контекстном меню выполнить команду **Добавить**. Далее необходимо выполнить последовательность действий с вкладками. На вкладке *Основные* задать имя справочника, на вкладке *Подсистемы* выбрать подсистемы, в которых будет участвовать данный справочник, на вкладке *Данные* сформировать структуру справочника, то есть задать реквизиты

справочника. Справочник *Склад* является простым справочником, который в своей структуре имеет только два стандартных поля – *Код* и *Наименование* (задаются по умолчанию), поэтому для него другие реквизиты не будут указаны. На вкладке *Прочее* задайте предопределенные значения справочника *Склад*. Для этого щелкните по кнопке <Предопределенные> и добавьте требуемые элементы.

7. Добавьте в конфигурацию справочник *ЕдиницыИзмерения*; данный справочник должен быть виден в подсистеме *Справочники*. В справочник добавьте одну предопределенную единицу измерения – штука. Для того чтобы посмотреть полученный результат, необходимо запустить отладку.

8. Добавьте справочник *Поставщики*. Справочник должен быть доступен в подсистеме *Справочники*.

9. Создайте иерархический справочник *Комплектующие*; справочник должен быть доступен в подсистеме *Справочники*.

В справочнике *Комплектующие* определите следующие реквизиты: *ЦенаПоставки* (тип Число), *ЦенаПродажи* (тип Число), *Поставщик* (тип СправочникСсылка.Поставщики), *ОснЕдиницаИзмерения* (тип СправочникСсылка.ЕдиницыИзмерения).

Справочники в системе «1С:Предприятие» могут быть иерархическими. Так, в случае наиболее часто используемой иерархии групп все элементы справочника делятся на два вида – простые элементы и группы. Другой вариант иерархии – иерархия элементов. В этом случае роль групп выполняют сами элементы. Для того чтобы вновь добавляемый справочник получил иерархический вид, необходимо на вкладке *Иерархия* поставить флажок *Иерархический справочник* и выбрать вид иерархии.

10. В режиме *Предприятие* создайте в справочнике *Комплектующие* группы *Компьютерные комплектующие*, *Сетевое оборудование*, *Информационные носители*. Заполните их элементами.

11. Добавьте обычный справочник *Должности*; справочник должен быть доступен в подсистеме *Предприятие*.

Добавьте справочник *Сотрудники* с табличной частью *Трудовая деятельность*; справочник должен быть доступен в подсистеме *Предприятие*.

Реквизиты справочника: *Фамилия* (тип Строка), *Имя* (тип Строка), *Отчество* (тип Строка), *ДатаРождения* (тип Дата);

Реквизиты табличной части *ПриемНаРаботу* (тип Дата), *ОкончаниеРаботы* (тип Дата), *Должность* (тип СправочникСсылка.Должности).

12. Для того чтобы посмотреть полученный результат, запустите отладку.

13. Заполните данными справочник *Сотрудники*, добавив 4–5 записей. Заполните справочник *Должности*, добавив 3–4 записи.

14. Выгрузите информационную базу, выполнив в конфигураторе команду **Администрирование – Выгрузить информационную базу**; имя файла должно быть следующее: Практическая работа 1.

Практическая работа 2

Управляемые формы в «1С:Предприятие»

Цель работы – формирование практических навыков работы с командным интерфейсом, а также разработки и редактирования форм объектов конфигурации «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» добавить форму констант. Форму элемента справочника *Комплектующие* и справочника *Сотрудники* отредактировать и привести к заданному виду.

Порядок выполнения работы

1. Добавьте в конфигурацию перечисление *Статус* со значениями: крупное предприятие, среднее предприятие, малое предприятие.

Перечисление как объект конфигурации предназначен для представления определенных статических списков. В дереве конфигурации выберите объект *Перечисление* и в контекстном меню выполните команду **Добавить**. На вкладке *Данные* задайте значения для перечисления.

2. Добавьте в конфигурацию четыре константы: *НаименованиеОрганизации* – ЧП «Ваша фамилия», константу *Юридический-*

Адрес – г. Тольятти, ул. Белорусская, 16В, константу *ДатаРегистрацииОрганизации* (тип Дата), константу *Статус* (тип Перечисление-Ссылка.Статус).

3. Создайте форму, на которой должны присутствовать поля для всех четырех констант конфигурации.

Система <1С:Предприятие> начиная с версии 8.2 поддерживает управляемый интерфейс, который работает на управляемых формах, но версия 8.3 поддерживает также и обычный интерфейс с обычными формами.

Управляемый интерфейс, или, как его еще называют, командный интерфейс, является динамическим. Он состоит из команд и окон, доступность которых зависит от прав пользователя, настроек конфигурации.

Создавая управляемую форму, разработчик программным образом описывает элементы, присутствующие на форме, может изменять их взаимное расположение согласно определенной структуре.

В дереве конфигурации в ветке любого объекта можно создать форму. Чтобы создать форму констант, существует два способа.

Первый. В дереве конфигурации выделить объект *Константы* и в контекстном меню выполнить команду **Создать форму констант**, как показано на рис. 17.

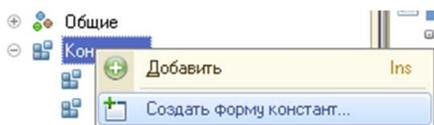


Рис. 17. Команда **Создать форму констант**

Второй. В дереве конфигурации в ветке *Общие* выделите *Общие формы* и выполните в контекстном меню команду **Добавить**. Далее необходимо выполнить последовательность действий с диалоговыми окнами конструктора форм: выбрать тип формы *Форма констант*, задать имя формы (оставьте имя, заданное по умолчанию ФормаКонстант), в качестве реквизитов формы установите флажок для каждой константы конфигурации и щелкните по кнопке <Готово>. В дереве конфигурации будет добавлен объект *ФормаКонстант*. При необходимости отредактируйте форму, для этого дважды щел-

кните по объекту *ФормаКонстант*. Откроется редактор форм, как показано на рис. 18.

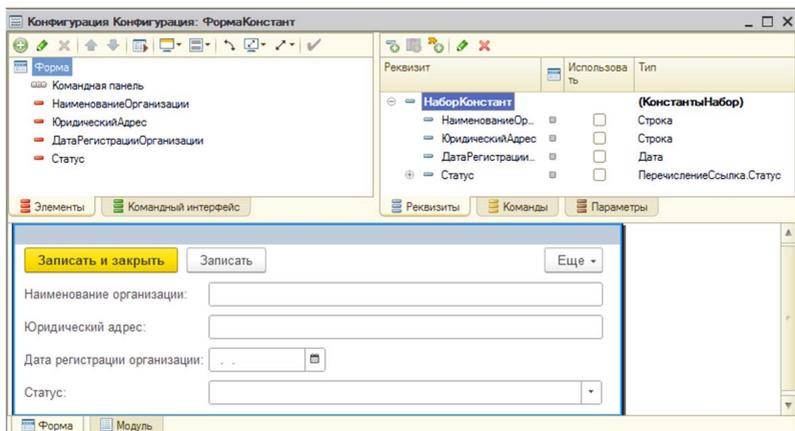


Рис. 18. Окно редактора форм

Неважно, каким способом была добавлена форма констант, она в любом случае будет находиться в ветке *Общие – Общие формы*.

4. Выведите константы в пользовательском режиме для просмотра.

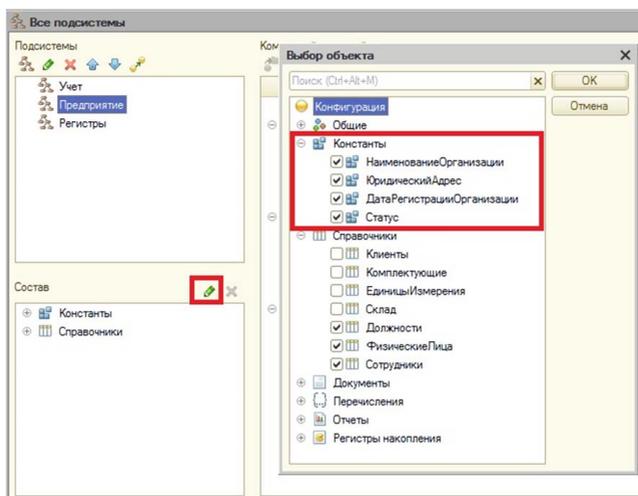


Рис. 19. Настройка командного интерфейса подсистем

Для этого выполните настройку командного интерфейса подсистем. Выделите в дереве конфигурации в ветке *Общие* объект *Подсистемы* и в контекстном меню выполните команду **Все подсистемы**. В диалоговом окне в списке подсистем выберите подсистему *Предприятие*, в разделе *Состав* щелкните по кнопке <Редактировать состав подсистемы>. В окне *Выбор объекта* установите флажки для всех констант, как показано на рис. 19. Подтвердите выбор, щелкнув по кнопке <ОК>. Константы появятся в разделе *Командный интерфейс* на *Панели действий.Сервис*.

5. Для справочника *Сотрудники* добавьте две формы – форму списка и форму элемента справочника. Для просмотра справочника на вкладке *Формы* в поле *ФормыСписка* установите значение *ФормаСписка*, а для отображения нового элемента установите *ФормаЭлемента*. Настройте *ФормуСписка* и замените наименование колонки *Наименование* на *ФИО*.

У каждого объекта конфигурации можно добавить форму. Формы могут быть разного предназначения. Например, если пользователь открывает справочник, то он видит все сохраненные элементы этого справочника. Такая форма представления записей называется формой списка. Когда пользователь добавляет новый элемент в справочник, то он работает с формой элемента справочника. Иерархические справочники поддерживают работу с группами, и в системе можно создать форму для группы. Также есть в системе форма выбора, форма выбора группы. В процессе разработки платформа по умолчанию создает для объектов стандартные формы. Но разработчик всегда может ее отредактировать.

Для того чтобы создать форму в дереве конфигурации, раскройте ветку нужного объекта, выберите *Формы* и в контекстном меню выполните команду **Добавить**. Далее в конструкторе форм определите тип новой формы, задайте ее имя, определите элементы формы.

Когда форма готова и необходимо отредактировать наименование колонки, то дважды щелкните по полю и в свойствах задайте новое наименование в поле *Заголовок*, как показано на рис. 20.

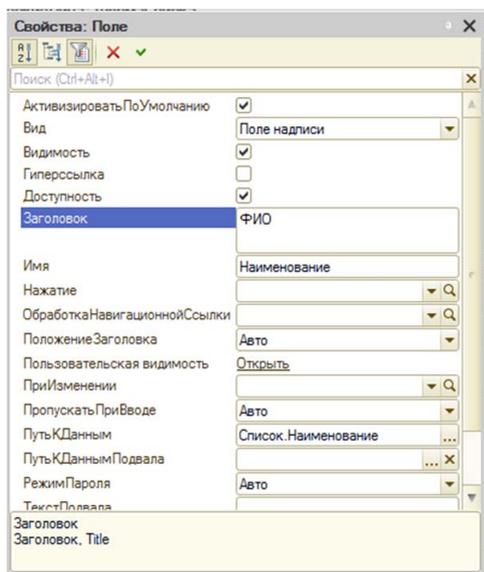


Рис. 20. Окно свойств поля

6. На *Форме элемента* справочника *Сотрудники* разместите реквизиты справочника и его табличную часть на разных страницах *Персональные данные* и *Трудовая деятельность* соответственно, как показано на рис. 21.

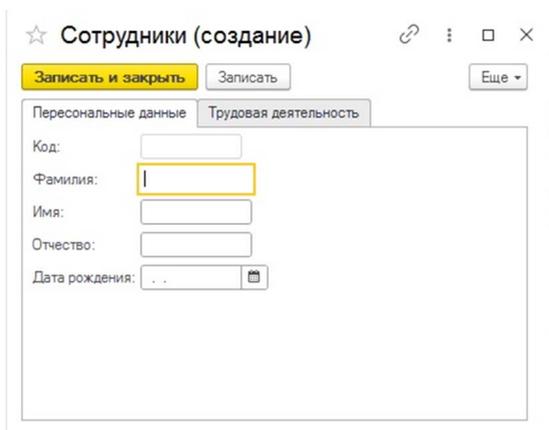


Рис. 21. Результатная форма справочника *Сотрудники*

Откройте редактор формы, щелкнув дважды по *Форме Элемента*. В верхней части редактора в виде дерева представлен состав элементов формы. Для того чтобы распределить текстовую и табличную части на две страницы формы, добавьте элемент *Группа-страницы*, как показано на рис. 22.

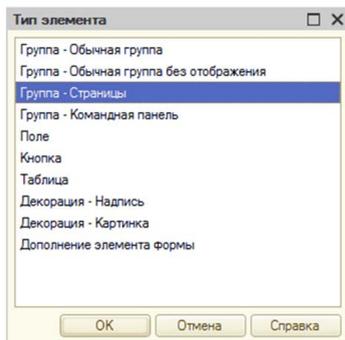


Рис. 22. Окно выбора типа элемента *Группа-Страницы*

Для каждой страницы в окне свойств задайте имя, настройте отображение страниц, как закладки сверху. Далее перетащите мышью элементы формы внутрь группы.

7. Настройте табличную часть так, чтобы реквизит табличной части *ПриемНаРаботу* и *ОкончаниеРаботы* находились в одной колонке.

Для того чтобы два поля объединить в одну колонку для табличной части, добавьте *Группу-Группа колонок*, как показано на рис. 23.

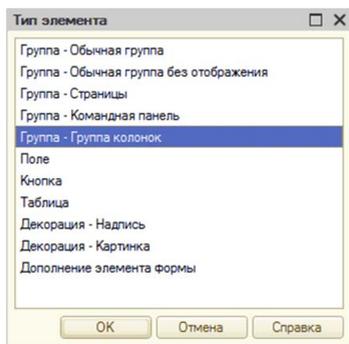


Рис. 23. Окно выбора типа элемента *Группа-группа колонок*

В окне свойств задайте имя группы, настройте вертикальную группировку. Далее перетащите мышью элементы формы в данную группу.

8. В справочнике *Комплектующие* добавьте *Форму элемента*. Измените расположение элементов на форме элемента справочника *Комплектующие* в соответствии с рис. 24.

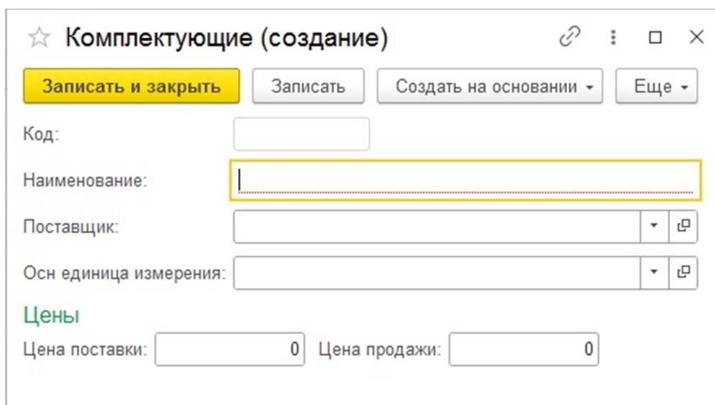


Рис. 24. Измененная форма элемента справочника *Номенклатура*

9. Для того чтобы посмотреть полученный результат, запустите отладку.

10. Выгрузите информационную базу, выполнив в конфигураторе команду **Администрирование – Выгрузить информационную базу**; имя файла должно быть следующее: Практическая работа 2.

Практическая работа 3 Документы в «1С:Предприятие»

Цель работы – формирование практических навыков работы с прикладным объектом *Документ* в «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» автоматизировать процесс поступления комплектующих на склад организации, выдачу их сотруднику, списание со склада. Все документы должны выводиться на печать.

Порядок выполнения работы

1. Добавьте документ *Поступление*; документ должен быть доступен в подсистеме *Документы*, которую при необходимости добавьте в конфигурацию.

Реквизиты документа *Поступление* следующие: *Бухгалтер* (тип СправочникСсылка.Сотрудники), *Поставщик* (тип СправочникСсылка.Поставщики), *Склад* (тип СправочникСсылка.Склад).

Табличная часть *Товары* документа *Поступление* имеет следующие реквизиты: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Количество* (тип Число), *ЦенаПоставки* (тип Число), *Сумма* (тип Число).

В целом документ относится к наиболее востребованным объектам конфигурации. Документ как объект конфигурации фиксирует происходящие в организации события, хозяйственные операции. Любой документ в системе обладает отличительными особенностями, такими, как номер документа, дата и время формирования документа. Объект *Документ* обладает также возможностью проведения, то есть сохранения всех данных документа в информационной базе.

Чтобы добавить объект *Документ* в дереве конфигурации, выберите ветку *Документы* и в контекстном меню выполните команду **Добавить**. Далее необходимо выполнить последовательность действий с диалоговыми окнами: на вкладке *Основные* введите наименование документа, на вкладке *Подсистемы* определите, где документ будет доступен пользователю, на вкладке *Данные* задайте реквизиты документа, здесь же при необходимости создайте табличную часть документа, расширив информацию об элементе дополнительными сведениями.

2. Предопределенный элемент *Основной* склад справочника *Склад* в свойствах установите как «Значение заполнения».

Каждодневная работа с документом подразумевает, что отдельные поля должны заполняться значениями автоматически. Чтобы использовать этот механизм и тем самым автоматизировать действия пользователя в «1С:Предприятие», существует несколько способов. Первый способ – настройка свойств объекта конфигурации. Для этого в свойствах реквизита *Склад* выберите значение *ОсновнойСклад* в поле *Значение заполнения*, как показано на рис. 25.

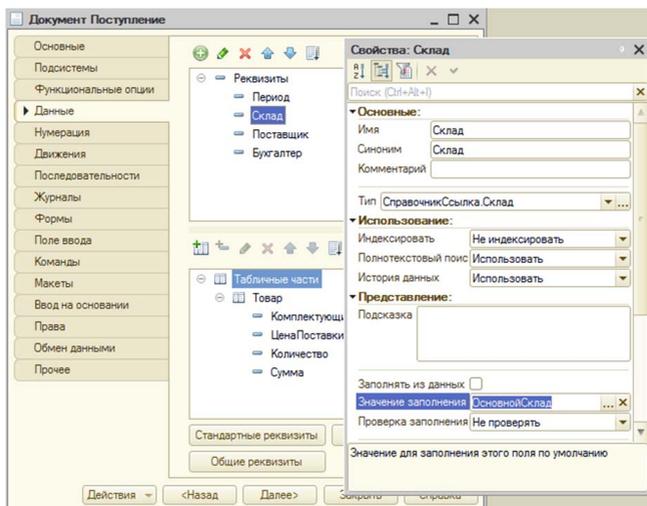


Рис. 25. Установка значения поля *Значение заполнения*

3. Создайте форму документа *Поступление*, назначьте ее основной.

У любого объекта конфигурации можно создать форму для того, чтобы визуально представить данный объект в интерфейсе пользователя. Объект *Документ* не исключение. Алгоритм добавления форм объектам в системе «1С:Предприятие» описан в практической работе 2.

4. Автоматизируйте расчет итоговой суммы поступившего товара (комплектующих для компьютера), используя формулу:

$$\text{Сумма} = \text{ЦенаПоступления} \times \text{Количество}. \quad (1)$$

В «1С:Предприятие» для автоматизации действий пользователя можно использовать разные механизмы системы. Одним из них является использование процедур и функций, описывающих поведение объекта, через его форму.

При заполнении документа *Поступление* для каждого товара требуется заполнить поле *Сумма*, выполняя расчет по указанной формуле. Сколько будет введено различных товаров, столько раз нужно произвести вычисления. Чтобы избежать многократного повторения расчета в системе «1С:Предприятие», можно создать процедуру, автоматизирующую данные вычисления.

В этом случае процедура пишется на встроенном языке «1С» в модуле формы. Чтобы открыть модуль формы, существует несколько способов. Первый способ. Откройте форму документа *Поступление* и перейдите на вкладку *Модуль*. Второй способ – дважды щелкните на элементе формы *Товары.Количество*, далее в окне свойств в разделе *События* выберите событие *ПриИзменении* и щелкните по кнопке с лупой, чтобы открыть модуль формы, в котором будет создана процедура обработчика события *ТоварыКоличествоПриИзменении()*. При создании обработчика события выберите вариант *&НаКлиенте*.

Аналогичный алгоритм действий повторите и для создания процедуры обработчика события *ТоварыЦенаПоставкиПриИзменении()* для поля формы *Товары.ЦенаПоставки*. Текст обеих процедур показан на рис. 26.

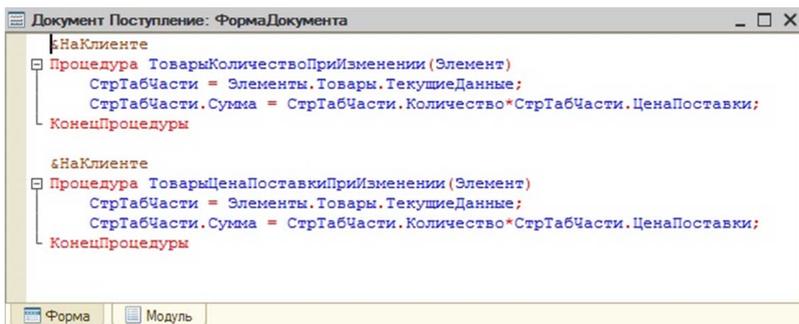


Рис. 26. Листинг процедур в модуле формы документа *Поступление*

5. Выполните отладку и проверьте работоспособность процедур, добавив несколько документов *Поступление* с разными данными.

6. Создайте печатную форму документа *Поступление* для вывода его на печать.

Очень часто при выполнении хозяйственных операций документы приходится выводить на печать. В рамках системы «1С:Предприятие» можно автоматизировать данный процесс и, используя конструктор печати, визуализировать печатную форму объекта конфигурации, например, справочника, документа, отчета.

Чтобы разработать печатную форму на вкладке *Макет* документа *Поступление*, запустите конструктор печати документов, щелкнув

по кнопке <Конструкторы>. Далее выполните последовательность действий с диалоговыми окнами: введите наименование команды *Печать*, определите поля текстовой и табличной частей, которые должны присутствовать на печатной форме документа, определите месторасположение команды в интерфейсе пользователя. Результат показан на рис. 27.

№	Комплектующие	Цена поставки	Количество	Сумма	
14	НомерСтроки	Комплектующие	ЦенаПоставки	Количество	Сумма
15					
16					
17					
18					

Рис. 27. Печатная форма документа *Поступление*

Обратите внимание, в дереве конфигурации в ветке документа *Поступление* добавились объекты — команда *Печать* и макет *Печать*. Система также автоматически сформировала в модуле команды *Печать* код процедуры *ОбработкаКоманды* и код процедуры *Печать*. Также в модуле менеджера документа *Поступление* конструктором была сформирована процедура *Печать*. Открыть модуль менеджера для просмотра кода можно на вкладке *Прочие* в окне настроек документа *Поступление*.

7. Аналогично добавьте в конфигурацию документ *ВыдачаСотруднику*; документ должен быть доступен в подсистеме *Документы*.

Реквизиты документа *ВыдачаСотруднику* следующие: *Бухгалтер* (тип *СправочникСсылка.Сотрудники*), *Комментарий* (тип *Строка*).

Табличная часть *Выдача* имеет следующие реквизиты: *Комплектующие* (тип *СправочникСсылка.Комплектующие*), *Количество* (тип *Число*), *Сотрудник* (тип *СправочникСсылка.Сотрудники*), *Склад* (тип *СправочникСсылка.Склад*).

8. Разработайте печатную форму документа *ВыдачаСотруднику*.

9. Добавьте в конфигурацию документ *ВозвратКомплектующих*; документ должен быть доступен в подсистеме *Документы*.

Реквизиты документа *ВозвратКомплектующих* следующие: *Сотрудник* (тип СправочникСсылка.Сотрудники), *Комментарий* (тип Строка).

Табличная часть *Возврат* имеет следующие реквизиты: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Количество* (тип Число).

10. Разработайте печатную форму документа *ВозвратКомплектующих*.

11. Добавьте в конфигурацию документ *СписаниеКомплектующих*; документ должен быть доступен в подсистеме *Документы*.

Реквизиты документа *СписаниеКомплектующих* следующие: *ПричинаСписания* (тип Строка), *Комментарий* (тип Строка).

Табличная часть *Списание* имеет следующие реквизиты: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Количество* (тип Число), *Склад* (тип СправочникСсылка.Склад).

12. Разработайте печатную форму документа.

13. Запустите отладку конфигурации и проверьте работоспособность процедуры *Печать* и добавьте в систему все документы.

14. Выгрузите информационную базу; имя файла должно быть следующее: Практическая работа 3.

Практическая работа 4

Механизм Ввод на основании

Цель работы – формирование практических навыков работы применения механизма ввода на основании в системе «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» автоматизировать процесс формирования документа *Поступление* на основании данных записи справочника *Комплектующие*, осуществить отбор по критериям.

Порядок выполнения работы

1. Настройте документ *Поступление* для ввода на основании элементов справочника *Комплектующие*.

Ввод нового документа является наиболее распространенной операцией в практической деятельности. Очень часто в хозяйственной деятельности необходимо создать документ, большая часть данных которого будет дублировать данные другого объекта конфигурации. В таких случаях система «1С:Предприятие» предоставляет механизм *Ввод на основании*, который позволяет создать документ и передать в него данные из другого документа (справочника или других объектов), взятого за основу.

Алгоритм обработки и передачи данных необходимо записать как обработчик события *ОбработкаЗаполнения*. Программный код в этом случае сохраняется в модуле объекта *Документ*. Код можно создать вручную или с помощью конструктора *Ввода на основании*, который с помощью визуальных средств позволяет составить текст обработчика события *ОбработкаЗаполнения*.

Чтобы начать работу с конструктором *Ввода на основании*, в настройках объекта документ *Поступление* на вкладке *Ввод на основании* установите объект, являющийся основанием, — справочник *Комплектующие*. Для этого щелкните по кнопке <Редактировать> и поставьте флажок около справочника *Комплектующие*, как показано на рис. 28.

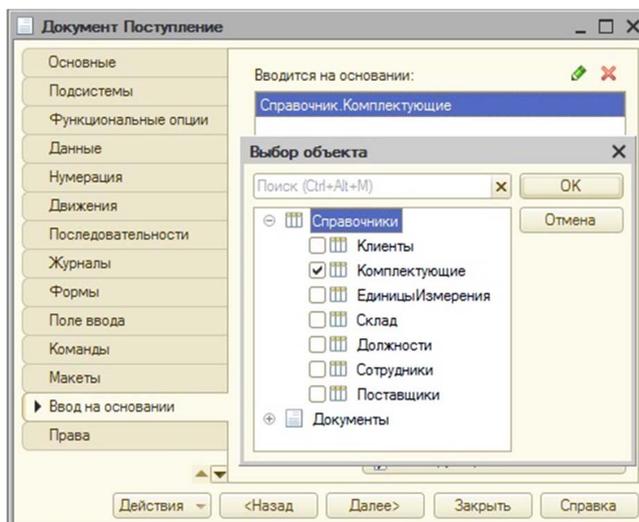


Рис. 28. Настройка *Ввода на основании* для документа *Поступление*

2. Откройте конструктор *Ввод на основании*, щелкнув по одной-единственной кнопке.

3. Установите соответствие полей документа *Поступление* и справочника *Комплектующие*. Для этого в нижней части, где указаны поля справочника, выделите нужное поле и дважды щелкните по соответствующему реквизиту основания справочника в верхней части конструктора. Можно воспользоваться кнопкой <Заполнить выражения>, но при этом контролируйте связи всех требуемых задач полей. При необходимости добавить связи можно вручную. Результат показан на рис. 29. Закройте конструктор, щелкнув по кнопке <ОК>.

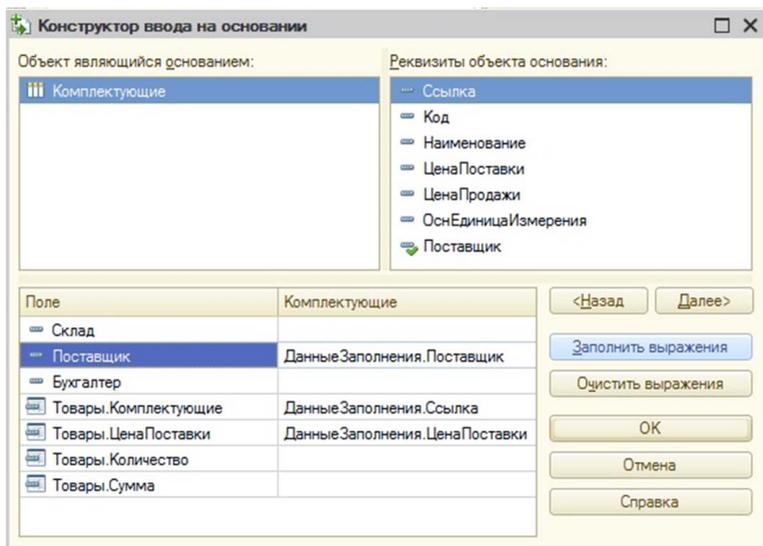


Рис. 29. Конструктор *Ввод на основании*

4. В модуле объекта документа *Поступление* конструктор создал программный код обработчика события *ОбработкаЗаполнения*, как показано на рис. 30.

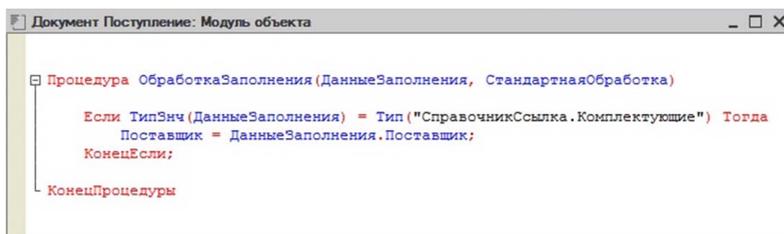


Рис. 30. Созданный конструктором код обработчика события *ОбработкаЗаполнения*

5. Допишите код. Для этого в коде обработки пропишите заполнение полей *Комплектующие* и *ЦенаПоставки* для табличной части *Товары* документа *Поступление*, как показано на рис. 31.

```

НоваяСтрока = Товары.Добавить ();
НоваяСтрока.Комплектующие = ДанныеЗаполнения.Ссылка;
НоваяСтрока.ЦенаПоставки = ДанныеЗаполнения.ЦенаПоставки;

```

Рис. 31. Заполнение табличной части документа в коде обработчика события *ОбработкаЗаполнения*

6. Запустите отладку конфигурации и проверьте работоспособность процедуры *ОбработкаЗаполнения*. Создайте несколько документов на основании элементов справочника *Комплектующие*, как показано на рис. 32.

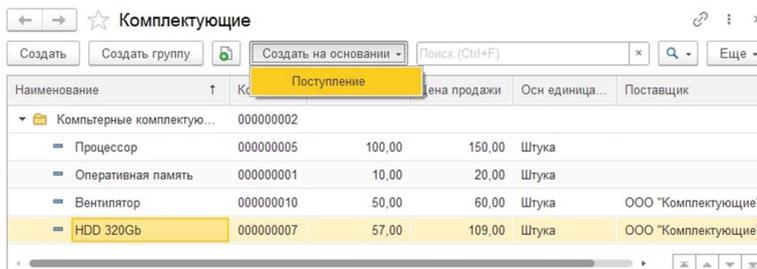


Рис. 32. Ввод документа на основании элемента справочника

7. Оформите документ *ВозвратКомплектующих* как основание для документа *СписаниеКомплектующих*.

8. Настройте для справочника *Поставщики* критерий отбора, чтобы при выборе поставщика можно было просмотреть весь список документов *Поступление*, где фигурирует данный поставщик.

Для выполнения потребуется объект *КритерииОтбора*. Объект *КритерииОтбора* как общий объект конфигурации используется, чтобы организовать поиск списка документов по определенному требованию. Для того чтобы добавить объект *КритерииОтбора* в конфигурацию, раскройте ветвь *Общие*, выделите объект *КритерииОтбора* и выполните в контекстном меню команду **Добавить**. Далее выполните последовательность действий с диалоговыми окнами: на вкладке *Основные* введите наименование *Поступление* (в соответствии с наименованием документа), на вкладке *Данные* укажите тип связи *СправочникСсылка.Поставщики*, на вкладке *Состав* поставьте флажок у реквизита *Поставщик* документа *Поступление*.

9. Настройте переход к списку документов *Поступление* при выборе конкретного поставщика.

Для этого откройте редактор формы элемента справочника *Поставщики* и на вкладке *Командный интерфейс* установите флажок видимости документа *Поступление*, как показано на рис. 33.

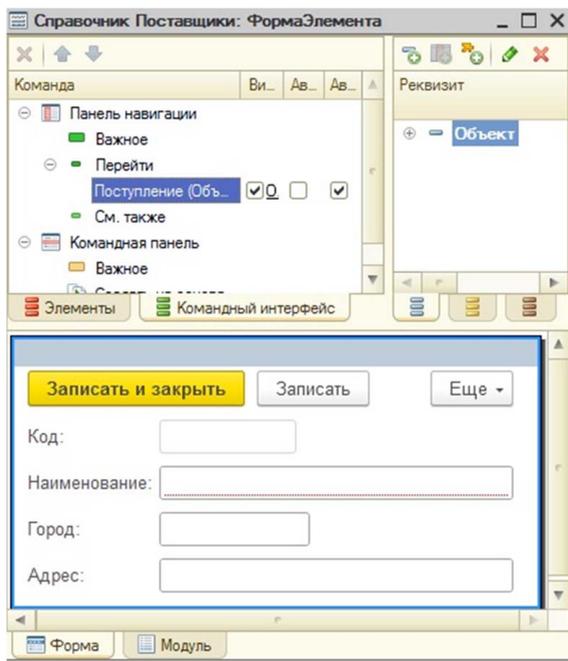


Рис. 33. Редактор формы элемента справочника *Поставщики*

10. Запустите отладку конфигурации и проверьте работоспособность перехода к списку документов *Поступление* при выборе конкретного поставщика, как показано на рис. 34.

☆ ООО "Комплекующие" (Поставщики) [Поступление](#)

Основное [Поступление](#)

Записать и закрыть Записать Еще ▾

Код:

Наименование:

Город:

Адрес:

Рис. 34. Переход к списку документов *Поступление* при выборе конкретного поставщика

11. Выгрузите информационную базу; имя файла должно быть следующее: Практическая работа 4.

Практическая работа 5 Регистры накопления

Цель работы – формирование практических навыков использования регистров накопления в «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих», используя регистры накопления, автоматизировать процесс учета данных об остатках комплектующих на складе и учета выдачи комплектующих сотруднику. В виде отчета вывести остатки комплектующих на складе за определенный период, а также представить отчет о выдаче комплектующих с группировкой по сотруднику.

Порядок выполнения работы

1. Добавьте в конфигурацию подсистему *Регистры*.
2. Добавьте объект *Регистр накопления ОстаткиКомплекующих-НаСкладе*; объект должен быть доступен в подсистеме *Регистры*.

Измерения регистра накопления: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Склад* (тип СправочникСсылка.Склад).

Ресурс: *Количество* (тип Число).

В системе «1С:Предприятие» одним из основных механизмов является учет накопления данных, за которые отвечают регистры.

Система предоставляет четыре вида регистров: регистры накопления, регистры сведений, регистры бухгалтерии, регистры расчета.

Регистр накопления как объект конфигурации предназначен для описания структуры накопления данных. Работает только с числовыми данными в разрезе нескольких измерений. При этом вид накапливаемой информации принято называть ресурсом регистра. Документы конфигурации в момент их проведения записывают свои движения в регистр, в этом случае их называют регистраторами. Отличительной особенностью регистра накопления является то, что у пользователя есть возможность просмотра записей и отсутствует возможность их редактирования.

Чтобы добавить объект *Регистр накопления* в конфигурацию, выделите ветку *Регистры накопления* и в контекстном меню выполните команду **Добавить**. Далее в диалоговом окне на вкладке *Основные* задайте имя регистра накопления *Выдача*, на вкладке *Подсистемы* установите флажок *Регистры*, на вкладке *Данные* задайте измерения, ресурсы и реквизиты регистра накопления.

3. Перейдите на вкладку *Регистраторы* и выберите в качестве документов-регистраторов документ *Поступление* и документ *ВыдачаСотруднику*. Закройте окно настроек регистра, щелкнув по кнопке <Закреть>.

4. В конфигураторе выделите ветку документа *Поступление* и в контекстном меню выполните команду **Конструкторы — Конструктор движения**.

5. В *Конструкторе движений* установите значения, как показано на рис. 35.

Установки следующие:

- тип движения регистра — *Приход*;
- в поле *Табличная часть* установите *Товары* (табличная часть документа);

— в нижней части конструктора в поле *Выражение* задайте формулы для вычисления значений измерений и ресурсов регистра при записи движений и установите соответствие измерений/ресурсов регистра с входными данными документа. Заполните поле *Выражение*, выбирая необходимое *Поле* и дважды щелкая по реквизиту документа. Нажмите кнопку <ОК>.

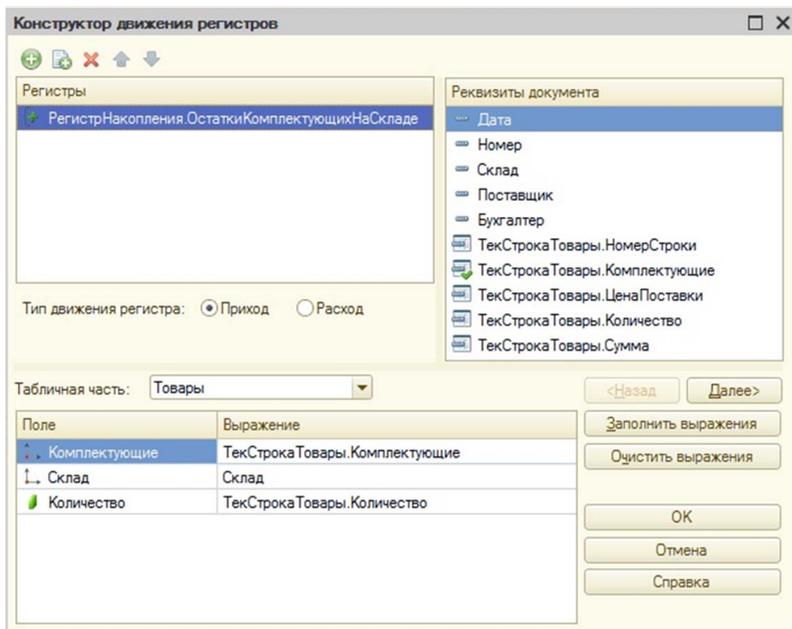


Рис. 35. Конструктор движений документа *Поступление*

В модуле объекта документа *Поступление* конструктор движений поместил обработчик события *ОбработкаПроведения*.

6. Настройте переход в регистр *ОстаткиКомплекующихНаСкладе* из формы документа *Поступление*.

Для этого откройте форму документа *Поступление* в редакторе форм, перейдите на вкладку *Командный интерфейс* и установите в ветке *Перейти* флажок видимости регистра *ОстаткиКомплекующихНаСкладе*, как показано на рис. 36.

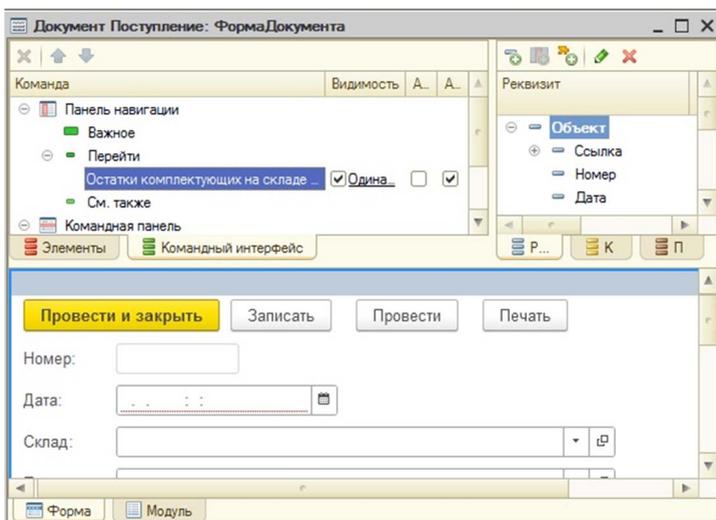


Рис. 36. Настройка командного интерфейса документа *Поступление*

7. Запустите отладку конфигурации и проверьте работоспособность системы. Перепроведите документы *Поступление*, хранящиеся в системе, или создайте новые документы. Проанализируйте данные регистра накопления *ОстаткиКомплектующихНаСкладе*.

8. Для документа *ВыдачаСотруднику* выполните аналогичные действия и установите в конструкторе движений настройки: Тип движения регистра — *Расход*, в поле *Табличная часть* выберите *Выдача* (табличная часть документа). Щелкните по кнопке <Заполнить выражения>. Проконтролируйте, что все связи установлены, при необходимости выполните вручную. Нажмите кнопку <ОК>.

9. Добавьте на форме документа *ВыдачаСотруднику* переход на регистр *ОстаткиКомплектующихНаСкладе*.

10. Запустите отладку конфигурации и проверьте работоспособность системы. Проанализируйте данные регистра накопления *ОстаткиКомплектующихНаСкладе* относительно документа *ВыдачаСотруднику*.

11. Настройте видимость регистра накопления *ОстаткиКомплектующихНаСкладе* в подсистеме *Регистры*.

Для этого в дереве конфигурации раскройте ветку *Общие*, выделите ветку *Подсистемы* и в контекстном меню выполните команду

Все подсистемы. В диалоговом окне в разделе *Подсистемы* выделите *Регистры* и в разделе *Командный интерфейс* установите флажок видимости регистра *ОстаткиКомплектующихНаСкладе*, как показано на рис. 37.

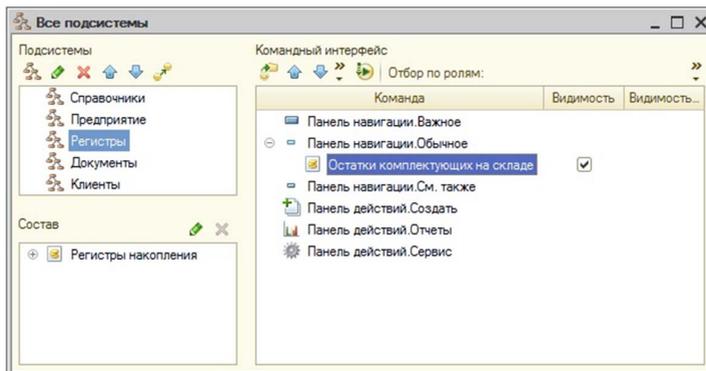


Рис. 37. Настройка видимости регистра *ОстаткиКомплектующихНаСкладе*

12. Для автоматизации учета выдачи комплектующих сотруднику добавьте в конфигурацию оборотный регистр накопления *Выдача*; объект должен быть доступен в подсистеме *Регистры*.

Измерения регистра накопления: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Сотрудник* (тип СправочникСсылка.Сотрудник).

Ресурс: *Количество* (тип Число).

Для этого в дереве конфигурации выделите объект *Регистры накопления* и в контекстном меню выполните команду **Добавить**. На вкладке *основные* введите наименование регистра *Выдача*, установите вид регистра – *Обороты*. На вкладке *Данные* задайте измерения, ресурсы регистра. На вкладке *Регистраторы* установите флажок у документа *ВыдачаСотруднику*.

13. В конфигураторе выделите ветку документа *ВыдачаСотруднику* и в контекстном меню выполните команду **Конструкторы – Конструктор движений**.

14. В Конструкторе движений в разделе *Регистры* добавьте в список регистр накопления *Выдача* и установите значения, как показано на рис. 38.

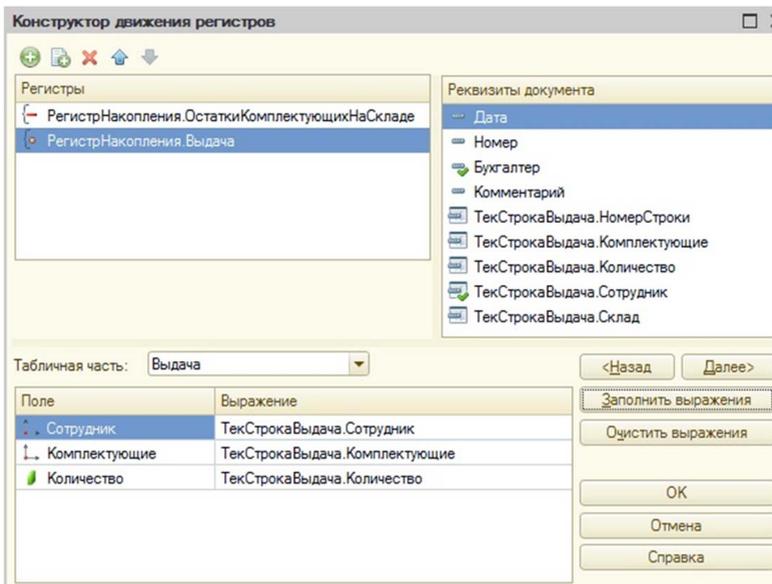


Рис. 38. Настройка регистра накопления *Выдача*

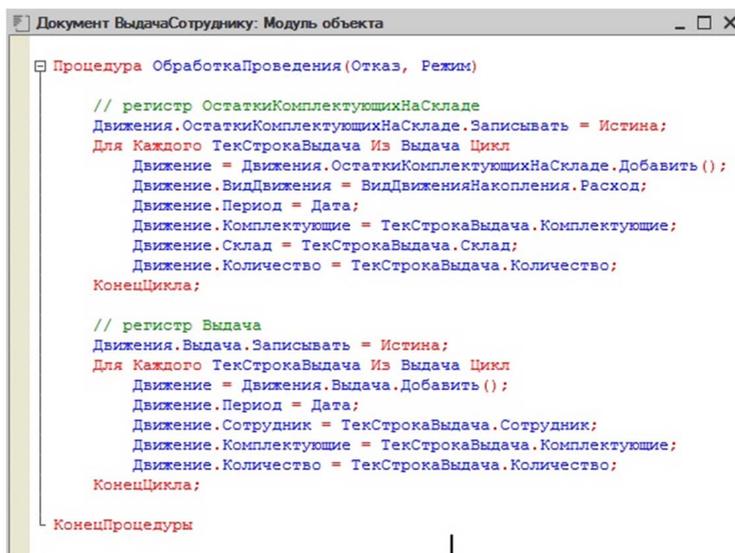


Рис. 39. Процедура *ОбработкаПроведения* в модуле документа *ВыдачаСотруднику*

В модуле объекта документа *ВыдачаСотруднику* в процедуру обработки события *ОбработкаПроведения* будет добавлен программный код проведения движений в регистр *Выдача*, как показано на рис. 39.

15. Для учета комплектующих у сотрудника добавьте в конфигурацию оборотный регистр накопления *КомплектующиеУСотрудников*; объект должен быть доступен в подсистеме *Регистры*. Регистр хранит движения документов *ВыдачаСотруднику* и *ВозвратКомплектующих*.

Измерения регистра накопления: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Сотрудник* (тип СправочникСсылка.Сотрудник).

Ресурс: *Количество* (тип Число).

Реквизит: *ДатаВозврата* (тип Дата).

16. Для представления данных о приходе, расходе, об остатках комплектующих добавьте в конфигурации объект отчет *ОстаткиКомплектующих*.

Для этого выделите в дереве объектов ветку *Отчеты* и выполните в контекстном меню команду **Добавить**. Далее выполните последовательность действий с диалоговыми окнами: на вкладке Основные введите имя отчета *ОстаткиКомплектующих* и откройте макет схемы компоновки данных, щелкнув по одноименной кнопке.

В конструкторе макета задайте имя схемы компоновки данных *ОсновнаяСхемаКомпоновкиДанных*.

На вкладке *Наборы данных* добавьте новый «набор данных — запрос».

Откройте конструктор запроса, щелкнув по одноименной кнопке, и выполните следующие действия: в списке *База данных* раскройте ветку *РегистрыНакопления*, выделите виртуальную таблицу *ОстаткиКомплектующихНаСкладе.ОстаткиИОбороты* и выберите поля *Склад*, *Комплектующие*, *КоличествоПриход*, *КоличествоРасход* и *КоличествоКонечныйОстаток* и нажмите кнопку <ОК>.

На вкладке *Настройки* конструктора в дереве структуры отчета выделите корневой элемент *Отчет* и в контекстном меню выберите команду **Новая группировка**. Не указывая поле группировки,

нажмите кнопку <ОК>. В структуре отчета появится группировка *Детальные записи*.

Настройте поля, которые будут выводиться в результат отчета. На закладке *Выбранные поля* в нижнем окне настроек выберите из списка полей: *Склад*, *Комплектующие*, *КоличествоПриход*, *КоличествоРасход*, *КоличествоКонечныйОстаток*.

На вкладке *Параметры* определите состав пользовательских настроек. Для этого установите флажки в первом столбце для каждого параметра *НачалоПериода* и *КонецПериода*. Эти поля будут находиться непосредственно в форме отчета.

В контекстном меню каждого параметра выберите *Свойства элемента пользовательских настроек* и установите флажок *Включать в пользовательские настройки*.

17. Установите видимость отчета в панели действий подсистемы *Документы*.

18. Запустите отладку конфигурации и проверьте работоспособность системы по формированию отчета *ОстаткиКомплектующих*.

19. Добавьте отчет *ОтчетВыдачаСотруднику* с группировкой по сотруднику. Для этого выполните аналогичные действия, как в предыдущем пункте, но при этом измените установки на вкладке *Настройки*. В дереве структуры отчета выделите корневой элемент *Отчет* и добавьте *Новую группировку* и в качестве поля группировки установите *Сотрудник*, нажмите кнопку <ОК>. Добавьте группировку *Детальные записи* с выбранными полями отчета.

20. Установите видимость отчета в панели действий подсистемы *Документы*.

21. Запустите отладку конфигурации и проверьте работоспособность системы, сформируйте отчет *ОтчетВыдачаСотруднику*.

22. Выгрузите информационную базу, имя файла должно быть следующим: Практическая работа 5.

Практическая работа 6

Регистры сведений

Цель работы — формирование практических навыков работы с регистрами сведений в «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» добавьте регистр сведений, с помощью которого автоматизируйте просмотр дополнительной информации о поставщиках на форме элемента справочника.

Порядок выполнения работы

1. Добавьте справочник *ВидИнформацииОПоставщике* со стандартными реквизитами; он должен быть доступен в подсистеме *Справочники*. Запустите отладку, заполните справочник данными *Город, Адрес, Телефон*.

2. Добавьте в конфигурацию объект регистр сведений *ИнформацияОПоставщике*; объект должен быть доступен в подсистеме *Регистры*.

Измерения регистра сведений: *Поставщики* (тип Справочник-Ссылка.Поставщики), *ВидИнформацииОПоставщике* (тип Справочник-Ссылка.ВидИнформацииОПоставщике).

Ресурс: *ЗначениеИнформации* (тип Строка).

Периодичность — *непериодический*, Режим записи — *независимый*.

Регистр сведений как объект конфигурации предназначен для хранения информации, входящей в состав некоторой комбинации значений, которую при необходимости можно привязать ко времени. В последнем случае регистры принято называть периодически. Отличительной особенностью регистров сведений от регистров накопления является то, что они могут хранить любые типы данных и у пользователя присутствует возможность изменять записи регистра.

Чтобы добавить регистр сведений в конфигурацию, выделите в дереве объектов ветку *Регистры сведений* и в контекстном меню выполните команду **Добавить**. Далее в диалоговом окне задайте имя регистра сведений *ИнформацияОПоставщике*, задайте установ-

ку регистру – *непериодический*, на вкладке *Подсистемы* установите флажок *Регистры*, на вкладке *Данные* задайте измерения, ресурсы и реквизиты регистра сведений.

3. Настройте автоматический переход к регистру сведений *ИнформацияОПоставщике* на форме элемента справочника *Поставщики*.

Чтобы на панели навигации формы элемента справочника *Поставщики* появилась ссылка на регистр сведений *Информация-ОПоставщике*, установите флажок свойства *Ведущее* для измерения *Поставщики* регистра сведений, как показано на рис. 40.

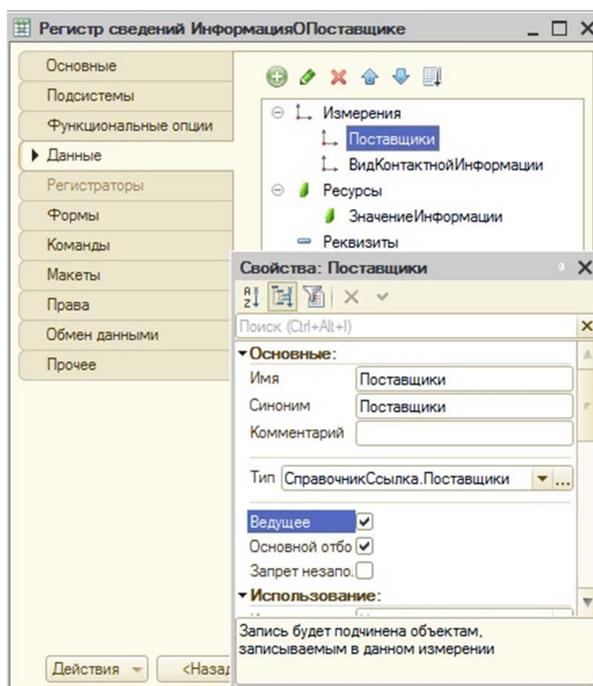


Рис. 40. Настройка свойства *Ведущее* для измерения *Поставщики*

В режиме *Предприятие* переход в регистр сведений *Информация-ОПоставщике* на форме элемента справочника *Поставщики* показан на рис. 41.

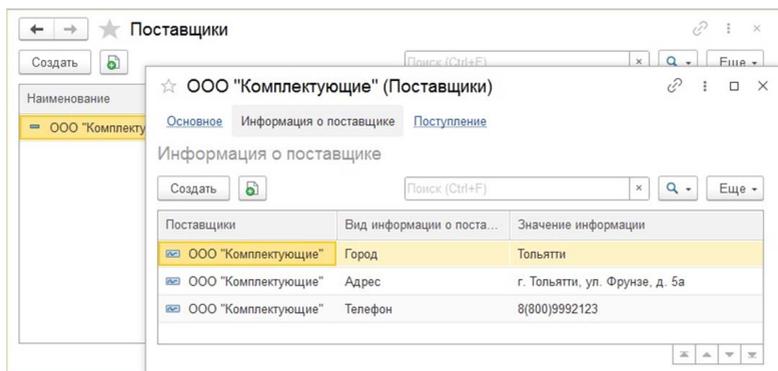


Рис. 41. Переход в регистр сведений *ИнформацияОПоставщике*

4. Настройте отбор записей в справочнике *Информация-ОПоставщике* по виду информации о поставщике *Город*.

Чтобы осуществить отбор записей, щелкните по кнопке <Еще> и выполните команду **Настройка списка**. Создайте правило, как показано на рис. 42.

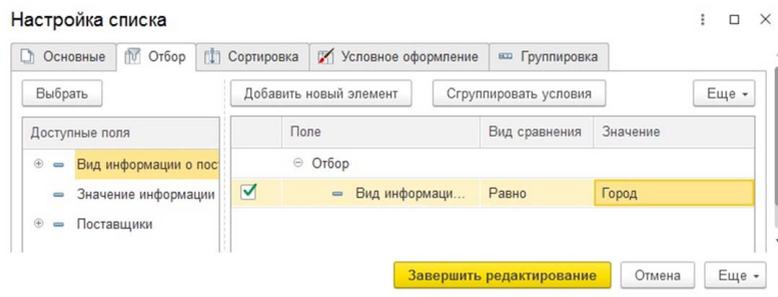


Рис. 42. Создание правила отбора записей по виду информации о поставщике *Город*

5. Осуществите программным способом вывод списка регистра сведений на форму элемента справочника *Поставщики* с отбором по текущей записи.

Динамический список как интерфейсный объект встроенного языка предназначен для отображения различных списков объектных данных (например, документы) или неobjектных данных (на-

пример, записей регистров) на экране, на форме. По сути, представляет собой табличную область с колонками и столбцами.

Откройте форму элемента справочника *Поставщики* в редакторе форм и добавьте на нее в разделе Реквизиты реквизит *ИнформацияОПоставщике* (тип Динамический список).

Установите источник данных для динамического списка и выберите в поле *Основная таблица* регистр сведений *ИнформацияОПоставщике*.

Перетащив реквизит на дерево *Формы*, добавьте элемент Таблица для отображения на форме элемента справочника *Поставщики* в соответствии с колонками регистра сведений.

6. Запустите отладку конфигурации и проверьте работоспособность системы, заполните значениями справочник *Поставщики* и регистр сведений *ИнформацияОПоставщике*, как показано на рис. 43.

Поставщики	Вид информации о по...	Значение информации
ООО "Марс"	Город	Самара
ООО "Марс"	Адрес	Самара, Московское ш., д. 111
ООО "Марс"	Телефон	8 (927) 1256823
ООО "Комплектующие"	Город	Тольятти
ООО "Комплектующие"	Адрес	г. Тольятти, ул. Фрунзе, д. 5а
ООО "Комплектующие"	Телефон	8(800)9992123

Рис. 43. Форма элемента справочника *Поставщики* в режиме *Предприятие*

7. Программным способом настройте отбор записей в регистре сведений на текущую запись поставщика.

Для этого откройте модуль формы элемента справочника *Поставщики* и добавьте заготовку для процедуры обработчика события *ПриОткрытии*, выполнив команду **Текст – Процедуры и функции**. Режим создания процедуры – *Создать на клиенте*.

Введите программный код, как показано на рис. 44.

```
4НаКлиенте
□ Процедура ПриОткрытии(Отказ)
    ЭлементОтбора = ЭтаФорма.ИнформацияПоставшике.Отбор.Элементы.Добавить(Тип("ЭлементОтбораКомпозовкиДанных"));
    ЭлементОтбора.ЛевоеЗначение = Новый.ПолеКомпозовкиДанных("Поставшики");
    ЭлементОтбора.ВидСравнения = ВидСравненияКомпозовкиДанных.Равно;
    ЭлементОтбора.ПравоеЗначение = Объект.Ссылка;
    ЭлементОтбора.Использование = Истина;
КонецПроцедуры
```

Рис. 44. Программный код процедуры *ПриОткрытии*

8. Запустите отладку конфигурации и убедитесь, что отбор по поставщику работает.

9. Добавьте в конфигурацию объект регистр сведений *ЦеныПродажиКомплектующих*, объект должен быть доступен в подсистеме Регистра.

Измерения регистра сведений: *Комплектующие* (тип СправочникСсылка.Компектующие).

Ресурс: *ЦенаПродажи* (тип Число).

Периодичность — *в пределах дня*, Режим записи — *независимый*.

10. Запустите отладку конфигурации и заполните регистр сведений *ЦеныПродажиКомплектующих*, установив цены продажи для каждой позиции комплектующих на разные даты.

11. Выгрузите информационную базу; имя файла должно быть следующее: Практическая работа 6.

Практическая работа 7

Создание произвольного макета отчета в «1С:Предприятие»

Цель работы — формирование практических навыков работы с прикладным объектом Макет в «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» автоматизировать процесс обработки сотрудником заказа клиентом комплектующих, все движения документа сохранить в регистре накопления. В виде отчета вывести сумму, полученную от клиента за определенный вид комплектующих.

Порядок выполнения работы

1. Добавьте в конфигурацию подсистему *ЗаказыКлиентов*.
2. Добавьте в конфигурацию справочник *Клиенты*; справочник должен быть доступен в подсистеме *ЗаказыКлиентов*. В справочнике *Клиенты* определите следующие реквизиты: *Адрес* (тип Строка), *Телефон* (тип Строка), *ЭлектроннаяПочта* (тип Строка).
3. Запустите отладку конфигурации и заполните данными справочник *Клиенты*.
4. Добавьте в конфигурацию документ *Заказ*; документ должен быть доступен в подсистеме *ЗаказыКлиентов*. В документе *Заказ* определите реквизиты *Клиент* (тип СправочникСсылка.Клиенты), *Адрес* (тип Строка), добавьте табличную часть *Оплата* с реквизитами *Комплектующие* (СправочникСсылка.Комплектующие), *ЦенаПродажи* (тип Число), *Количество* (тип Число), *ВнесеннаяСумма* (тип Число).
5. Автоматизируйте заполнение поля *Адрес* при выборе клиента из справочника.

Для этого на форме документа *Заказ* дважды щелкните по полю *Клиент*, далее в окне свойств в разделе *События* выберите событие *ПриИзменении* и щелкните по кнопке с лупой, чтобы открыть модуль формы, в котором будет создана процедура обработчика события *КлиентПриИзменении()*. При создании обработчика события выберите вариант *Создать на клиенте*. В модуле формы документа напишите программный код; текст процедуры показан на рис. 45.

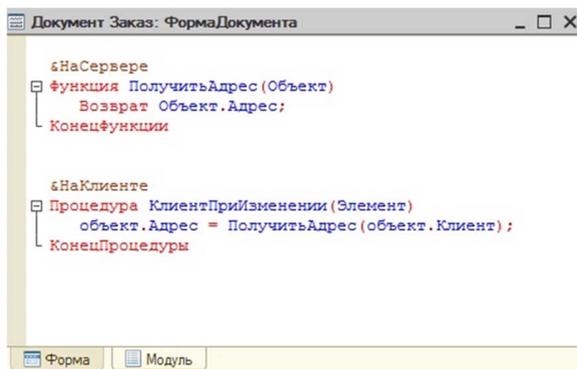


Рис. 45. Процедура *КлиентПриИзменении()*

6. Автоматизируйте подстановку *ЦеныПродажи* из регистра сведений *ЦеныПродажиКомплектующих*.

Для этого на форме документа *Заказ* дважды щелкните в табличной части *Оплата* по полю *Комплектующие*, далее в окне свойств в разделе *События* выберите событие *ПриИзменении* и щелкните по кнопке с лупой, чтобы открыть модуль формы, в котором будет создана процедура обработчика события *ОплатаКомплектующие-ПриИзменении()*. При создании обработчика события выберите вариант *Создать на клиенте*. В модуле формы документа в части *&НаКлиенте* введите программный код, как показано на рис. 46.

```
«НаКлиенте
□ Процедура ОплатаКомплектующиеПриИзменении (Элемент)
  СтрокаОплата = Элементы.Оплата.ТекущиеДанные;
  СтрокаОплата.ЦенаПродажи = ПолучитьЦенуПродажиНаСервере (Объект.Дата, СтрокаОплата.Комплектующие);
КонецПроцедуры
```

Рис. 46. Программный код процедуры *ОплатаКомплектующиеПриИзменении*

В программном коде присутствует функция *ПолучитьЦенуПродажиНаСервере()*, которая имеет параметры *Объект.Дата* и *СтрокаОплаты.Комплектующие*, которые передают соответственно дату документа для отбора *ЦеныПродажи* и значение выбранного комплектующего.

7. Программный код функции *ПолучитьЦенуПродажиНаСервере()* напишите в части *&НаСервереБезКонтекста*, как показано на рис. 47.

```
«НаСервереБезКонтекста
□ функция ПолучитьЦенуПродажиНаСервере (Дата, Комплектующие) Экспорт
  Отбор = Новый Структура ("Комплектующие", Комплектующие);

  Ресурс = РегистрыСведений.ЦенаПродажиКомплектующих.ПолучитьПоследнее (Дата, Отбор);

  Возврат Ресурс.ЦенаПродажи;
Конецфункции
```

Рис. 47. Код функции *ПолучитьЦенуПродажиНаСервере()*

Стоит отметить, что обратиться к регистру сведений можно только с сервера. В функции *ПолучитьЦенуПродажиНаСервере()* используется метод *ПолучитьПоследнее()*, где в качестве параметров указывается *Дата* и *Отбор*, который является структурой, в которую передается измерение *Комплектующие*.

8. Автоматизируйте в документе расчет суммы заказа.

Для этого на форме документа *Заказ* дважды щелкните в табличной части *Оплата* по полю *Количество*, далее в окне свойств в разделе *События* выберите событие *ПриИзменении* и щелкните по кнопке с лупой, чтобы открыть модуль формы, в котором будет создана процедура обработчика события *ОплатаКоличествоПриИзменении()*. При создании обработчика события выберите вариант *Создать на клиенте*. В модуле формы документа напишите программный код; текст процедуры показан на рис. 48.

```
сНаКлиенте
□ Процедура ОплатаКоличествоПриИзменении(Элемент)
  СтрТабЧасти = Элементы.Оплата.ТекущиеДанные;
  СтрТабЧасти.ВнесеннаяСумма = СтрТабЧасти.Количество*СтрТабЧасти.ЦенаПродажи;
КонецПроцедуры
```

Рис. 48. Обработчик события *ОплатаКоличествоПриИзменении*

Аналогичные действия выполните для поля *ЦенаПродажи*.

9. Добавьте в конфигурацию регистр накопления *Заказы* для учета данных документа *Заказ*; объект должен быть доступен в подсистеме *Регистры*.

Измерения регистра накопления: *Комплектующие* (тип СправочникСсылка.Комплектующие), *Клиент* (тип СправочникСсылка.Клиенты).

Ресурс: *Количество* (тип Число), *Сумма* (тип Число).

Для этого в дереве конфигурации выделите объект *Регистры накопления* и в контекстном меню выполните команду **Добавить**. На вкладке *Основные* введите наименование регистра *Заказы*, установите вид регистра – *Остатки*. На вкладке *Данные* задайте измерения, ресурсы регистра. На вкладке *Регистраторы* установите флажок у документа *Заказ*.

В конфигураторе выделите ветку документа *Заказ* и в контекстном меню выполните команду **Конструкторы – Конструктор движений**. Выполните настройки движений регистров, как показано на рис. 49.

10. Запустите отладку конфигурации и проверьте работоспособность записи движений документа *Заказ*.

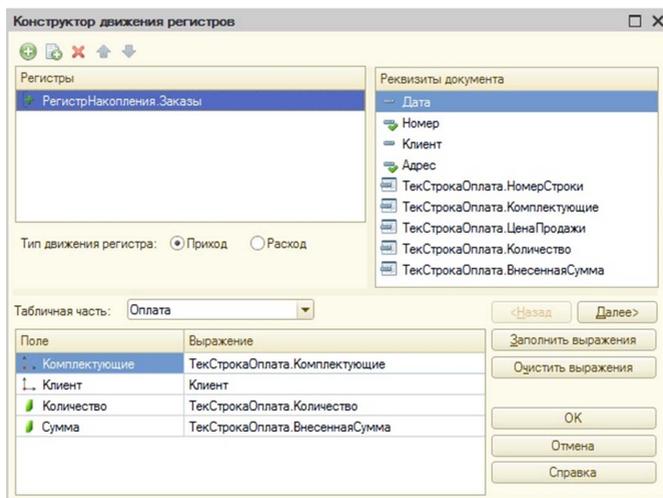


Рис. 49. Окно конструктора движения регистров для документа *Заказ*

11. Добавьте отчет *ОплатаКомплектующих*. Отчет должен быть доступен в подсистеме *ЗаказыКлиентов*.

Одной из отличительных особенностей системы «1С:Предприятие» является получение информации в удобной для пользователя форме. К таким объектам конфигурации относят *Отчет*.

Отчет как объект конфигурации применяется для разработки механизмов получения сводной информации на основе критериев пользователя. Как все остальные объекты системы «1С:Предприятие», отчет может иметь определенную форму и макет. Форма в данном случае отвечает, как данные будут визуализированы в интерфейсе пользователя, а макет предназначен для приведения всей информации к требуемому виду, например, в виде табличного документа.

При построении отчетов в системе «1С:Предприятие» важную роль играют регистры. Регистры, как объекты конфигурации, накапливают данные, фиксируют различные факты событий деятельности организации. Использование регистров является достаточно эффективным методом при построении отчетов. В дальнейшем именно использование регистров и будет применено в конфигурации для построения отчетов. При выполнении этого задания разработаем макет отчета, используя только объект *Документы*, присутствующий в конфигурации.

Для добавления *Отчета* в дереве конфигурации выделите объект *Отчет* и в контекстном меню выполните команду *Добавить*. Задайте имя *ОплатаКомплектующих*.

12. Добавьте макет отчета. Для этого в ветке отчета *Оплата-Комплектующих* выделите объект *Макеты* и в контекстном меню выполните команду *Добавить*. В диалоговом окне выполните действия: введите имя *МакетОтчета*, выберите тип макета, установив *Табличный документ*.

13. Постройте макет «с нуля». Добавьте область *Заголовок*.

При построении макета, представленного как табличный документ, используется механизм областей. Области могут быть горизонтальными или вертикальными. Горизонтальная секция объединяет одну или несколько строк табличного документа. Чтобы добавить область, выделите определенное количество строк и выполните команду *Таблица – Имена – Назначить имя*. Задайте имя области *Заголовок*. Далее в пустую ячейку области *Заголовка* введите текст *Отчет по оплате комплектующих*, как показано на рис. 50.

	1	2	3	4	5	6	7	8
Заголовок								
		Отчет по оплате комплектующих						

Рис. 50. Область *Заголовок* макета

14. Аналогично добавьте область *Шапка*, содержащую текст *Клиент, Комплектующие, Сумма* как наименования колонок.

15. Добавьте область *Детали*, содержащую переменные для каждого столбца, которые будут использованы в коде процедуры заполнения макета отчета.

Для этого аналогичным способом, как и для предыдущих областей, добавьте область *Детали*. Далее в свойствах ячейки в разделе *Макет* в поле *Заполнение* установите *Параметр*, в поле *Параметр* введите его значение, например, как показано на рис. 51, *Клиент*. Наименования параметров должны совпадать с наименованиями объектов конфигурации, их реквизитов и прочее. Аналогично выполните все действия для каждой колонки макета.

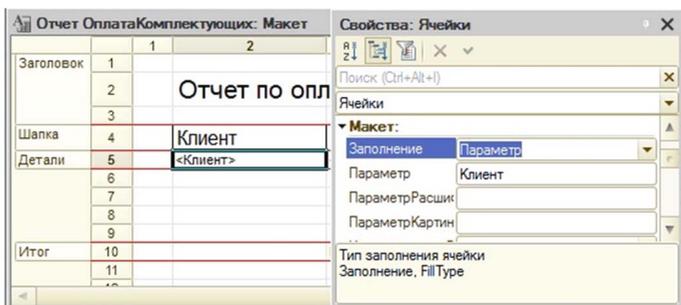


Рис. 51. Установка параметра

16. Оформите отчет. Для этого в свойствах ячейки установите шрифт, кегль шрифта, цвет текста, цвет и стиль линий границ таблицы макета и прочие настройки, как показано на рис. 52.

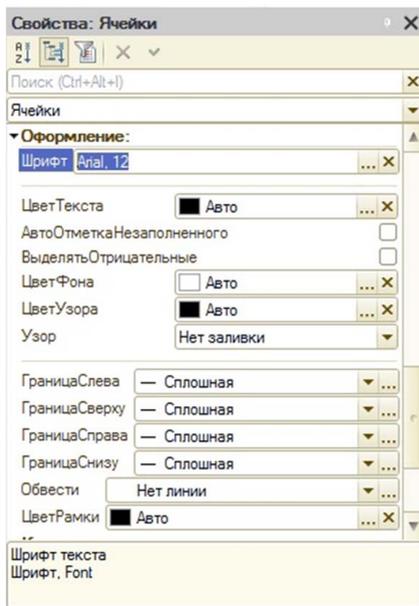


Рис. 52. Оформление макета отчета

17. Добавьте область *Итого*. В пустую ячейку введите текст «*Всего*» и также добавьте параметр *Всего*.

18. Добавьте форму для отчета *ОплатаКомплектующих*, чтобы его визуализировать в интерфейсе пользователя.

Форму отчета можно разработать несколькими способами: с помощью системы компоновки данных и самостоятельно.

Система компоновки данных – механизм, который предоставляет платформа «1С:Предприятие». Данный механизм включает в себя работу с конструктором форм, что помогает пользователю быстро, выбирая нужные объекты, устанавливая связи между ними, разработать ту или иную форму.

Для того чтобы добавить форму, выделите в ветке отчета *ОплатаКомплектующих* объект *Формы* и в контекстном меню выполните команду *Добавить*. Далее в диалоговом окне задайте имя формы *ФормаОтчета*, можно сразу ее назначить основной, установив флажок.

Выполняя данное задание, разработайте форму отчета самостоятельно, не надо использовать систему компоновки данных, поэтому флажок не ставим. После щелчка по кнопке <Готово> откроется редактор управляемой формы с пустой формой, как показано на рис. 53.

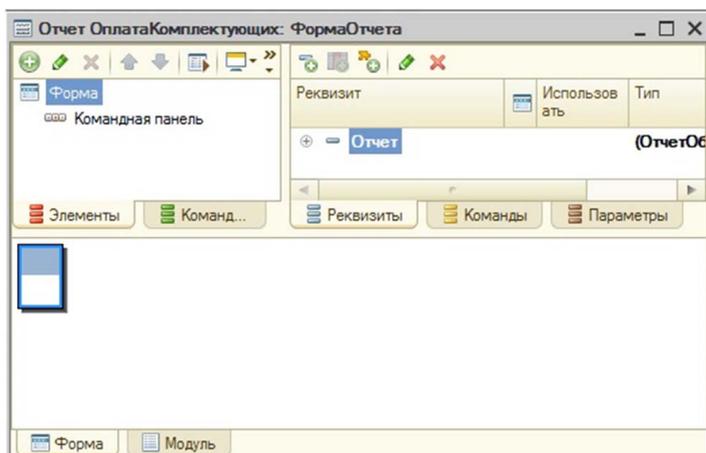


Рис. 53. Форма *ФормаОтчета*

19. Создайте форму отчета, добавьте программный код обработки данных для формирования отчета.

Отчет должен отображаться в пользовательском интерфейсе, чтобы с ним было удобно работать. Для этого необходимо выпол-

нить следующие действия: добавить область, куда будут выводиться данные, добавить кнопку <Сформировать>, чтобы запустить обработку данных, и вывести отчет согласно разработанному макету, написать программный код обработки данных и назначить его выполнение на кнопку <Сформировать>.

Чтобы создать область данных на форме отчета, в редакторе формы в разделе *Реквизиты* добавьте реквизит *ТабДок* и установите в свойствах тип *Табличный документ*, как показано на рис. 54. Перетащите реквизит *ТабДок* мышью в раздел *Элементы* на корень дерева *Форма*. Область появится на форме отчета. В свойствах области *ТабДок* установите в поле *ПоложениеЗаголовка* значение *Нет*, чтобы заголовок *ТабДок* не отображался на форме.

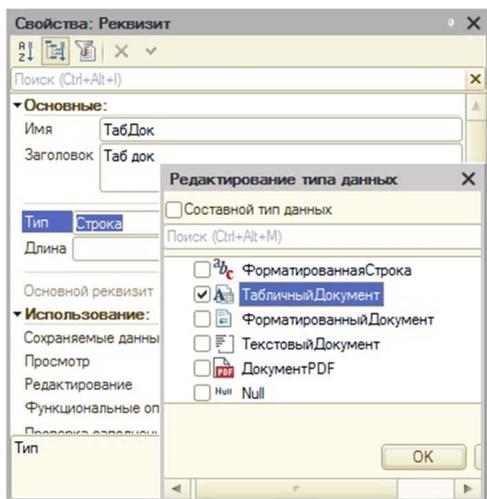


Рис. 54. Выбор типа области *ТабДок*

Чтобы добавить кнопку <Сформировать>, на форму отчета на вкладке *Команды* раздела *Реквизиты* добавьте команду *Сформировать*. В свойствах команды в разделе *Представление* задайте стандартную картинку *СформироватьОтчет* и установите в поле *Отображение* значение *Картинка и текст*. Перетащите команду *Сформировать* в дерево элементов *Формы* на *Командную панель*. Для того чтобы кнопка <Сформировать> отобразилась на форме, в свойствах *Формы* в разделе *Использование* установите в поле *Поло-*

жение *Командной Панели* значение *Авто* (по умолчанию установлено значение *Нет*), как показано на рис. 55.

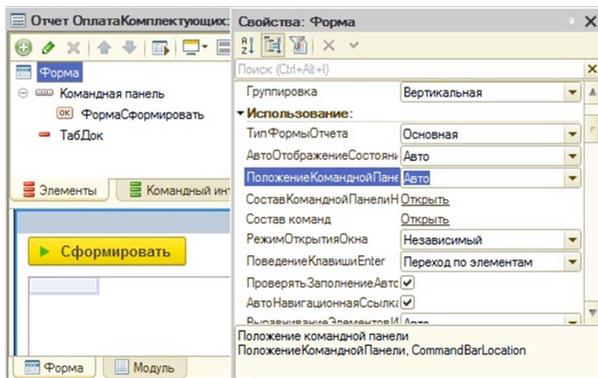


Рис. 55. Настройка свойств *Формы*

Чтобы процесс формирования отчета заработал, необходимо создать процедуру обработки действия. В свойствах команды *Сформировать* выберите свойство *Действие* и щелкните по кнопке с лупой. Далее в диалоговом окне выберите вариант разработчика. Так как в процедуре обработчика будет использован запрос, а запросы выполняются только на сервере, выберите вариант *Создать на клиенте* и процедуру на сервере, как показано на рис. 56.

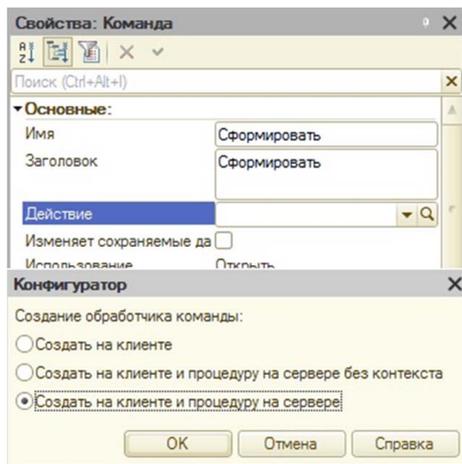


Рис. 56. Выбор вида разработчика для команды *Сформировать*

И последнее. Напишите программный код процедуры *СформироватьНаСервере* по следующему алгоритму:

– сформируйте запрос, с помощью которого будет происходить выборка данных из документа *Заказ* и его табличной части *Оплата*, как показано на рис. 57;

```
Запрос = Новый Запрос;  
Запрос.Текст =  
    "ВЫБРАТЬ  
    |   Заказ.Клиент,  
    |   Заказ.Оплата.(Комплектующие, ВнесеннаяСумма)  
    |ИЗ  
    |   Документ.Заказ КАК Заказ";
```

Рис. 57. Текст запроса

– установите с помощью методов *ПолучитьМакет()*, *ПолучитьОбласть()* связь с разработанным макетом *МакетОтчета* и выведите его заголовок, области макета в отчет с помощью метода *Вывести()*, как показано на рис. 58;

```
Макет = Отчеты.ОплатаКомплектующих.ПолучитьМакет("МакетОтчета");  
ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");  
ОбластьШапка = Макет.ПолучитьОбласть("Шапка");  
ОбластьДетали = Макет.ПолучитьОбласть("Детали");  
ОбластьИтог = Макет.ПолучитьОбласть("Итог");  
ТабДок.Очистить();  
ТабДок.Вывести(ОбластьЗаголовок);  
ТабДок.Вывести(ОбластьШапка);
```

Рис. 58. Определение макета

– запустите запрос на выполнение, используя метод *Выполнить()*, а с помощью метода *Выбрать()* выберите данные из полученного результата, как показано на рис. 59;

```
Выборка = Запрос.Выполнить().Выбрать();
```

Рис. 59. Выполнение запроса. Выборка данных

– осуществите с помощью цикла *Пока Цикл* перебор записей документа *Заказ* и данных его табличной части *Оплата*, каждый раз выводя их в область *Детали* отчета. Введите в код переменную *СуммаИтог*, произведите подсчет итоговой суммы. Выведите результат в области *Итог*, как показано на рис. 60.

```

СуммаИтог = 0;

Пока Выборка.Следующий() Цикл
    ОбластьДетали.Параметры.Заполнить(Выборка);

    ВыборкаТабЧасти = Выборка.Оплата.Выбрать();
    Пока ВыборкаТабЧасти.Следующий() Цикл
        ОбластьДетали.Параметры.Заполнить(ВыборкаТабЧасти);
        ТабДок.Вывести(ОбластьДетали, ВыборкаТабЧасти.Уровень());
        СуммаИтог = СуммаИтог + ВыборкаТабЧасти.ВнесеннаяСумма;
    КонечЦикла;
КонечЦикла;

ОбластьИтог.Параметры.Всего = СуммаИтог;
ТабДок.Вывести(ОбластьИтог);

```

Рис. 60. Организация цикла *Пока Цикл*. Расчет суммы

20. Запустите отладку конфигурации и проверьте работоспособность заполнения документа *Заказ* и вывода отчета.

21. Выгрузите информационную базу; имя файла должно быть следующее: Практическая работа 7.

Практическая работа 8

Объект План видов характеристик в «1С:Предприятие»

Цель работы – формирование практических навыков использования объекта План видов характеристик при решении прикладных задач.

Задание

В информационной системе «Учет компьютерных комплектующих» автоматизировать ввод свойств характеристик комплектующих и их значений.

Порядок выполнения работы

Часто встречаются задачи, когда требуется добавить дополнительные характеристики для различных объектов учета. Когда речь идет об оборудовании, недостаточно назвать модель, помимо модели надо знать дополнительные характеристики оборудования, например, для USB-флешки дополнительными характеристиками будут являться ее объем или интерфейс подключения.

1. Создайте справочник *ХарактеристикиКомплектующих* для хранения характеристик позиций комплектующих в справочни-

ке *Комплектующие*; объект должен быть доступен в подсистеме *Справочники*.

2. В справочнике *ХарактеристикиКомплектующих* будут храниться характеристики для всех позиций комплектующих. И чтобы как-то разделять позиции комплектующих, сделайте его подчиненным справочнику *Комплектующие*.

Для этого на вкладке *Владельцы* щелкните по кнопке <Редактировать список>, в диалоговом окне *Выбор владельца* установите флажок для справочника *Комплектующие*, как показано на рис. 61.

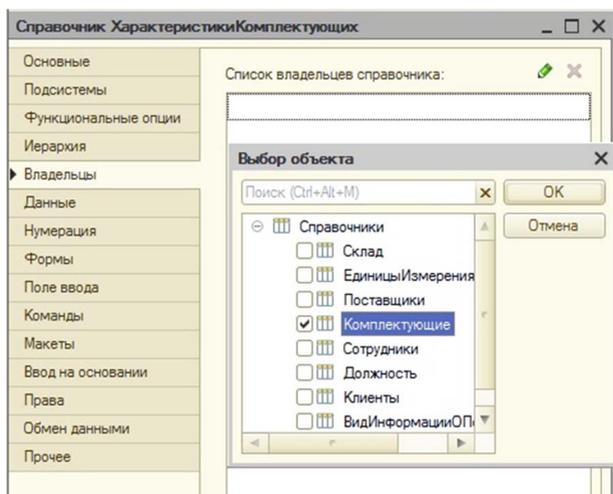


Рис. 61. Окно выбора владельца для справочника *ХарактеристикиКомплектующих*

3. В документе *Поступление* в табличной части *Товары* добавьте реквизит *Характеристика* (Тип *СправочникСсылка.ХарактеристикиКомплектующих*).

4. Отредактируйте форму элемента, чтобы визуализировать новый реквизит на форме.

5. Отредактируйте также регистр накопления *ОстаткиКомплектующихНаСкладе*, добавив измерение *Характеристика*.

6. Отредактируйте обработку проведения документа *Поступление*. Для этого в модуле объекта добавьте строку, как показано на рис. 62.

Движение.Характеристика = ТекСтрокаТовары.Характеристика;

Рис. 62. Фрагмент кода обработки проведения

7. Запустите отладку конфигурации и проверьте работоспособность системы.

8. Выполните настройку, чтобы при выборе характеристик определенной позиции комплектующих показывался не весь список характеристик, а только те, которые требуются для данной позиции.

Для этого в свойствах для реквизита *Характеристика* документа *Поступление* установите в поле *Связи параметров выбора* в качестве владельца *Товары.Комплектующие*, как показано на рис. 63.

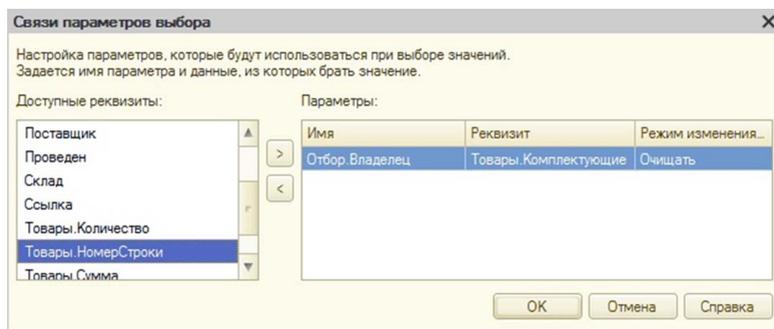


Рис. 63. Окно *Связи параметров выбора*

Такой установкой показано, что в момент открытия формы выбора характеристик интересны только те, у которых владелец соответствует позиции из текущей строки.

9. Запустите отладку конфигурации и каждый сохраненный в системе документ *Поступление* перепроведите, предварительно для каждой позиции документа добавьте характеристики комплектующих, например, для USB-флешки – 32Гб, USB 2.0.

10. Проанализируйте проводки регистра накопления *Остатки.КомплектующихНаСкладе*.

Задача решена, но есть существенные недостатки: нет ограничения на значения характеристики, то есть можно написать одну или несколько характеристик сразу в одну строку, а так как все добавленные характеристики имеют тип *Строка*, это не позволит проводить аналитический учет в разных разрезах.

Чтобы это исправить, вводится в конфигурацию объект *План видов характеристик*.

План видов характеристик представляет собой подобие справочника, но имеет отличительную особенность — он хранит информацию о типе значения для каждого элемента.

11. Добавьте в конфигурацию объект *План видов характеристик СвойстваКомплектующих*; объект должен быть доступен в подсистеме *Справочники*.

Для этого в дереве объектов выделите ветку *Планы типов характеристик* и в контекстном меню выполните команду **Добавить**. Далее в диалоговом окне задайте имя *СвойстваКомплектующих*.

12. Также на вкладке *Основные* задайте установку типа значений характеристик.

Для этого в поле *Тип значений характеристик* откройте окно *Редактирование типа данных*, установите флажок *Составной тип данных* и отметьте в списке разные типы, как показано на рис. 64.

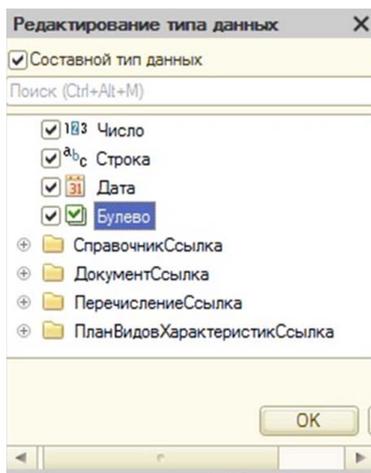


Рис. 64. Настройка *Тип значений характеристик*

13. Для того чтобы предусмотреть всевозможные свойства комплектующих, создайте справочник *ЗначенияСвойствКомплектующих*, где будут храниться все свойства комплектующих, хранение которых будет осуществляться в виде списка. Объект должен быть доступен в подсистеме *Справочники*.

14. Чтобы удобно было с ним работать, справочник подчините объекту *План видов характеристик СвойстваКомплектующих*, для того чтобы распределить всевозможные значения и свойства комплектующих.

Для этого на вкладке *Владельцы* устанавливаем *план видов характеристик СвойстваКомплектующих*.

15. В плане видов характеристик *СвойстваКомплектующих* в поле *Тип значений характеристик* добавьте только что созданный справочник *ЗначенияСвойствКомплектующих*. Также в поле *Дополнительные значения характеристик* установите справочник *ЗначенияСвойствКомплектующих*.

16. Запустите отладку конфигурации и заполните план видов характеристик *СвойстваКомплектующих* элементами, например, объем, частота, интерфейс и прочие.

17. Заполните справочник *ЗначенияСвойствКомплектующих*, введя по несколько элементов значений для каждого свойства комплектующих.

18. Установите соответствие между справочником *ХарактеристикиКомплектующих* и планом видов характеристик *СвойстваКомплектующих* для каждого из свойств.

Для этого добавьте регистр сведений *ЗначенияХарактеристик*; объект должен быть доступен в подсистеме *Регистры*.

Измерения регистра сведений: *Характеристика* (тип СправочникСсылка.ХарактеристикиКомплектующих), *Свойства* (тип ПланВидовХарактеристикСсылка.СвойстваКомплектующих).

Ресурс регистра сведений: *Значение* (тип Характеристика.СвойстваКомплектующих).

19. Настройте переход в регистр сведений *ЗначенияХарактеристик* из формы объекта *Характеристика*.

Для этого в свойствах измерения *Характеристик* регистра сведений *ЗначенияХарактеристик* установите флажок *Ведущее*.

20. Настройте тип значения свойства. Для этого в свойствах ресурса *Значение* в поле связи по выбору установите реквизит *Свойства*.

21. Для ресурса *Значение* установите *Связи параметров выбора*, для того чтобы выводились только подчиненные ему свойства. Для

этого в свойствах ресурса в поле *Связи параметров выбора* подставьте *Свойства*, как показано на рис. 65.

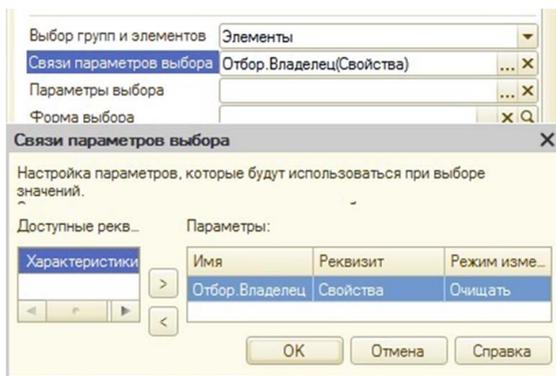


Рис. 65. Окно свойства *Связи параметров выбора*

22. Запустите отладку конфигурации и откройте запись в справочнике *Комплектующие* и заполните все характеристики комплектующего в соответствии со значениями его свойств. Выполните для всех записей справочника.

23. Добавьте отчет *ХарактеристикиКомплектующихНаСкладе*.

Для этого в дереве объектов конфигурации выделите ветку *Отчеты* и выполните в контекстном меню команду *Добавить*. В диалоговом окне введите наименование отчета *ХарактеристикиКомплектующихНаСкладе*.

Создайте схему компоновки данных, щелкнув по одноименной кнопке.

В окне конструктора добавьте новый набор данных — *Запрос* и откройте конструктор запросов, щелкнув по соответствующей кнопке.

В конструкторе запросов в разделе *Базы данных* раскройте ветку *Регистры* накопления и выберите *ОстаткиКомплектующихНаСкладеОстатки*, в разделе таблицы выберите все ее поля для отчета, закройте конструктор запросов, щелкнув по кнопке <ОК>.

В конструкторе *ОсновнаяСхемаКомпоновкиДанных* на вкладке *Ресурсы* выберите все доступные поля таблицы.

На вкладке *Настройки* добавьте новую группировку *ДетальныеЗаписи*, поле не заполняйте.

В нижней части на вкладке *Выбранные поля* добавьте поля *Комплектующие*, *Характеристика*, *Склад*, *Количество**Остаток* в отчет.

Установите отбор по выбранной характеристике. Для этого щелкните по кнопке <Свойства элементов пользовательских настроек>. Установите флажок *Отбор*, чтобы пользователь мог устанавливать настройки, установите также значение режима редактирования – *Обычный*, как показано на рис. 66.

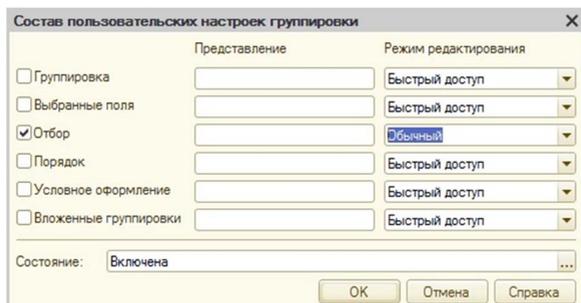


Рис. 66. Окно настройки пользовательских настроек

Для того чтобы при выполнении отбора по характеристикам они отображались в списке, выполните настройку справочника *ХарактеристикиКомплектующих*. Для этого в дереве объектов откройте справочник *ХарактеристикиКомплектующих* и в окне настроек справочника на вкладке *Данные* щелкните по кнопке <Характеристики>.

В диалоговом окне укажите, где находятся виды характеристик, в какой таблице хранятся значения характеристик, в каком поле хранится значение свойств характеристик, как показано на рис. 67.

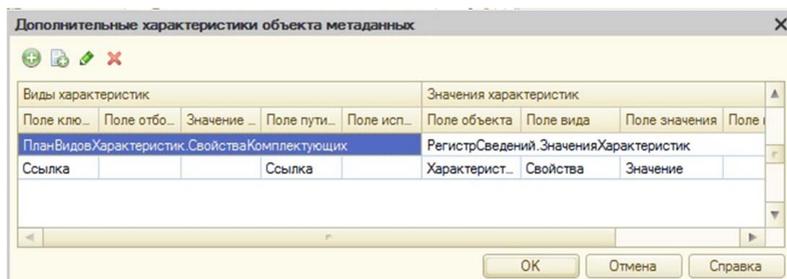


Рис. 67. Окно настройки дополнительных характеристик объекта метаданных

24. Запустите отладку конфигурации и сформируйте отчет *Характеристики Комплектующих На Складе*.

25. Произведите отбор записей по заданному свойству характеристики. Для этого щелкните по кнопке <Настройки> и введите значение отбора, например, как показано на рис. 68.

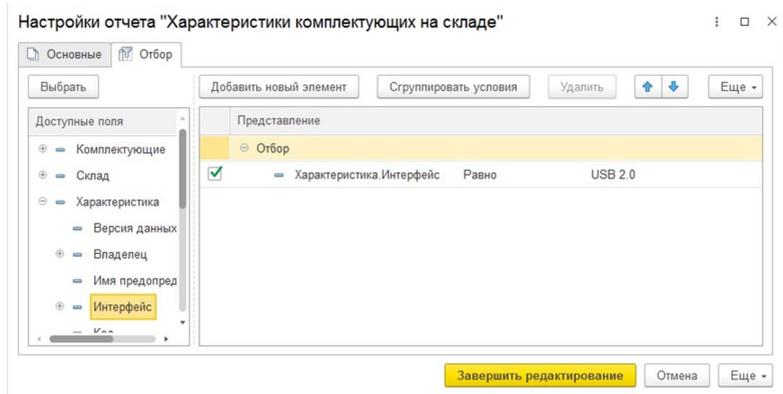


Рис. 68. Установка значения свойства *Интерфейс*

26. Выгрузите информационную базу; имя файла должно быть следующее: Практическая работа 8.

Практическая работа 9 Настройка прав. Роли и интерфейсы в «1С:Предприятие»

Цель работы – формирование практических навыков по настройке прав, ролей и интерфейсов в «1С:Предприятие».

Задание

В конфигурации информационной системы «Учет компьютерных комплектующих» спроектировать интерфейсы пользователей *Администратор*, *Директор* и *Бухгалтер*.

Порядок выполнения работы

1. Добавьте в конфигурацию пользователя *Администратор* с административными правами.

В системе «1С:Предприятие» присутствует механизм ролевой модели.

Право – минимальная единица ролевой модели.

Право – сущность, определяющая конкретное действие применительно к определенному объекту системы.

Например, для отчетов или обработок существуют только два вида права – использование и просмотр, а для документов существует большая совокупность прав – добавление, изменение, просмотр, проведение, удаление и прочие.

Роль – стандартный объект конфигурации, основное назначение которого состоит в настройке конкретных прав для работы с конкретным объектом.

Алгоритм добавления в систему пользователя с определенными правами следующий: добавить роль, установить для роли совокупность прав, добавить пользователя, назначить ему роль.

Чтобы добавить роль, раскройте в дереве объектов ветку *Общие*, выделите *Роли* и в контекстном меню выполните команду **Добавить**. Далее в диалоговом окне введите наименование роли *Администратор*. Установите для данной роли совокупность прав. Так как роль администратора не подразумевает работу с данными информационной базы, то выполните команду **Действия – Снять все права**, как показано на рис. 69.

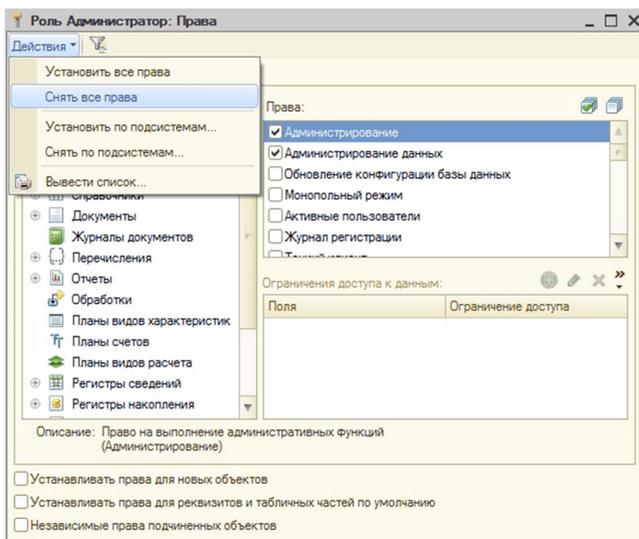


Рис. 69. Настройка прав роли *Администратор*

В разделе *Права* установите флажки *Администрирование* и *Администрирование данных*. А также проконтролируйте и снимите флажок *Устанавливать права для реквизитов и табличных частей по умолчанию*.

Обязательно обновите конфигурацию, выполнив команду **Конфигурация – Обновить конфигурацию информационной базы**.

Чтобы добавить пользователя *Администратор*, выполните команду **Администрирование – Пользователи**. В диалоговом окне *Список пользователей* выполните команду **Добавить**. Далее на вкладке *Основные* введите имя пользователя, на вкладке *Прочее* установите флажок для роли *Администратор*.

Запустите систему «1С:Предприятие» и в диалоговом окне запуска выберите учетную запись *Администратор*.

2. Добавьте пользователей *Директор* и *Бухгалтер*, назначьте пользователям роль *ДоступКДанным* для управления всей системой в целом.

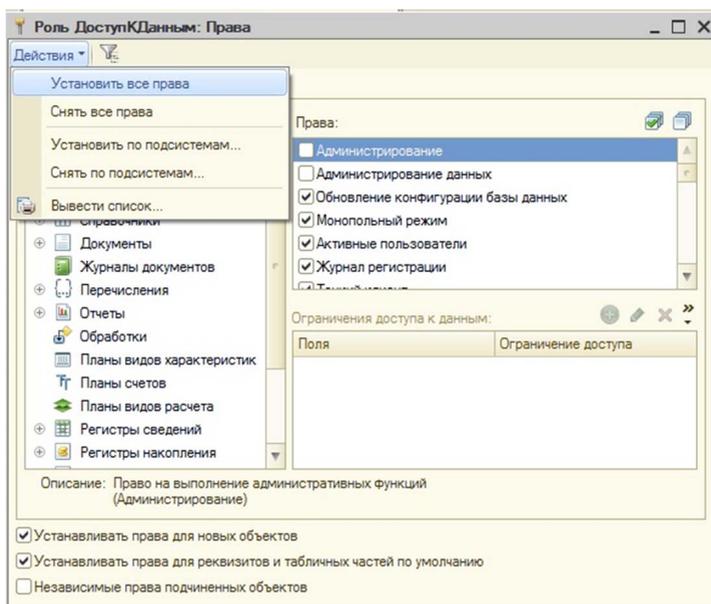


Рис. 70. Настройка прав роли *ДоступКДанным*

В дереве конфигурации раскройте ветку *Общие*, выделите *Роли* и в контекстном меню выполните команду **Добавить**. Далее в диалоговом окне введите наименование роли *ДоступКДанным*. Установите для данной роли все права, кроме администрирования, для этого выполните команду **Действия — Установить все права**, как показано на рис. 70, затем вручную снимите флажки *Администрирование* и *Администрирование данных*.

В окне *Список пользователей* добавьте пользователей *Директор* и *Бухгалтер*, выбрав для каждого роль *ДоступКДанным*.

3. Настройте рабочий стол пользователя *Директор*.

Первое, что видят пользователи, запуская программное решение, — это пользовательский интерфейс. Правильно спроектированный интерфейс обеспечивает удобство ввода данных для пользователей, так как если пользователю неудобно работать с данными, то они не могут эффективно выполнять свои функциональные обязанности по регистрации тех или иных событий и объектов, производительность их работы снижается.

Вторая важная задача пользовательского интерфейса — увеличение скорости поиска информации, чтобы пользователи могли быстро принимать решения на основании полученной информации и выполнять свои рабочие функции. Также при проектировании интерфейса формируются основные понятия предметной области.

Алгоритм создания интерфейса пользователя следующий: создать общую форму, на которой разместить все элементы интерфейса, добавить интерфейсный слой, чтобы интерфейс был доступен только определенному пользователю.

Чтобы создать общую форму, в дереве объектов раскройте ветку *Общие*, выделите *Общие формы* и в контекстном меню выполните команду **Добавить**. В диалоговом окне введите наименование формы *ИнтерфейсДиректора*.

В редакторе форм в дереве *Форма* добавьте три элемента *Декорация—Надпись* для вывода ссылок на существующие в системе отчеты *ОтчетВыдачиСотруднику*, *ОстаткиКомплектующих*, *ОплатаКомплектующих* в пользовательском интерфейсе. При этом в палитре свойств для каждого элемента установите флажок в поле *Гиперссылка* для отображения перехода к объекту конфигурации

в виде гиперссылки. Также уберите с формы *Командную панель*, сняв в свойствах флажок *Автозаполнение*.

Чтобы данная форма открывалась при входе пользователя в систему, в дереве объектов выделите корень конфигурации и в контекстном меню выполните команду **Открыть рабочую область начальной страницы**, как показано на рис. 71.

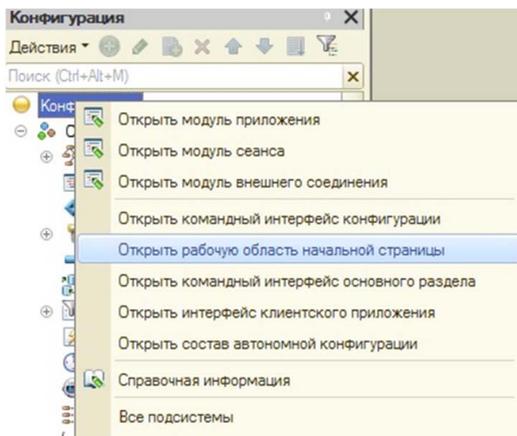


Рис. 71. Команда **Открыть рабочую область начальной страницы**

Далее в редакторе выбрать форму *ИнтерфейсДиректора* для автоматического ее открытия в программном решении.

Чтобы разработанный рабочий стол отображался только у пользователя Директор, добавьте интерфейсный слой. Для этого добавьте роль *ИнтерфейсДиректора* и отключите *Все права*, выполнив команду меню **Действия – Снять все права**. Данная роль предназначена не для установки набора прав пользователя, а является идентификатором интерфейсного слоя, под которым будет настроена видимость интерфейсных элементов. Для этого откройте окно настройки рабочей области начальной страницы и снимите флажок *Видимость*, перейдите по ссылке *Видимость* в настройки и установите видимость для интерфейсного слоя *ИнтерфейсДиректора*, как показано на рис 72.

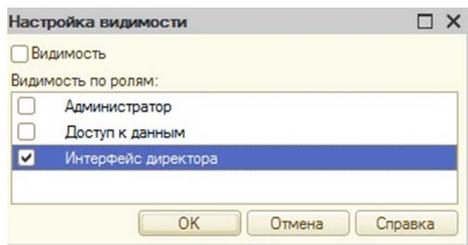


Рис. 72. Окно настройки видимости интерфейсного слоя *ИнтерфейсДиректора*

Обновите конфигурацию базы данных. И последнее. Назначьте интерфейсный слой пользователю *Директор*. Для этого выполните команду **Администрирование – Пользователи**. Для пользователя *Директор* добавьте роль *ИнтерфейсДиректора*, как показано на рис. 73.

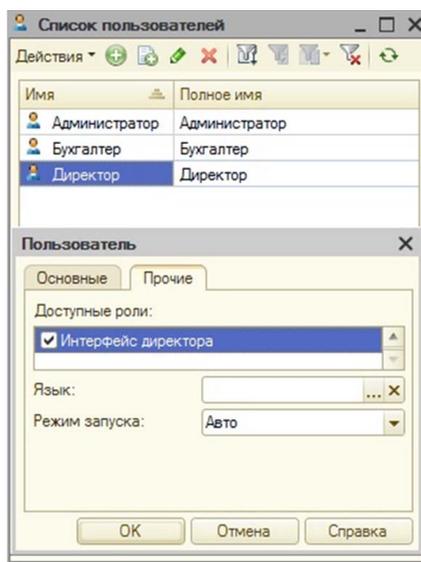


Рис. 73. Назначение роли *ИнтерфейсДиректора* пользователю *Директор*

4. Разместите на рабочем столе *Директора* вывод отчетов на странице *ПросмотрОтчетов*.
5. Напишите программный код для открытия отчета *Отчет-ВыдачаСотруднику* при нажатии на соответствующую ссылку.

Для этого на общей форме *ИнтерфейсДиректора* в свойствах надписи *ОтчетВыдачаСотруднику* выберите событие *Нажатие* и откройте модуль формы, щелкнув по кнопке с лупой. В модуле формы отредактируйте процедуру обработчика события *ОтчетВыдачаСотрудникуНажатие*, как показано на рис. 74.

```

&НаКлиенте
| Процедура ОтчетВыдачаСотрудникуНажатие (Элемент)
|   ПолучитьФорму ("Отчет.ОтчетВыдачаСотруднику.форма") .Открыть ();
| КонецПроцедуры

```

Рис. 74. Программный код обработчика события *ОтчетВыдачаСотрудникуНажатие*

6. Аналогично создайте обработчики событий *ОплатаКомплек-
тующихНажатие* и *ОстаткиКомплектующихНажатие*.

7. Запустите отладку конфигурации и проверьте работоспо-
собность системы, выведите отчеты в интерфейсе пользователя
Директор.

8. Спроектируйте интерфейс для пользователя *Бухгалтер* с вы-
водом документов *Поступление* и *Заказ*.

9. Запустите отладку конфигурации и проверьте работоспособ-
ность системы.

10. Выгрузите информационную базу; имя файла должно быть
следующее: Практическая работа 9.

Практическая работа 10 Запросы в «1С:Предприятие»

Цель работы – формирование практических навыков по напи-
санию и использованию для решения прикладных задач запросов
в «1С:Предприятие».

Задание

В информационной системе «Учет компьютерных комплектую-
щих» осуществите различные виды запросов для решения приклад-
ных задач по выборке данных.

Порядок выполнения работы

1. Добавьте в конфигурацию подсистему *Запросы*.

2. Добавьте в конфигурацию обработку *Запросы*; объект должен быть доступен в подсистеме *Запросы*.

Обработка как объект конфигурации предназначена для выполнения различных действий над информацией.

Как и другие объекты системы «1С:Предприятие», обработка может иметь несколько форм, с помощью которых можно организовать ввод каких-либо параметров, влияющих на ход алгоритма.

Чтобы добавить обработку в дерево конфигурации, выделите объект *Обработки* и в контекстном меню выполните команду **Добавить**. Выполните в диалоговом окне основные настройки: введите имя *Запросы*, на вкладке *Подсистемы* выберите подсистему *Запросы*. На вкладке *Формы* добавьте форму обработки и назначьте ее основной.

3. Добавьте на форму обработки *Запросы* кнопку <КОНСТРУКЦИЯ ВЫБРАТЬ>.

Для этого на вкладке *Команды* редактора форм добавьте команду **КонструкцияВыбрать**. Чтобы на форме появилась соответствующая кнопка, перетащите команду **КонструкцияВыбрать** в левую часть на дерево *Форма*.

Далее откройте палитру свойств команды **КонструкцияВыбрать** и, щелкнув по кнопке с лупой, в поле *Действие* создайте процедуру обработчика события *КонструкцияВыбрать*. Выберите тип *Создать на Клиенте* и *процедуру на сервере*.

4. В модуле формы обработки *Запросы* напишите запрос, с помощью которого получите поля *Наименование*, *Адрес*, *Телефон* для всех записей справочника *Клиенты*.

Чтобы получить не все записи, хранящиеся в таблице объекта справочника *Клиенты*, а только определенные поля из таблицы, создайте запрос, как показано на рис. 75.

В данном запросе после ключевого слова **ВЫБРАТЬ** перечислены полные имена требуемых полей результата запроса. Ключевое слово **ИЗ** помогает задать таблицу, где данные хранятся, — *Справочник.Клиенты*. Ключевое слово **КАК** сокращает путь к данным до псевдонима *Клиенты*.

```

" ВЫБРАТЬ
|   Клиенты.Наименование,
|   Клиенты.Адрес,
|   Клиенты.Телефон
| ИЗ Справочник.Клиенты КАК Клиенты";

```

Рис. 75. Код запроса

Запрос формируется и выполняется из встроенного языка. Для этого предназначены следующие программные объекты:

- *Запрос*;
- *РезультатЗапроса*;
- *ВыборкаИзРезультатаЗапроса*.

Весь программный код процедуры *КонструкцияВыбрать* показан на рис. 76.

```

&НаСервере
□ Процедура КонструкцияВыбратьНаСервере ()
  Запрос = Новый Запрос;
  Запрос.Текст =
    "ВЫБРАТЬ
    |   Клиенты.Наименование,
    |   Клиенты.Адрес,
    |   Клиенты.Телефон
    | ИЗ Справочник.Клиенты КАК Клиенты";

  Вывод = Запрос.Выполнить (). Выбрать ();

  Сообщение = Новый СообщениеПользователю;
  Пока Вывод.Следующий () Цикл
    Сообщение.Текст=Строка (Вывод.Наименование) +", "+
      Строка (Вывод.Адрес) +", "+Строка (Вывод.Телефон) ;
    Сообщение.Сообщить ();
  КонечЦикла;
КонецПроцедуры

&НаКлиенте
□ Процедура КонструкцияВыбрать (Команда)
  КонструкцияВыбратьНаСервере ();
КонецПроцедуры

```

Рис. 76. Код процедуры *КонструкцияВыбрать*

С помощью объекта *СообщениеПользователю* формируется вывод сообщения пользователю после окончания обработки.

В целом организация вывода сообщений отвечает следующим задачам: отражает ход выполнения текущего процесса (показывает стадию выполнения процесса или расчетные значения), выдает сообщение об ошибке, выдает пояснения и рекомендации.

5. Добавьте на обработку *Запросы* кнопку <УПОРЯДОЧИТЬ> и напишите запрос, с помощью которого получите список комплектующих из регистра сведений *ЦеныПродажиКомплектующих* в порядке возрастания цены на продажу.

Для упорядочивания данных в тексте запроса используется конструкция УПОРЯДОЧИТЬ ПО, как показано на рис. 77.

Ключевое слово ВОЗР используется, если необходимо результат запроса вывести в порядке возрастания. Настройка используется по умолчанию. Ключевое слово УБЫВ – результат запроса выводится в порядке убывания.

```

    &НаСервере
    □ Процедура УпорядочитьНаСервере ()
        Запрос = Новый Запрос;
        Запрос.Текст =
            "ВЫБРАТЬ
            | Комплектующие,
            | ЦенаПродажи КАК Цена
            | ИЗ РегистрСведений.ЦеныПродажиКомплектующих
            | УПОРЯДОЧИТЬ ПО Цена ВОЗР";

        Вывод = Запрос.Выполнить().Выбрать();
        Сообщение = Новый СообщениеПользователю;
        Пока Вывод.Следующий() Цикл
            Сообщение.Текст = Вывод.Комплектующие;
            Сообщение.Сообщить();
        КонечЦикла;
    КонечПроцедуры

    &НаКлиенте
    □ Процедура Упорядочить (Команда)
        УпорядочитьНаСервере();
    КонечПроцедуры

```

Рис. 77. Код процедуры *Упорядочить*

6. Добавьте на обработку *Запросы* кнопку <ВЫБРАТЬ ПЕРВЫЕ> и напишите запрос, с помощью которого получите только две записи в списке комплектующих из регистра сведений *ЦеныПродажиКомплектующих* с минимальной ценой на продажу.

В определенных случаях необходимо в большом объеме информации выявлять несколько максимальных или минимальных значений. При решении таких задач при организации запроса в конструкции ВЫБРАТЬ добавьте ключевое слово ПЕРВЫЕ.

Отредактированный текст запроса показан на рис. 78.

```
"ВЫБРАТЬ ПЕРВЫЕ 2
| Комплектующие,
| ЦенаПродажи КАК Цена
| ИЗ РегистрСведений.ЦеныПродажиКомплектующих
| УПОРЯДОЧИТЬ ПО Цена ВОЗР";
```

Рис. 78. Запрос *Выбрать Первые*

7. Добавьте на форму обработки *Запросы* кнопку <ИЕРАРХИЯ> и напишите запрос, с помощью которого выведете из справочника *Комплектующие* наименования комплектующих и поставщиков с выводом групп справочника.

Справочник *Комплектующие* имеет иерархическую структуру.

Запрос с учетом иерархии можно организовать при упорядочивании с использованием ключевого слова ИЕРАРХИЯ. Код запроса показан на рис. 79.

```
"ВЫБРАТЬ
| Наименование, Поставщики
| ИЗ Справочник.Комплектующие
| УПОРЯДОЧИТЬ ПО Наименование ИЕРАРХИЯ";
```

Рис. 79. Запрос с использованием иерархии

8. Добавьте на форму обработки *Запросы* кнопку <УСЛОВИЯ> и напишите запрос, с помощью которого получите список поступлений комплектующих, зафиксированных в документах *Поступление* в количестве больше 10 штук, при условии их суммарной стоимости более 15 000 рублей.

В языке запросов можно использовать конструкцию ГДЕ, если необходимо обработать какое-либо условие. Условия могут быть сложными, в таких случаях используются логические операторы И, ИЛИ, НЕ.

Код запроса показан на рис. 80

```
"ВЫБРАТЬ
| Комплектующие, Количество, Сумма
| ИЗ Документ.Поступление.Товары
| ГДЕ Количество > 10 И Сумма > 15000";
```

Рис. 80. Запрос с использованием условий

9. Добавьте на форму обработки *Запросы* кнопку <ЛОГИЧЕСКАЯ КОНСТРУКЦИЯ> и напишите запрос, с помощью которого получите номер и дату документов *Поступление*, сформированные в 2023 году с мая по октябрь.

Для того чтобы отобразить документы за текущий месяц, в условии отбора сравним поле документа *Дата* с литералом даты ДАТАВРЕМЯ(2023, 11, 01).

Код запроса показан на рис. 81.

```
"ВЫБРАТЬ
| Поступление.Номер,
| Поступление.Дата
|ИЗ Документ.Поступление   КАК Поступление
| ГДЕ Поступление.Дата >= ДатаВремя(2023,05,01)
|           И Поступление.Дата <= ДатаВремя(2023,10,30)";
```

Рис. 81. Запрос с использованием логических конструкций

10. Добавьте на форму обработки *Запросы* кнопку <СРЕДНЕЕ ПО ТОВАРУ> и напишите запрос, с помощью которого определите среднюю сумму каждого товара по всем заказам клиентов.

Для реализации таких запросов необходимо использовать агрегатные функции и группировки. Любая агрегатная функция принимает в качестве аргумента какой-либо столбец и возвращает единственное значение.

Используемые в системе «1С:Предприятие» агрегатные функции:

СУММ — вычисляет сумму всех значений, находящихся в заданном столбце.

МАКСИМУМ — определяет максимальное значение в заданном столбце.

МИНИМУМ — определяет минимальное значение в заданном столбце.

СРЕДНЕЕ — вычисляет среднее арифметическое значение по заданному столбцу.

КОЛИЧЕСТВО — подсчитывает количество значений, находящихся в заданном столбце.

Конструкция **СГРУППИРОВАТЬ ПО** применяется, если необходимо сгруппировать записи по определенному полю. Причем нуж-

но помнить: если осуществляется выборка поля таблицы, но оно не используется агрегатной функцией, то это поле обязательно должно быть включено в группировку.

Код запроса показан на рис. 82.

```
"ВЫБРАТЬ Заказ.Комплектующие,  
| Среднее (Заказ.ВнесеннаяСумма) КАК Среднее  
| ИЗ Документ.Заказ.Оплата КАК Заказ  
| СГРУППИРОВАТЬ ПО Заказ.Комплектующие " ;
```

Рис. 82. Запрос с использованием агрегатной функции с группировкой

11. Добавьте на форму обработки *Запросы* кнопку <ПАРАМЕТР> и напишите запрос, с помощью которого получите из регистра накопления *Заказы* наименования документов, указанная сумма заказов которых не превышает максимальную. Ввод значения максимальной суммы осуществлять в поле на форме.

Возникают ситуации, когда, выполняя определенную задачу, необходимо вводить значения не в тексте запроса, а в диалоге.

Для решения таких задач запрос строится с использованием параметров и в программном коде вводится строка *Запрос.УстановитьПараметр()*.

Метод *УстановитьПараметр()* определяет значение параметра для запроса. При этом первый параметр обозначает имя параметра, второй параметр определяет его значение.

В редакторе форм добавьте команду *Параметр* и перетащите ее в дерево *Формы* для добавления кнопки <ПАРАМЕТР> на форме.

Добавьте реквизит *МаксимальнаяСумма* (тип Число). Перетащите его в дерево *Формы*, добавив тем самым поле ввода на форме. Для лучшего визуального представления сгруппируйте кнопку <ПАРАМЕТРЫ> и поле *МаксимальнаяСумма*, добавив в дерево *Формы* элемент *Группа – Обычная группа*.

На форме обработки должны появиться два элемента, как показано на рис. 83.



ПАРАМЕТР Максимальная сумма: 0

Рис. 83. Фрагмент формы обработки *Запросы*

Код процедуры *Параметр* показан на рис. 84.

```

&НаСервере
□ Процедура ПараметрНаСервере ()
  Запрос = Новый Запрос;
  Запрос.Текст =

    "ВЫБРАТЬ Заказ.Регистратор
    |
    |ИЗ РегистрНакопления.Заказы КАК Заказ
    |ГДЕ Сумма<=&МаксимальнаяСумма ";

  Запрос.УстановитьПараметр ("МаксимальнаяСумма", ЭтаФорма.МаксимальнаяСумма);
  Вывод = Запрос.Выполнить().Выбрать();

  Сообщение = Новый СообщениеПользователю;

  Пока Вывод.Следующий() Цикл
    Сообщение.Текст = Вывод.Регистратор;
    Сообщение.Сообщить();
  КонечЦикла;
КонецПроцедуры

&НаКлиенте
□ Процедура Параметр(Команда)
  ПараметрНаСервере();
КонецПроцедуры

```

Рис. 84. Процедура *Параметр*

В задаче необходимо получить наименования документа, следовательно, в коде используется поле *Регистратор*, где хранятся данные о документах-регистраторах, связанных с регистром *Заказы*.

12. Добавьте на форму обработки *Запросы* кнопку <ИМЕЮЩИЕ> и напишите запрос, с помощью которого получите из регистра накопления *Заказы* наименования документов, итоговая сумма которых не превышает заданного значения. Ввод значения максимальной суммы осуществлять в поле на форме.

Если в запросах применяются агрегатные функции, то ключевое слово *ИМЕЮЩИЕ* позволяет накладывать на их значения условия.

Текст запроса показан на рис. 85.

```

"ВЫБРАТЬ Заказ.Регистратор
|
|ИЗ РегистрНакопления.Заказы КАК Заказ
|СГРУППИРОВАТЬ ПО Заказ.Регистратор
|ИМЕЮЩИЕ
|СУММА (Сумма)<=&Сумма ";

```

Рис. 85. Запрос с использованием *ИМЕЮЩИЕ*

13. Добавьте на форму обработки *Запросы* кнопку *ИТОГИ* и напишите запрос, с помощью которого получите общие итоги по комплекующим в документе *Заказ*.

Вывод итоговой информации – наиболее востребованная функция информационной системы. Система «1С:Предприятие» имеет в своем арсенале большое количество средств, позволяющих проводить расчет итогов. В языке запросов есть ключевое слово ИТОГИ, которое в результате запроса добавляет отдельные строки, содержащие общие и промежуточные итоги по заданным полям и группировкам.

Код процедуры ИТОГИ показан на рис. 86.

```

        <НаСервере
    [ Процедура ИтогиНаСервере ()
        Запрос = Новый Запрос;
        Запрос.Текст =

            "ВЫБРАТЬ
            | Заказ.Комплектующие, Заказ.ВнесеннаяСумма
            | ИЗ Документ.Заказ.Оплата КАК Заказ
            | ИТОГИ СУММА (ВнесеннаяСумма) ПО ОБЩИЕ ";

        Вывод = Запрос.Выполнить ().Выбрать ();

        Сообщение = Новый СообщениеПользователю;

        Пока Вывод.Следующий () Цикл
            Сообщение.Текст = Строка (Вывод.Комплектующие) + ", " +
                Строка (Вывод.ВнесеннаяСумма) ;
            Сообщение.Сообщить ();
        КонечЦикла;
    КонечПроцедуры

        <НаКлиенте
    [ Процедура Итоги (Команда)
        ИтогиНаСервере ();
    КонечПроцедуры
    
```

Рис. 86. Процедура ИТОГИ

14. Добавьте на форму обработки *Запросы* кнопку <ИТОГИ ПО ГРУППИРОВКАМ> и напишите запрос, с помощью которого получите итоги по комплектующим в документе *Заказ*.

Итоги можно получить и по группировкам. В таких случаях вычисляются значения агрегатных функций по строкам выборки с одинаковыми значениями полей, по которым выполняется группировка. Текст запроса показан на рис. 87.

```
"ВЫБРАТЬ
| Заказ.Комплектующие, Заказ.ВнесеннаяСумма
|ИЗ Документ.Заказ.Оплата КАК Заказ
|ИТОГИ СУММА (ВнесеннаяСумма) ПО Заказ.Комплектующие ";
```

Рис. 87. Текст запроса, включающий итоги по группировкам

15. Добавьте на форму обработки *Запросы* кнопку <ВЛОЖЕННЫЙ ЗАПРОС> и напишите запрос, с помощью которого выведете список поставщиков, комплектующие которых были заказаны клиентами в документе *Заказ*.

Возникают ситуации, когда необходимо выполнить запрос, который, в свою очередь, выполняется на результате другого запроса.

При выполнении внешнего запроса сначала выполняется вложенный запрос, и результат его выполнения подставляется в условие отбора внешнего запроса.

Текст запроса показан на рис. 88.

```
"ВЫБРАТЬ
| Поступление.Поставщик
|ИЗ Документ.Поступление КАК Поступление
|ГДЕ Поступление.Товары.Комплектующие В
| (
|     ВЫБРАТЬ Заказ.Комплектующие
|     ИЗ Документ.Заказ.Оплата КАК Заказ
| )";
```

Рис. 88. Текст запроса с использованием вложенного запроса

В условии отбора внешнего запроса *Поступление.Товары.Комплектующие В ()* значение поля *Комплектующие* из табличной части *Товары* документа *Поступление* проверяется на попадание в перечень возможных значений.

Для получения этого набора значений используется вложенный запрос, который выбирает *Комплектующие* из табличной части *Оплата* документа *Заказ*.

16. Добавьте на форму обработки *Запросы* кнопку <ЛЕВОЕ СОЕДИНЕНИЕ> и напишите запрос, с помощью которого выведете список комплектующих, их поставщиков и установленную в регистре сведений цену продажи на соответствующую позицию.

Существуют ситуации, когда необходимо выводить данные, полученные из нескольких таблиц одновременно, объединяя их в одну общую таблицу.

Это выполняется путем сравнения данных одной таблицы со строками другой таблицы. Чтобы произвести соединение таблиц в запросе, необходимо прописать условия и типы тех полей, которые нужно связать друг с другом.

В системе «1С:Предприятие» можно выполнить четыре вида соединений.

При использовании левого соединения в результате отображаются все записи из левой таблицы с добавлением записей правой таблицы, для которых выполняются условия связи.

В задаче используются две таблицы – справочник *Комплектующие* и регистр сведений *ЦеныПродажиКомплектующих*. При применении левого соединения в результате будут выведены все записи справочника *Комплектующие* и записи из регистра сведений *ЦеныПродажиКомплектующих*, которые удовлетворяют условию связи – одно и то же комплектующее.

Код процедуры показан на рис. 89.

```

        §НаСервере
    □ Процедура СоединениеЛевоеНаСервере ()
        Запрос = Новый Запрос;
        Запрос.Текст =

            "ВЫБРАТЬ
            | Комплект.Наименование,
            | Комплект.Поставщики,
            | Цены.ЦенаПродажи
            |
            | ИЗ Справочник.Комплектующие КАК Комплект
            | ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ЦеныПродажиКомплектующих КАК Цены
            | ПО Комплект.Ссылка = Цены.Комплектующие ";
        Вывод = Запрос.Выполнить ().Выбрать ();

        Сообщение = Новый СообщениеПользователю;

        Пока Вывод.Следующий () Цикл
            Сообщение.Текст = Строка (Вывод.Наименование) + ", " +
                Строка (Вывод.Поставщики) + ", " + Строка (Вывод.ЦенаПродажи);
            Сообщение.Сообщить ();
        КонечЦикла;
    КонечПроцедуры

    §НаКлиенте
    □ Процедура СоединениеЛевое (Команда)
        СоединениеЛевоеНаСервере ();
    КонечПроцедуры
    
```

Рис. 89. Код процедуры СоединениеЛевое

17. Проанализируйте полученный результат, добавив одну позицию в справочник *Комплектующие*.

18. Добавьте на форму обработки *Запросы* кнопку <ПРАВОЕ СОЕДИНЕНИЕ> и напишите запрос, с помощью которого выведете список комплектующих, их поставщиков и установленную в регистре сведений цену продажи на соответствующую позицию. Сравните с результатом пунктов 16 и 17.

При использовании правого соединения в результате отображаются все записи из правой таблицы, которая теперь считается главной, и к ним добавляются записи левой таблицы, для которых выполняются условия связи.

19. Добавьте на форму обработки *Запросы* кнопку <ПОЛНОЕ СОЕДИНЕНИЕ> и напишите запрос, с помощью которого выведете список комплектующих, их поставщиков и установленную в регистре сведений цену продажи на соответствующую позицию.

При использовании полного соединения в результате отображаются все записи как из первой, так и из второй таблицы, независимо от выполнения условия связи.

20. Сравните результат с предыдущими задачами на соединение.

21 Выгрузите информационную базу, выполнив в конфигураторе команду **Администрирование – Выгрузить информационную базу**; имя файла должно быть следующее: Практическая работа 10.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Трофимова, М. В. Предметно-ориентированные информационные системы : учеб. пособие / М. В. Трофимова. — Ставрополь : Издательство Северо-Кавказского федерального университета, 2014. — 187 с.
2. Вдовин, В. М. Предметно-ориентированные экономические информационные системы : учеб. пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. — 3-е изд. — Москва : Дашков и К, 2016. — 385 с. — ISBN 978-5-394-02262-3.
3. Радченко, М. Г. Архитектура и работа с данными «1С:Предприятия 8.2» / М. Г. Радченко, Е. Ю. Хрусталева. — Москва : 1С-Пабблишинг, 2011. — 268 с. — (1С. Профессиональная разработка). — ISBN 978-5-9677-1555-6.
4. Гладких, Т. В. Разработка прикладных решений для информационной системы 1С: Предприятие 8.2 : учеб. пособие / Т. В. Гладких, Е. В. Воронова. — Воронеж : Воронежский государственный университет инженерных технологий, 2016. — 55 с. — URL: www.iprbookshop.ru/50639.html (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-00032-182-9.
5. Скороход, С. В. Программирование на платформе 1С: Предприятие 8.3 : учеб. пособие / С. В. Скороход. — Ростов-на-Дону [и др.] : Издательство Южного федерального университета, 2019. — 133, [1]с. — URL: znanium.com/catalog/document?id=357443 (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-9275-3315-2.
6. Разработка управляемого интерфейса / В. А. Ажеронков, А. В. Островерх, М. Г. Радченко, Е. Ю. Хрусталева. — Москва : 1С-Пабблишинг, 2010. — 723 с. — ISBN 978-5-9677-1148-0.
7. Хрусталева, Е. Ю. Язык запросов «1С:Предприятия 8» / Е. Ю. Хрусталева. — Москва : 1С-Пабблишинг, 2013. — 369 с. — ISBN 978-5-9677-1987-5.
8. Заика, А. А. Разработка прикладных решений для платформы 1С:Предприятие 8.2 в режиме «Управляемое приложение» : учеб. пособие / А. А. Заика. — 3-е изд.(электрон.). — Москва : Интернет-Университет Информационных Технологий [и др.], 2021. —

- 238 с. — URL: www.iprbookshop.ru/73721.html (дата обращения: 05.11.2023). — Режим доступа: по подписке.
9. Основы конфигурирования в системе «1С:Предприятие 8.0»: учеб. пособие. — 3-е изд. (электрон.). — Москва : Интернет-Университет Информационных Технологий [и др.], 2021. — 222 с. — URL: www.iprbookshop.ru/102027.html (дата обращения: 05.11.2023). — Режим доступа: по подписке. — ISBN 978-5-4497-0876-2.
 10. Разработка предметно-ориентированных систем. Учебник. Часть 1 // topuch.com : [сайт]. — URL: topuch.com/uchebnika-razrabotka-predmetno-orientirovannih-sistem/index.html (дата обращения: 26.10.2023).
 11. Азарова, А. А. Электронная цифровая подпись как средство защиты информационной модели предприятия / А. А. Азарова, К. В. Ивчук, М. И. Кукуруза. — EDN SJNBHB // Экономика, социология и право. — 2014. — № 2. — С. 7–9.
 12. Бочарников, В. Н. Информационные технологии в туризме : учеб. пособие / В. Н. Бочарников, Е. Г. Лаврушина, Я. Ю. Блиновская. — Москва : Флинта, 2008. — 356, [1] с. — ISBN 978-5-9765-0251-2.
 13. Информационно-технологическое сопровождение программ «1С:Предприятие» // 1С : [сайт]. — URL: 1c.ru/rus/support/its/rodpiska/its.htm (дата обращения: 27.10.2023).
 14. Записки программиста 1С // 1С help : It Help You : [сайт] / В. Ганзюк. — URL: ithelpyou.ru/2021/12/13/универсальные-коллекции-значений (дата обращения: 27.10.2023).
 15. ДЛЯ ИЗМЕНЕНИЯ в запросах 1С 8 // 1С-Архитектор бизнеса : [сайт]. — URL: programmist1s.ru/dlya-izmeneniya-v-zaprosah-1s/ (дата обращения: 27.10.2023).

ГЛОССАРИЙ

Документ — прикладной объект конфигурации, который вводится в систему для фиксации каких-то событий, которые происходят в организации.

Запрос — основной способ получения данных из информационной базы.

Клиентское приложение (или клиент) — это программа, работающая на устройстве пользователя.

Клиентский контекст — код выполняется на локальной машине клиента.

Константа — объект метаданных, который нужен для хранения постоянной и условно-постоянной информации.

Конфигурация — совокупность взаимосвязанных объектов с определенной структурой информационных массивов и набор описанных алгоритмов обработки данных.

Массив — это объект, который представляет собой совокупность значений произвольного типа, включая и массивы.

Организация — это стабильная формальная социальная структура, которая получает ресурсы из окружающего мира и перерабатывает их в продукты своей деятельности.

Отчет — объект, предназначенный для представления алгоритмов и инструментов получения выходных данных.

Переменная — это временная сущность, необходимая для временного хранения информации.

Подсистема — объект, предназначение которого в формировании пользовательского интерфейса путем группировки объектов конфигурации по определенным правилам.

Предметная область — определенная часть деятельности человека, его реального мира, для решения задач которой предназначена информационная система, в том числе для их автоматизации.

Процедура — это фрагмент кода с определенным набором действий.

Регистр — объект, предназначенный для хранения информации о состояниях и количествах объектов базы данных.

Серверный контекст — код выполняется на машине кластера серверов «1С:Предприятие».

Справочник — объект, предназначенный для хранения постоянной информации, которая может иметь множество значений.

Структура — это коллекция значений, позволяющая строить динамический набор данных, каждый элемент которой формируется по правилу <ключ — значение>.

Таблица значений — объект, который позволяет строить динамические наборы значений и манипулировать ими.

Технологическая платформа — совокупность взаимосвязанных составных частей и специальных механизмов, которые обеспечивают целостное функционирование системы «1С:Предприятие» и решают учетные задачи в определенной предметной области.

Универсальные коллекции значений — это объекты, предназначенные для хранения временной информации.

Функция — это тоже фрагмент кода с определенным набором действий, но функция возвращает определенное значение.