

**Автономная некоммерческая организация
профессионального образования
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»
(АНО ПО «ПГТК»)**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
МЕЖДИСЦИПЛИНАРНОГО КУРСА
МДК.01.02 Прикладное программирование**

для студентов специальности

09.02.03 Программирование в компьютерных системах
(код и наименование специальности)

Квалификация выпускника

Техник-программист
(базовая подготовка)

Форма обучения

Очная

Пермь, 2019 г

Методические рекомендации по изучению междисциплинарного курса МДК.01.02 Прикладное программирование предназначены для студентов и преподавателей АНО ПО «ПГТК». Методические указания определяют ориентиры и способствуют более обстоятельному усвоению программного материала, организации самостоятельного процесса изучения учебного предмета обучающимися по специальности Программирование в компьютерных системах.

Данные методические рекомендации помогут организовать самостоятельную деятельность студентов на основе деятельного и компетентного подходов к обучению, что соответствует ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах.

Автор-составитель: Тимохова Н.А., ст. преподаватель

Утверждено на заседании кафедры математических и естественно-научных дисциплин, протокол № 6 от «21» января 2019 г.

Рекомендованы к утверждению педагогическим советом АНО ПО «ПГТК» (протокол от «5» февраля 2019 г. № 3).

Оглавление	
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
КОНТРОЛЬНЫЕ ВОПРОСЫ.....	21
Список рекомендованной литературы.	21
Интернет-источники:	22

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания по выполнению практических работ дисциплины «Прикладное программирование» предназначены для студентов специальности 09.02.03 «Программирование в компьютерных системах». Основной целью методической разработки является повышение профессионального образования, обеспечение глубокой и специализированной подготовки студентов по данному предмету.

В результате изучения профессионального модуля обучающийся должен:

иметь практический опыт:

- разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;
- разработки кода программного продукта на основе готовой спецификации на уровне модуля;
- использования инструментальных средств на этапе отладки программного продукта;
- проведения тестирования программного модуля по определенному сценарию;
- уметь:
- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- оформлять документацию на программные средства;
- использовать инструментальные средства для автоматизации оформления документации;
- знать:
- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные принципы отладки и тестирования программных продуктов;
- методы и средства разработки технической документации

Часть 2 Документы

Лабораторная работа № 15. Документ «заказ»

Лабораторная работа № 16. Программное создание и запись документов

Лабораторная работа № 17. Поиск документов

Лабораторная работа № 18. Включение в условие отбора параметра

Лабораторная работа № 19. Упорядочивание

Лабораторная работа № 20. Программная установка свойств документов

Лабораторная работа № 21. Заполнение документа на основании имеющихся данных

Лабораторная работа № 22. Обработка событий

Лабораторная работа № 15. Документ «Заказ»

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

В этой и последующих работах основное внимание будет уделено документам. Объект конфигурации "документ" является прикладным и предназначен для фиксации информации о событиях, происходящих в организации. Ранее один из примеров был связан с разработкой в конфигураторе документа, фиксирующего денежные поступления от клиентов. Мы познакомились с электронной формой документа, а также с его печатным вариантом (печатной формой). С практической точки зрения документ можно считать прикладным объектом, назначением которого является ввод и фиксация информации о событиях, происходящих в организации. Эти события, как правило, влияют на количественные показатели в рассматриваемой предметной области (чаще всего учет связан с движением денежных средств и материальных ценностей). В качестве примеров можно предложить факт выписки денежных сумм сотрудникам, передачу товаров и материалов с одного склада на другой и т. д. Именно для того чтобы отражать подобные факты в системе учета, объект конфигурации "документ" и был разработан. События, происходящие в организации, отличаются привязкой ко времени, в связи с чем параметр, характеризующий точное время, является обязательным реквизитом любого документа. Мы уже видели, что действия при разработке прикладных документов начинаются в конфигураторе с создания структуры документа — набора полей с указанием их типов данных. Кроме того, с помощью программных процедур реализуются необходимые действия с документом. Результатом работы в конфигураторе является формирование информационной структуры, которая позволяет пользователю в режиме 1С:Предприятие с такими документами работать (создавать, вносить в них информацию, изменять, удалять). Основной чертой, отличающей документы от других объектов, является их способность проведения (возможность формировать проводки, являющиеся результатом информации, присущей документам). Это очень важная особенность, и ее необходимо пояснить. В информационной системе 1С:Предприятие имеются разнообразные структуры информации, которые описывают количественное состояние дел в организации (например, стоимостный и количественный учет разнообразных товаров). Проведение документов означает факт внесения изменений в эти структуры данных (причина изменений связана с информацией, содержащейся в документах). Например, поступление по документу определенного товара изменяет суммарное количество данного товара на складе. До тех пор, пока документ не проведен, состояние учета неизменно. В этом случае функциональность документа не реализована, но сам он, тем не менее, зарегистрирован в базе данных. С другой стороны, в информационной системе при необходимости должна существовать и возможность отмены проведения (возвращения информации в первоначальное состояние). Например, при отмене проведения документа "Поступление товара" должна быть скорректирована информация в регистре, который используется для учета количества товаров на складах. Подобная возможность отмены проведения документов в системе 1С:Предприятие также реализована. Важно подчеркнуть, что документы, разработанные для различных стандартных конфигураций системы 1С:Предприятие, являются аналогами реальных документов, которые используются в работе организаций. Многочисленные документы, используемые в реальном документообороте, различны по структуре. Например, основное содержание одних документов могут составлять таблицы (скажем, списки номенклатуры, услуг или сотрудников), а в других они могут отсутствовать. Существуют также документы с наличием нескольких табличных частей (например, одна табличная часть отводится для списка сотрудников, работающих на постоянной основе, а другая — для списка совместителей). В связи с этим объект конфигурации "документ" системы 1С:Предприятие должен обладать хорошими возможностями конструирования. Использование рассматриваемого объекта конфигурации при разработке прикладных решений должно позволять легко создавать различные документы — приходную накладную, счет или документ о перемещении товаров со склада на склад. В системе

1С:Предприятие 8.1 ситуация выглядит именно так — все конструктивные возможности во власти разработчика.

Опишем теперь основные параметры рассматриваемого объекта конфигурации. В любом документе (созданном в системе 1С:Предприятие, а также в реальных документах) присуствует информация, которая характеризует его в целом. Это дата создания и порядковый номер (в последовательности документов одного типа).

Поскольку документ вносит изменения в состояние учета, он всегда привязан к конкретному моменту времени.

Кроме того, в любом документе присуствует и общая информация другого плана. Например, организация, подразделение или ответственное лицо. Для подобных параметров в объекте конфигурации "документ" отводится область, называемая шапкой. Ее фактически составляют реквизиты, характеризующие документ в целом. Кроме общей области, документ может содержать список данных, схожих по структуре. Например, список сотрудников, организаций или товаров. Каждый из этих списков включает в себя еще и уточняющую и дополнительную информацию. Например, вместе с названием товара указываются количество и цена. Для отражения такой информации в разрабатываемый объект конфигурации следует включить один или несколько подчиненных (внутренних) объектов, которые называются табличными частями. Теперь мы рассмотрим практические приемы программирования документов. При этом мы предварительно разработаем новую конфигурацию, включающую необходимые объекты.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Первым объектом конфигурации, который мы создадим, будет справочник *Фирмы*. Дополнительные реквизиты в этом справочнике нам не потребуются, поэтому перейдем в режим 1С:Предприятие и внесем в него несколько записей (рисунок 2.1). Также создадим справочник Номенклатура, где будут перечислены названия наших товаров (рисунок 2.2).

Разработаем документ Заказ

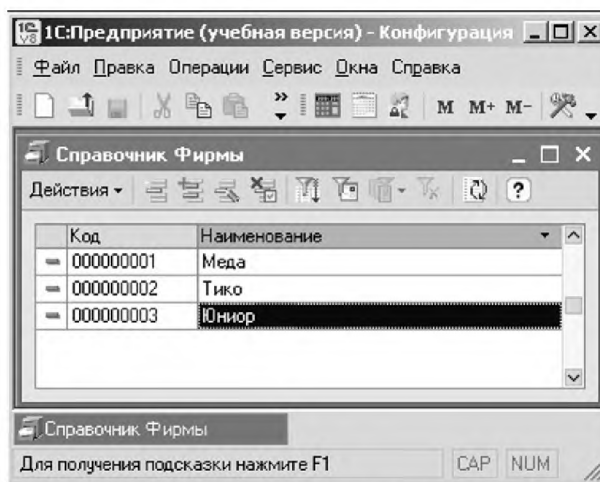


Рисунок 2.1 Заполнение справочника Фирмы

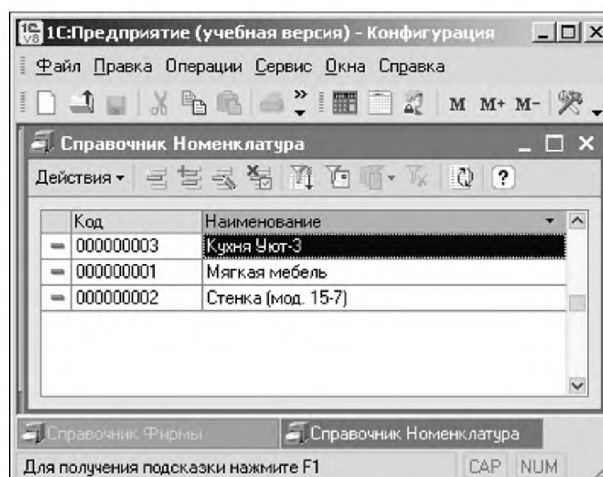


Рисунок 2.2 Внесение данных в справочник Номенклатура

На рисунок 2.3 показана закладка Данные окна редактирования объекта конфигурации. Видно, что область шапки документа содержит один дополнительный реквизит — Заказчик (тип данных — СправочникСсылка.Фирмы). Для табличной части документа выберем имя Перечень и внесем в нее два реквизита:

- Товар (тип данных — СправочникСсылка.Номенклатура);
- Количество (тип данных — Число без дробной части) для внесения количества заказанных изделий.

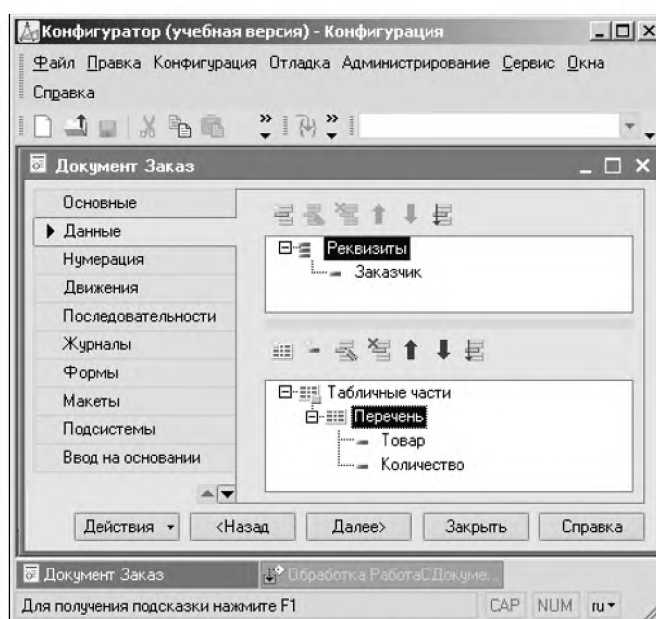


Рисунок 2.3 Структура документа Заказ

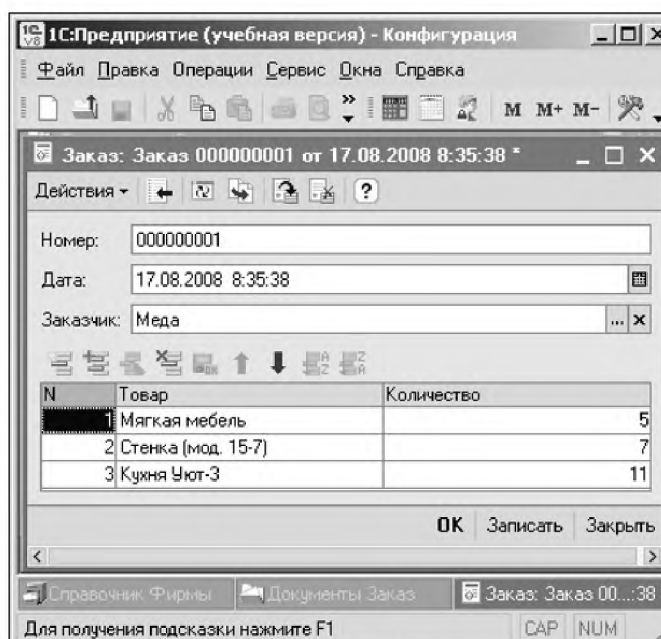


Рисунок 2.4 Заполненный документ Заказ

Документ Заказ будет фиксировать в базе данных заказы товаров нашими партнерами. Все необходимые действия в конфигураторе выполнены, поэтому перейдем в режим 1С:Предприятие и заполним несколько документов Заказ. Один из них показан на рисунок 2.4.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как с помощью встроенного языка вывести в табличный документ новую область?
2. Как изменить внешний вид и поведение элемента управления, расположенного в форме?
3. Как отобразить сумму по колонке табличного поля?

ЛАБОРАТОРНАЯ РАБОТА № 16. Программное создание и запись документов

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Для интерфейса разработки мы создадим обработку. Данный объект конфигурации достаточно часто используется для реализации различных алгоритмов обработки данных

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

В уже знакомом нам окне конфигурации щелкнем правой кнопкой мыши по пиктограмме Обработки и в открывшемся контекстном меню воспользуемся пунктом Добавить. Далее в окне редактирования объекта конфигурации дадим обработке имя РаботаСДокументами (рисунок 2.5).

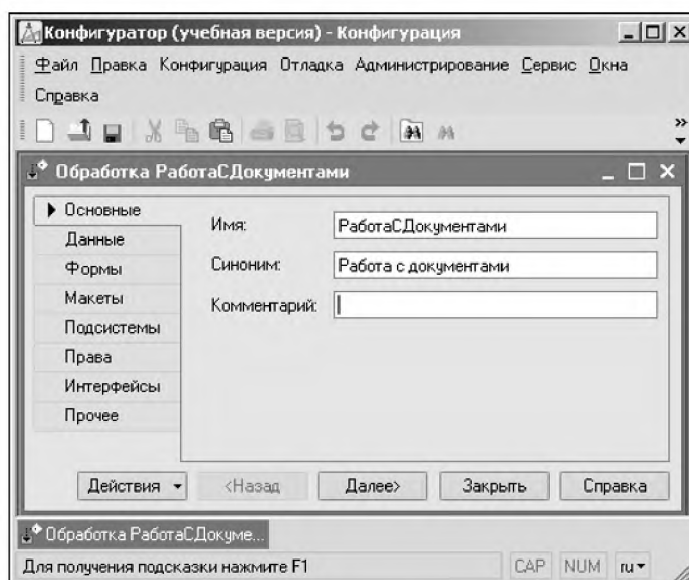


Рисунок 2.5 Окно редактирования обработки РаботаСДокументами

В этой обработке нам потребуется подчиненный объект конфигурации — форма, на которой будет размещено несколько элементов управления. Для ее создания перейдем на закладку **Формы**. Здесь с помощью соответствующей пиктограммы (**Добавить**) начнем разработку формы. В диалоге с конструктором (рисунок 2.6) введем имя формы — **ФормаДляТеста**. В результате уже знакомых действий мы получим экранную форму (рисунок 2.7) с двумя стандартными кнопками: **Выполнить** и **Закреть**.

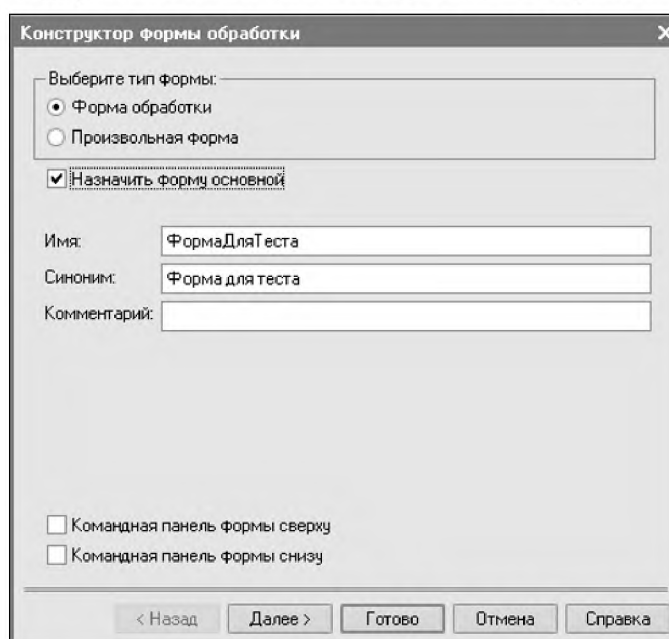


Рисунок 2.6 Окно конструктора формы обработки

Теперь добавим на форму элемент управления — поле ввода. Для этого в меню **Форма** воспользуемся пунктом **Вставить элемент управления**. Выберем **Поле ввода**, в качестве имени укажем **ФирмаЗаказчик**, а в качестве типа данных — **СправочникСсылка.Фирмы** (рисунок 2.8). Все основные технические действия по разработке произойдут на закладке **Модуль**, предназначенной для размещения текстов программных процедур на встроенном языке. На этой закладке следует расположить процедуру обработки нажатия на кнопку **Выполнить**. Ее текст приведен в листинге 2.1. Здесь мы воспользовались методом **СоздатьДокумент()** для программного создания документа, что аналогично интерактивным действиям. Встроенная функция

ТекущаяДата() выдает текущую (системную) дату на компьютере, которую мы заносим в качестве даты создаваемого документа.

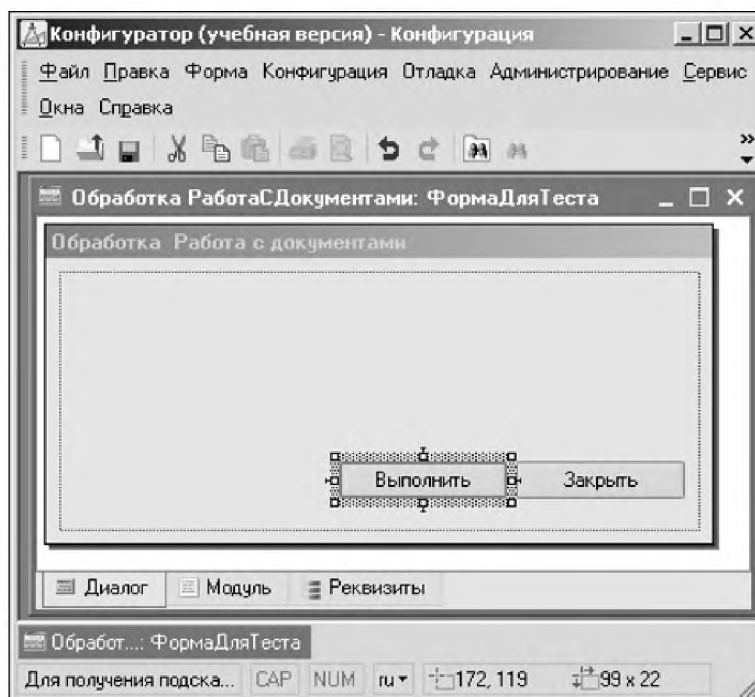


Рисунок 2.7 Окно формы обработки РаботаСДокументами в режиме конфигуратора

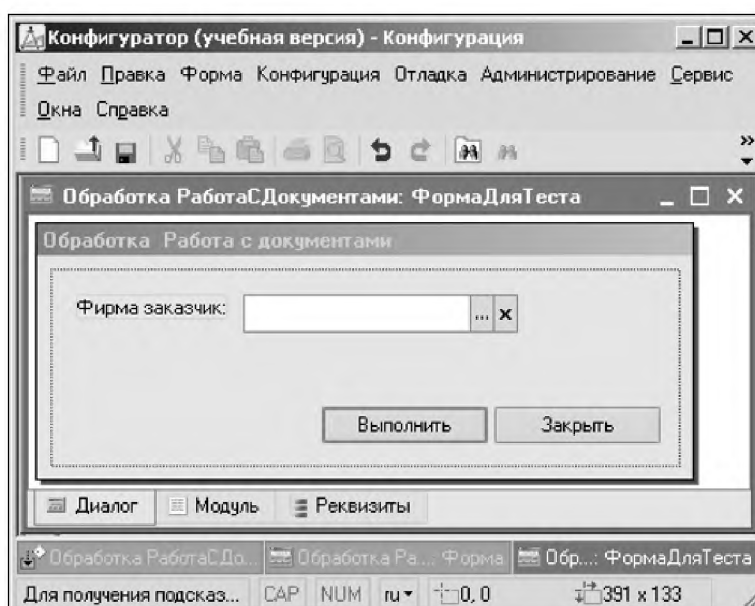


Рисунок 2.8 Добавление в форму обработки поля ввода

После создания нового документа его следует сохранить в базе данных. Сохраняется документ с помощью метода Записать().

Листинг 2.1. Процедура для создания нового документа

Процедура КнопкаВыполнитьНажатие(Кнопка)

НовыйЗаказ = Документы.Заказ.СоздатьДокумент();

НовыйЗаказ.Дата = ТекущаяДата();

НовыйЗаказ.Заказчик = ФирмаЗаказчик;
НовыйЗаказ.Записать();
КонецПроцедуры

После размещения процедуры можно перейти в режим 1С:Предприятие и в меню Операции воспользоваться пунктом Обработка. Здесь следует выбрать только что созданную обработку Работа с документами и открыть ее форму. Теперь нужно указать в поле ввода одну из фирм и нажать кнопку Выполнить. В результате в списке документов Заказ появится новый документ с внесенными датой и фирмой. Заметим, что табличная часть заказа пуста.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего может понадобиться проведение документа по нескольким регистрам?
2. Как с помощью конструктора создать движения документа по нескольким регистрам?
3. Исходя из каких соображений, конструктор формирует текст процедуры проведения по нескольким регистрам?

ЛАБОРАТОРНАЯ РАБОТА № 17. Поиск документов

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Все введенные документы, как мы уже видели, отражаются в форме списка. Однако часто требуется из многочисленного количества документов отобрать только определенную категорию документов по какому-либо принципу. Продемонстрируем, как это сделать с помощью программных ресурсов (система 1С:Предприятие предоставляет широкие возможности по поиску и отбору отдельных документов).

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Рассмотрим ситуацию, когда требуется обеспечить отбор документов по дате. Здесь нам потребуется форма с новым набором элементов управления. Для практической работы можно воспользоваться несколькими вариантами:

- создать новую обработку и разработать форму ввода;
- скорректировать имеющуюся форму обработки РаботаСДокументами;
- создать новую форму в обработке РаботаСДокументами.

Воспользуемся первым вариантом и создадим обработку ВыборкаДокументов. На закладке Формы окна редактирования объекта конфигурации сформируем форму, на которой разместим набор элементов (рисунок 2.9):

- поле ввода Дата1 для ввода начальной даты (тип данных — Дата);
- поле ввода Дата2 для ввода конечной даты (тип данных — Дата);
- поле списка ПереченьДокументов.

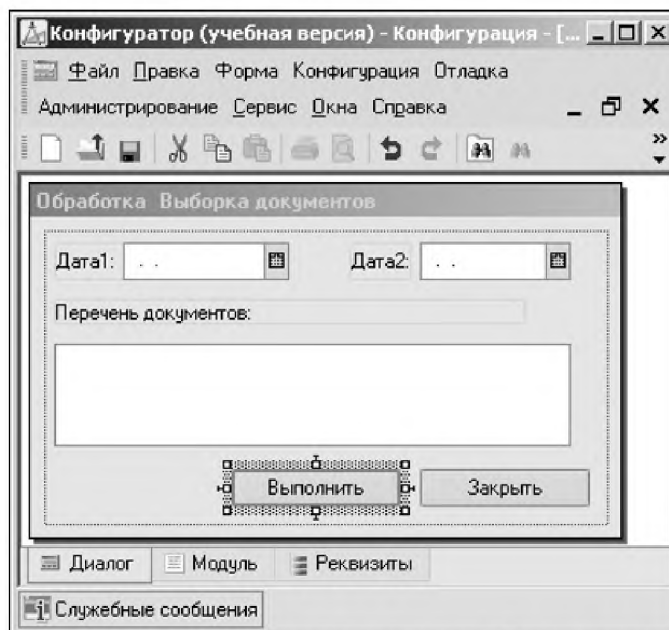


Рисунок 2.9 Форма для организации отбора документов

Теперь на закладке Модуль оформим процедуру обработки нажатия на кнопку Выполнить, которая представлена в листинге 2.2.

Листинг 2.2. Процедура отбора документов по дате

```

Процедура КнопкаВыполнитьНажатие(Кнопка)
НачальнаяДата = НачалоДня(Дата1);
КонечнаяДата = КонецДня(Дата2);
ВыборкаЗаказ = Документы.Заказ.Выбрать(НачальнаяДата, КонечнаяДата);
ПереченьДокументов.Очистить();
Пока ВыборкаЗаказ.Следующий() Цикл
    Информация = "Номер документа: " + ВыборкаЗаказ.Номер +
        " Дата документа: " + ВыборкаЗаказ.Дата;
    ПереченьДокументов.Добавить(Информация);
КонецЦикла;
КонецПроцедуры
    
```

Здесь поля Дата1 и Дата2 используются для указания, соответственно, начала и окончания интересующего нас интервала по дате. Для того чтобы получить документы, попадающие в интервал, мы воспользовались двумя встроенными функциями:

- НачалоДня(), которая возвращает дату и время начала дня для указанной даты;
- КонецДня(), которая определяет дату и времени конца дня для указанной даты.

Отбор найденных документов отражается в поле списка. Для этого мы его предварительно очищаем, вызывая метод Очистить(), а затем с помощью метода Выбрать() производим выборку документов в диапазоне указанных дат. Первый параметр метода задает начальную дату поиска, а второй — конечную. После этого организуется цикл обхода отобранных записей с помощью метода Следующий(). Теперь, если перейти в режим 1С:Предприятие, открыть форму обработки и установить даты, мы получим список отобранных документов (рисунок 2.10).

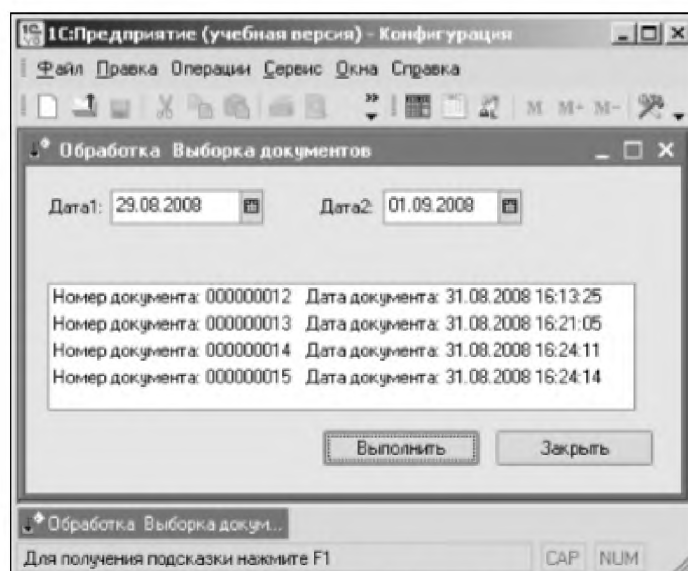


Рисунок 2.10 Результат отбора документов по дате

Если в методе Выбрать() не указывать интервал по дате (это необязательные аргументы), то из базы данных будут выбраны все документы данного вида.

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Как система 1С:Предприятие выполняет обращение к ссылочным данным?
- Как используется кэш объектов?

ЛАБОРАТОРНАЯ РАБОТА № 18. Включение в условие отбора параметра

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Часто требуется выбирать не все записи из базы данных, а лишь те, которые имеют конкретное значение в определенном поле (например, заказы конкретной фирмы). В этом разделе мы продемонстрируем типичные действия в данной ситуации.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

На рисунок 2.11 приведена форма с элементами, которые нам потребуются.

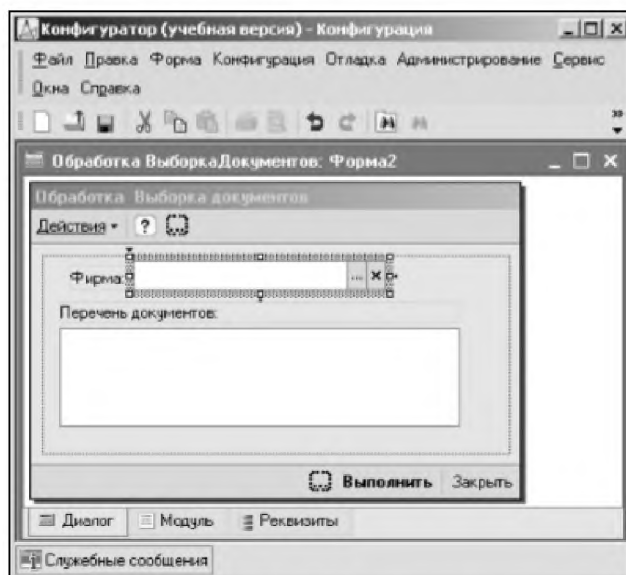


Рисунок 2.11 Форма для отбора документов по фирме

Технически в обработке ВыборкаДокументов мы создадим новую форму. Расположим на ней следующие элементы управления:

- поле ввода Фирма (тип данных — СправочникСсылка.Фирмы);
- поле списка ПереченьДокументов.

Теперь для кнопки Выполнить следует создать процедуру обработки в соответствии с листингом 2.3.

Листинг 2.3. Процедура отбора документов по фирме

Процедура КнопкаВыполнитьНажатие(Кнопка)

Условие = Новый Структура("Заказчик", Фирма);

ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие);

ПереченьДокументов.Очистить();

Пока ВыборкаДокументов.Следующий() Цикл

Информация = "Номер документа: " + ВыборкаДокументов.Номер +
" Дата: " + ВыборкаДокументов.Дата;

ПереченьДокументов.Добавить(Информация);

КонецЦикла;

КонецПроцедуры

В тексте листинга 2.3 для передачи параметра мы использовали объект Структура. В данном случае нас не интересует интервал дат при выборе документов из базы данных. Поэтому первые два аргумента функции оставлены пустыми, а условие реализовано в третьем аргументе:

ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие);

На рисунок 2.12 продемонстрирован результат выбора документов из базы данных по указанному условию.

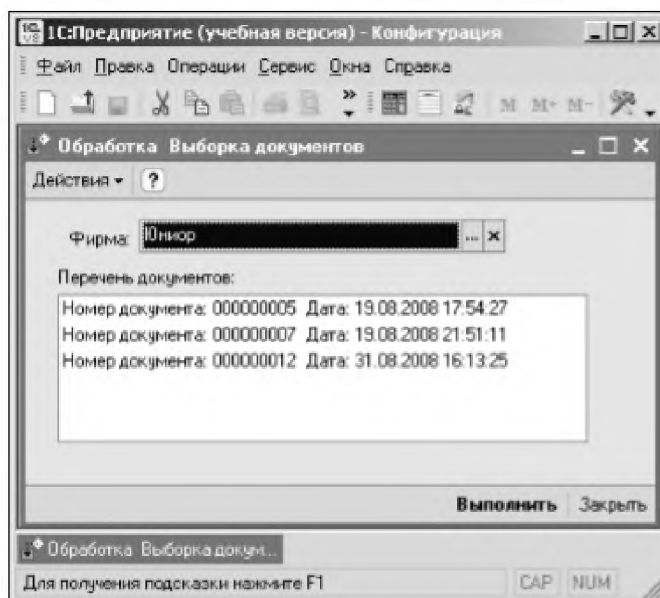


Рисунок 2.12 Пример отбора документов по фирме

КОНТРОЛЬНЫЕ ВОПРОСЫ

- почему для доступа к массивам данных информационной базы предпочтительнее использовать запросы?
- как запросом получить остатки регистра накопления?

ЛАБОРАТОРНАЯ РАБОТА № 19. Упорядочивание

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Несколько изменим предыдущий пример. Будем считать, что нам требуется дополнительно упорядочить результат отбора по дате в порядке возрастания.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Для этого в методе Выбрать() предназначен четвертый аргумент. В листинге 2.4 представлена процедура, позволяющая отобрать документы, относящиеся к нужной фирме, и упорядочить их по дате.

Листинг 2.4. Процедура отбора документов с упорядочиванием по дате

Процедура КнопкаВыполнитьНажатие(Кнопка)

Условие = Новый Структура("Заказчик", Фирма);

Упорядочивание = "Дата Возр";

ВыборкаДокументов = Документы.Заказ.Выбрать(, , Условие, Упорядочивание);

ПереченьДокументов.Очистить();

Пока ВыборкаДокументов.Следующий() Цикл

112 Глава 2

Информация = "Номер документа: " + ВыборкаДокументов.Номер +

" Дата: " + ВыборкаДокументов.Дата;

ПереченьДокументов.Добавить(Информация);

КонецЦикла;

КонецПроцедуры

В режиме 1С:Предприятие результат выполнения данной процедуры показан на рисунок 2.13.

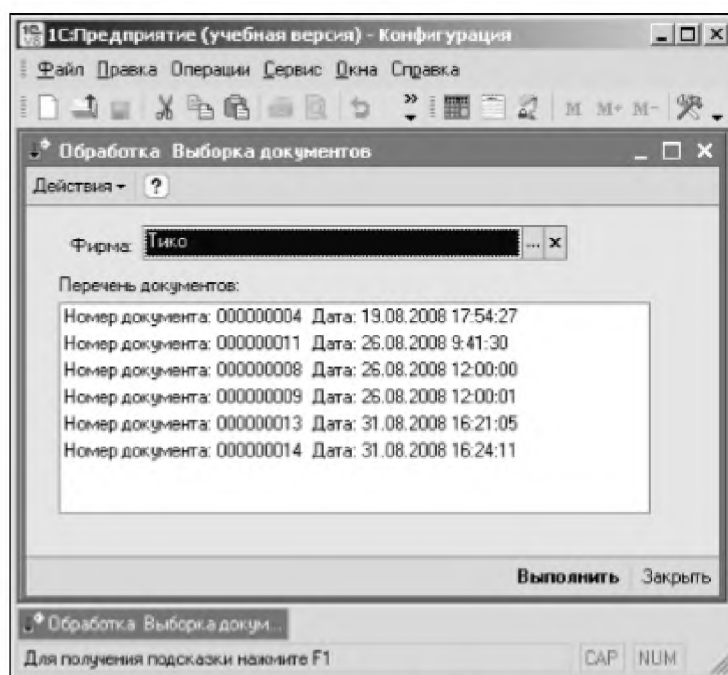


Рисунок 2.13 Форма для отбора документов по фирме с упорядочиванием по дате

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Когда следует использовать ключевое предложение **ДЛЯ ИЗМЕНЕНИЯ**?
- На что следует обращать внимание при задании параметров виртуальных таблиц запросов?

ЛАБОРАТОРНАЯ РАБОТА № 20. Программная установка свойств документов

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

В системе 1С:Предприятие имеется объект ДокументОбъект, который позволяет обратиться к значениям реквизитов, задаваемым в конфигураторе. Для получения данного объекта следует воспользоваться методом ПолучитьОбъект(). Используя этот объект, мы можем не только считывать, но и устанавливать необходимые реквизиты документа.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

В листинге 2.5 приведена процедура, которая позволяет сменить даты документов на текущую дату. Ее можно расположить в качестве процедуры обработки нажатия на кнопку, расположенную на форме. Метод Записать() позволяет записать элемент справочника в базу данных; если его не вызвать, изменения будут утеряны.

Результат выполнения данной процедуры отражен в форме списка документов (рисунок 2.14).

Листинг 2.5. Процедура установки даты документов

```
Процедура КнопкаВыполнитьНажатие(Кнопка)
ВыборкаЗаказ = Документы.Заказ.Выбрать();
Пока ВыборкаЗаказ.Следующий() Цикл
    Объект = ВыборкаЗаказ.ПолучитьОбъект();
    Объект.Дата = ТекущаяДата();
    Объект.Записать();
КонецЦикла;
КонецПроцедуры
```

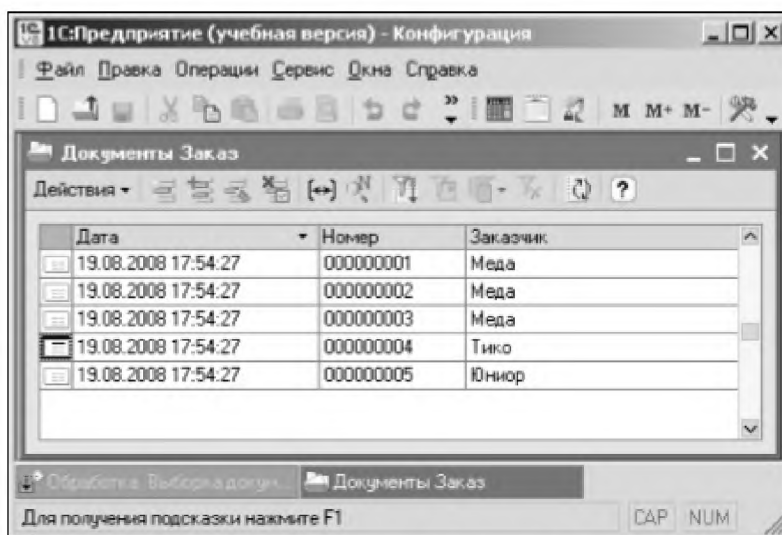


Рисунок 2.14 Результат изменения дат документов

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Какие средства входят в состав механизма универсального обмена данными?
- Для чего предназначен объект конфигурации План обмена?
- Каковы основные составляющие плана обмена?
- Что такое узлы плана обмена?

ЛАБОРАТОРНАЯ РАБОТА № 21. Заполнение документа на основании имеющихся данных

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Рассмотрим программное заполнение одного документа на основании данных, содержащихся в другом, уже заполненном документе. Это очень полезное действие — оно исключает повторный ввод информации, что повышает производительность работы и уменьшает вероятность ошибки. Один документ (Заказ) у нас есть, и для начала требуется создать еще один — Заявка (рисунок 2.15).

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

На рисунок 2.15 показана закладка Данные окна редактирования объекта конфигурации. Видно, что область шапки документа содержит один дополнительный

реквизит — ФирмаЗаказчик (тип данных — СправочникСсылка.Фирмы). Для табличной части документа выберем имя ПереченьТоваров и внесем в нее два реквизита:

- Номенклатура (тип данных — СправочникСсылка.Номенклатура);
- Количество (тип данных — Число без дробной части) для внесения количества заявляемых изделий.

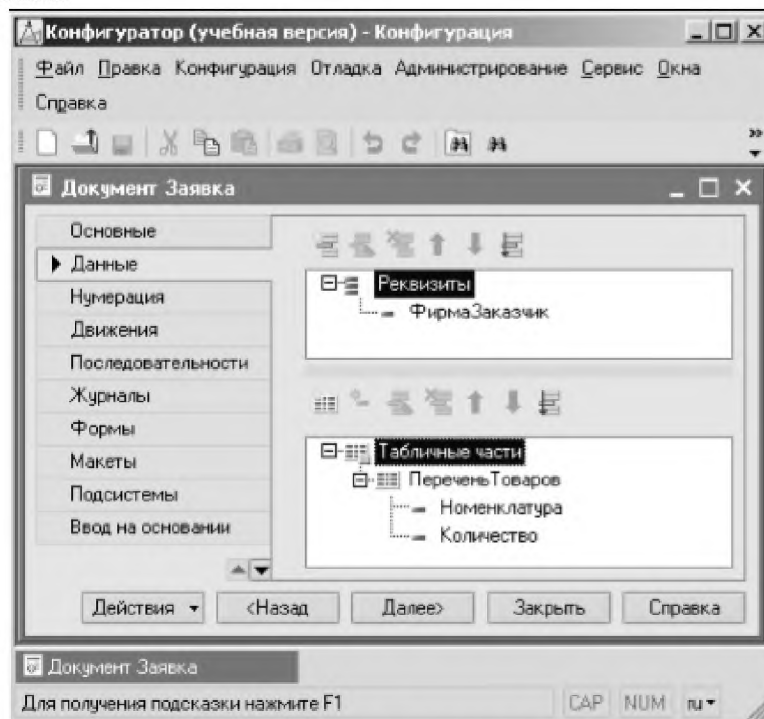


Рисунок 2.15 Окно редактирования документа Заявка

После этого потребуется в режиме 1С:Предприятие заполнить несколько документов по созданному шаблону (сформировать несколько заявок).

Один из заполненных документов показан на рисунок 2.16. Теперь наша задача заключается в том, чтобы на основании заполненной заявки программно сформировать документ Заказ. В нем будет указана фирма, а также заполнена табличная часть с перечнем товаров.

Для технической реализации данной задачи создадим форму обработки (рисунок 2.17). На форме расположим поле ввода Заявка (тип данных — ДокументСсылка.Заявка). Теперь для обеспечения функциональности нажатия на кнопку Выполнить следует добавить процедуру, представленную в листинге 2.6.

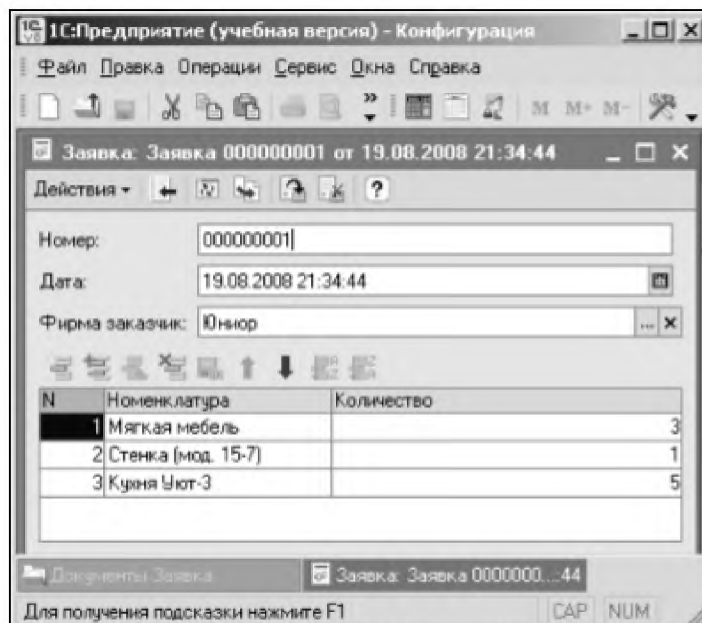


Рисунок 2.16 Заполнение заявки в режиме 1С:Предприятие

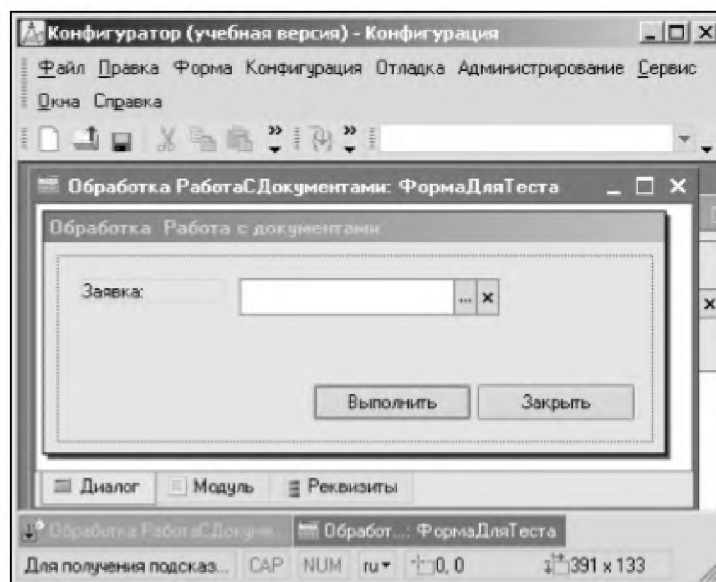


Рисунок 2.17 Форма обработки для создания заказа на основании заявки

Листинг 2.6. Процедура создания заказа на основании заявки

Процедура КнопкаВыполнитьНажатие(Кнопка)

НовыйЗаказ = Документы.Заказ.СоздатьДокумент();

НовыйЗаказ.Заказчик = Заявка.ФирмаЗаказчик;

НовыйЗаказ.Дата = ТекущаяДата();

Для Каждого СтрокаЗаявка Из Заявка.ПереченьТоваров Цикл

СтрокаЗаказТовар = НовыйЗаказ.Перечень.Добавить();

СтрокаЗаказТовар.Товар = СтрокаЗаявка.Номенклатура;

СтрокаЗаказТовар.Количество = СтрокаЗаявка.Количество;

КонецЦикла;

НовыйЗаказ.Записать();

КонецПроцедуры

Здесь мы с помощью уже знакомого нам метода СоздатьДокумент() создаем новый документ Заказ, копируем заказчика и устанавливаем дату создаваемого документа. Далее организуется цикл, проходящий по всем записям в документе Заявка:

Для Каждого Строка Заявка Из Заявка.Перечень Товаров Цикл

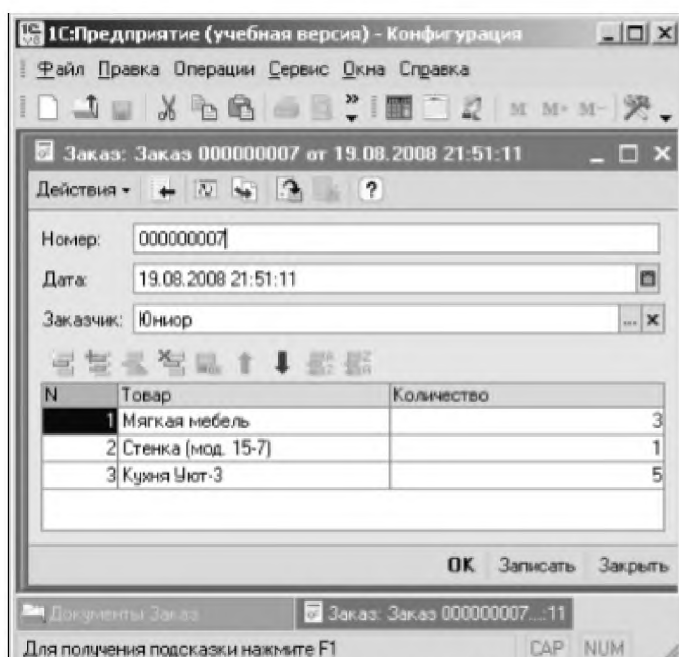


Рисунок 2.18 Программно созданный документ Заказ

В цикле производятся добавление строки в формируемый документ Заказ и заполнение табличной части в этом документе. В заключительной части процедуры созданный документ вносится в базу данных:

НовыйЗаказ.Записать();

Для проверки работоспособности созданной обработки следует перейти в режим 1С:Предприятие и открыть только что созданную форму. Выбрав в поле списка одну из имеющихся заявок и выполнив процедуру обработки щелчка по кнопке ОК, мы получим новый документ Заказ (рисунок 2.18).

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Что такое состав плана обмена, и для каких элементов данных возможен обмен данными?
- Что такое авторегистрация?
- Для чего предназначен механизм регистрации изменений?
- Как работает инфраструктура сообщений?

ЛАБОРАТОРНАЯ РАБОТА № 22. Обработка событий

Цель: Рассмотрим ряд вопросов, связанных с программным созданием документов и доступом к реквизитам документов

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Наиболее существенную составляющую прикладных решений на платформе 1С:Предприятие составляет обработка событий. В этой лабораторной работе мы рассмотрим обработку типичных событий, которые могут встретиться на практике.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Воспользовавшись событием ПриЗаписи, можно отменить запись, что приведет к тому, что документ не будет записан в базу данных. Например, мы можем программно запретить запись документов, если указана дата ранее текущей.

Событие ПриЗаписи определено для модуля объекта. Поэтому в окне редактирования документа Заказ нужно перейти на закладку Прочее и воспользоваться кнопкой Модуль объекта. Здесь и необходимо создать текст процедуры (листинг 2.7). В этом случае мы проверяем дату записываемого документа. Если эта дата меньше текущей даты, то происходит отмена записи информации в базу данных.

Листинг 2.7. Процедура обработки события ПриЗаписи

Процедура ПриЗаписи(Отказ)

Если ЭтотОбъект.Дата < ТекущаяДата() Тогда

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

Еще одно событие — ПередУдалением. С помощью его обработки можно оставить необходимые документы (не допустить их удаления из базы данных). Так, с помощью процедуры, приведенной в листинге 2.8, производится запрет удаления тех документов, дата которых ранее 5 августа 2008 года. Рассматриваемую процедуру также необходимо расположить в модуле объекта.

118 Глава 2

Листинг 2.8. Процедура обработки события ПередУдалением

Процедура ПередУдалением(Отказ)

Если ЭтотОбъект.Дата < Дата(2008,08,05) Тогда

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Как создать план обмена?
- Как настроить конфигурацию для обмена данными?
- Как реализовать обмен данными в общем виде?

Список рекомендованной литературы.

Основные источники:

1. Медведев, М. А. Программирование на СИ# : учебное пособие для СПО / М. А. Медведев, А. Н. Медведев ; под редакцией А. В. Присяжного. — 2-е изд. — Саратов, Екатеринбург : Профобразование, Уральский федеральный университет, 2019. — 62 с. — ISBN 978-5-4488-0471-7, 978-5-7996-2833-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87851.html> .. — Режим доступа: для авторизир. пользователей

Дополнительная литература:

1. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманов. — 2-е изд. — Саратов : Профобразование, 2019. — 310 с. — ISBN 978-5-4488-0127-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/89870.html>. — Режим доступа: для авторизир. пользователей
2. Иванов, В. Б. Прикладное программирование на C/C++: с нуля до мультимедийных и сетевых приложений / В. Б. Иванов. — Москва : СОЛОН-Пресс, 2018. — 240 с. — ISBN 978-5-91359-308-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/90397.html>. — Режим доступа: для авторизир. пользователей
3. Митина, О. А. Прикладное программирование : учебное пособие / О. А. Митина. — Москва : Московская государственная академия водного транспорта, 2017. — 94 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/76716.html>. — Режим доступа: для авторизир. пользователей

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п.п.	Содержание изменения	Дата, номер протокола заседания педагогического совета
1	2	3
1	Внесены изменения в перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.	решение от 27.08.2020 №7
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		