

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Кафедра интеллектуальных информационных технологий**

## **Методические указания**

к выполнению лабораторных работ по курсу  
**«Программные средства систем обработки данных»**  
для студентов специальности 53 01 02  
**«Автоматизированные системы обработки информации»**

**Часть 2**

**Брест 2004**

Методические указания к выполнению лабораторных работ по курсу «Программные средства систем обработки данных»

Методические указания предназначены для студентов специальности «Автоматизированные системы обработки данных», изучающих курс «Программные средства систем обработки данных» на примере программного комплекса «1С:Предприятие» и выполняющих лабораторные работы по данному курсу. Методические указания содержат теоретическую информацию и задания на выполнение лабораторных работ.

Рассмотрено на заседании кафедры интеллектуальных информационных технологий 10.05.2004 г. протокол №6

Составители: Селезнёв П.В., ассистент  
Гладышева Е.Л., ассистент

## Введение

Курс лабораторных работ по дисциплине «Программные средства систем обработки данных» включает изучение системы «1С:Предприятие 7.7». Программный комплекс «1С:Предприятие 7.7» представляет собой универсальный инструмент автоматизации управления экономической деятельностью предприятия. С его помощью могут быть автоматизированы различные виды бухгалтерского учета, расчет заработной платы, ведение кадровой документации и многое другое.

В результате освоения курса лабораторных работ по данной дисциплине студент должен:

- уметь представить задачу, подлежащую конфигурированию, в виде совокупности взаимосвязанных объектов метаданных и алгоритмов их обработки;
- настраивать интерфейсы и наборы прав доступа, а также определять список пользователей, обладающих конкретным видом интерфейса и набором прав;
- уметь проводить сохранение и восстановление данных;
- проводить корректировку существующей конфигурации;
- создавать, удалять и корректировать свойства объектов метаданных;
- уметь редактировать форму объектов, создавать описание объекта;
- значения;
- уметь пользоваться средствами отладчика (точки останова, таблица значений, пошаговое выполнение) для отладки модулей конфигурации;
- уметь пользоваться средствами просмотра журнала регистрации в режимах запуска "Монитор" и "1С:Предприятие".

## Лабораторная работа №1

### СОЗДАНИЕ СОБСТВЕННОЙ ТИПОВОЙ КОНФИГУРАЦИИ

Цель работы. Получение практических навыков по формированию информационной базы системы, основных объектов метаданных, а также возможностей настройки пользовательских интерфейсов и администрирования системы 1С:Предприятие.

#### 1 Краткие теоретические сведения

Готовое учетное решение, поставляемое разработчиками программы, называется типовой конфигурацией. Фирмой 1С программа поставляется с одной из существующих типовых конфигураций.

Возможны два основных режима работы в системе 1С:Предприятие:

- **пользовательский режим**, когда осуществляется ввод документов, операций, заполнение справочников, выполнение расчетов, формирование различной отчетности;

- **режим конфигурирования**, когда типовая конфигурация может быть изменена, а именно выполняется изменение состава и свойств различных объектов системы (объектов метаданных) в соответствии с потребностями конкретного предприятия. Для решения перечисленных задач используются два основных механизма:

- редактор форм;
- встроенный язык программирования.

Редактор форм – это инструмент визуального конструирования, предназначенный, в первую очередь для модификации и настройки пользовательского интерфейса. В его состав входят:

- редактор диалогов;
- табличный редактор;
- текстовый редактор;
- конструкторы.

Особенностью работы с данными в системе 1С:Предприятие является то, что для визуального представления любого объекта данных используется **форма**, которая является совокупностью 3 вариантов описания этого объекта (при желании пользователь может отказаться от любого из трех вариантов):

- в виде диалогового окна, содержащего интерактивные элементы (кнопки, текстовые поля и т.д.);
- в виде печатного документа;
- в виде алгоритма на встроенном языке программирования (модуля).

Редактор диалогов обеспечивает создание и модификацию интерфейса объекта данных, табличный редактор предназначен для подготовки печатного представления, а текстовый редактор используется для работы с модулем. Конструктор (или Мастер) – это сервисный инструмент, упрощающий работу пользователя при выполнении наиболее типичных операций по созданию и редактированию объекта данных. Встроенный язык системы 1С:Предприятие может быть отнесен к предметно-ориентированным языкам высокого уровня. Основу языка составляют конструкции, предназначенные для работы с логическими типами данных, характерными для рассматриваемой предметной области, - документами, справочниками, таблицами и т.д. Данный язык сочетает в себе процедурную и объектную технологию программирования.

## **1.1 Регистрация информационной базы и установка атрибутов конфигурации.**

На одном компьютере может быть создано несколько информационных баз. Каждая информационная база хранится в нескольких взаимосвязанных файлах. Перечень подключенных к системе 1С:Предприятие информационных баз отображается в списке *Информационные базы* окна запуска системы. Каждая информационная база помещается в отдельную папку. Чтобы создать новую информационную базу следует выполнить последовательность действий:

- 1) на одном из жестких дисков создать пустую папку (например, *New\_Bas*);
- 2) в окне запуска системы нажать кнопку *Добавить*;
- 3) в окне регистрации информационной базы ввести название информационной базы (например, *Учебная*) и путь к папке, содержащей файлы информационной базы (*New\_Bas*);
- 4) в окне выбора кодовой страницы принять по умолчанию предлагаемый системой 1С:Предприятие выбор.

Все последующие действия по формированию конфигурации выполняются с помощью Конфигуратора. Для этого в списке *В режиме* окна запуска системы выбрать пункт *Конфигуратор*. В окне Конфигуратора выполнить команду *Конфигурация – Открыть конфигурацию*. Система 1С:Предприятие сгенерирует «заготовку» новой конфигурации (т.н. дерево метаданных) и откроет ее в окне *Конфигурация*.

Атрибуты конфигурации устанавливаются с помощью панели свойств (п. *Свойства* контекстного меню корневого элемента конфигурации).

## **1.2 Создание объектов метаданных.**

Метаданные – это множество классов (типов) объектов, используемых для описания предметной области. Каждый класс характеризуется определенным набором свойств. Указав принадлежность объекта к некоторому классу, мы тем самым определяем тот набор свойств, которым характеризуется данный объект.

К основным видам метаданных, используемым в данной работе, относятся: *константы, справочники, документы, журналы*.

### **1.2.1 Константы**

Константы предназначены для хранения постоянных или условно-постоянных (редко изменяющихся данных). Для создания новой константы используется команда контекстного меню *Новая Константа*. Получить значение константы можно с помощью метода *ПолучитьАтрибут* («Имя реквизита») с последующим присвоением некоторой переменной. Например:

*Адрес=Константа.ПолучитьАтрибут("Op\_адрес");*

### **1.2.2 Справочники**

Справочники предназначены для хранения списков однородных элементов данных. Основное назначение справочников – это ускорение и упрощение ввода данных в документы.

Каждый элемент справочника имеет два обязательных реквизита – код и наименование (по умолчанию):

Справочники могут быть **многоуровневые**, в которых некоторые элементы справочника являются наименованием групп элементов более низкого уровня иерархии. Справочник может использоваться как детализированное описание некоторого элемента справочника более высокого уровня. При этом он будет называться **подчиненным справочником**, а справочник, содержащий ссылку на него, будет называться **справочником-владельцем**.

Справочник может быть создан с помощью команды Новый справочник контекстного меню вида метаданных Справочники.

Для справочника предусмотрены 3 вида форм:

- форма элемента справочника используется для ввода, просмотра и корректировки значений реквизитов элемента справочника; для элемента справочника обязательно используется интерфейсная часть формы (диалог) и, как правило, программный модуль;

- форма группы применяется в многоуровневых справочниках; служит для просмотра и редактирования значений реквизитов группы; для группы используются, как правило, 2 компонента формы – диалог и программный модуль;

- форма списка определяет формат визуального представления содержимого справочника. Данная форма используется для всех объектов метаданных, содержимое которых может быть представлено как список однотипных элементов. Для формы списка используются все 3 компонента (диалог, программный модуль и печатное представление). Построение форм списка выполняется с помощью специального конструктора, входящего в состав Редактора форм. С помощью формы списка могут решаться 2 основные задачи: просмотр содержимого объекта (просмотр списка) и его редактирование (ввод новых элементов или удаление имеющихся): выбор элемента списка. Формы списка создаются и редактируются при нажатии кнопки *Формы списка – Редактировать* диалогового окна Справочник.

### 1.2.3 Документы

Документы - средство для ввода первичной информации о совершаемых хозяйственных операциях. У любого вида документа существует два обязательных реквизита, которые создаются автоматически – дата и номер документа. Другие реквизиты определяются в конфигураторе конкретно для каждого создаваемого вида документа. Большинство документов, используемых в системе 1С:Предприятие, являются электронными аналогами бумажных документов, используемых на предприятии. В документах выделяют две основные структурные части: шапка документа и многострочная табличная часть. Реквизитом шапки считается любой реквизит, значение которого используется в документе только один раз. Табличная часть документа по структуре аналогична справочнику и содержит последовательность однотипных строк, в которых, собственно, и отображается «информационное наполнение» документа. Документ может включать только одну табличную часть. Важным свойством документов является возможность их проведения. Проведение

документа предполагает автоматическое формирование проводок, отражающих в бухгалтерском учете факт хозяйственной деятельности, зафиксированный документом, либо изменения в справочниках и документах, вызываемые проведением данного документа. Действия, предусмотренные документом, фиксируются в двух модулях – *модуле формы документа и модуле документа*.

#### **1.2.4 Журналы**

Журналы документов – обеспечивают представление списков объектов типа Документ и взаимодействие с ними пользователей. Распределение документов по журналам производится произвольным образом, в соответствии с потребностями конкретной прикладной задачи. Вместе с тем, все документы при создании автоматически помещаются в системный журнал, имеющий условное наименование *Общий (Полный)*. Интерфейс любого журнала обеспечивает выполнение операций по созданию, редактированию и удалению документов, входящих в данный журнал.

#### **1.3 Редактор диалогов**

Данный инструмент позволяет в режиме визуального редактирования создавать графический интерфейс объекта метаданных, посредством которого пользователь может взаимодействовать с этим объектом. Процедура разработки диалога сводится к вставке в поле диалога требуемых элементов графического пользовательского интерфейса, их пространный компоновке и установке соответствующих атрибутов этих элементов. Вызов Редактора форм происходит при открытии формы объекта одновременно с загрузкой двух других основных инструментов Редактора форм.

В качестве элементов графического пользовательского интерфейса используются как стандартные («объектно-независимые») для Windows-приложений (кнопки, переключатели, флажки, списки, текстовые поля), так и специфические («объектно-зависимые»), соответствующие основному типу прикладных задач, решаемых средствами системы 1С:Предприятие.

К специфическим элементам относятся:

- *многострочная часть (таблица)* используется при создании диалога для объектов, содержимое которых организовано в виде списка однородных элементов; диалог может содержать только один элемент такого типа;

- *дерево групп* обеспечивает визуальное представление в виде дерева табличных данных, имеющих иерархическую структуру; в диалоге используется вместе с элементом типа Многострочная часть; и соответственно в диалоге может присутствовать только один элемент такого типа;

- *реквизит объекта метаданных* предназначен для просмотра и редактирования одного из реквизитов агрегатного объекта метаданных (справочника, документа или журнала); визуально может быть реализован в виде поля ввода, флажка, переключателя или комбинированного элемента, состоящего из поля ввода и кнопки вызова связанного с ним объекта.

Возможность использования специфических элементов в диалоге зависит от типа и свойств объекта метаданных, для которого создается диалог. Вставка их в диалог выполняется с помощью специального диалогового окна *Размещение реквизитов*.

К средствам пространственной компоновки элементов интерфейса относится механизм слоев. Механизм слоев используется в тех случаях, когда состав отображаемых элементов зависит от текущей ситуации или когда приходится размещать большое количество элементов на одной диалоговой панели. Слои используются также для разделения диалога на несколько «страниц» - вкладок. (Разделение диалога на вкладки выполняется средствами встроенного языка с помощью метода *ИспользоватьЗакладки*).

Слой – это воображаемая поверхность, параллельная плоскости шаблона, предназначенная для размещения и управления видимостью элементов диалога. Атрибутами слоя являются видимость и активность. В исходном состоянии в списке слоев присутствует единственный слой – Основной. Для создания и установки параметров слоев используется специальное диалоговое окно, вызываемое на экран командой *Диалог – Слои*. Окно содержит список слоев (например, Слой1, Слой2 и т.д.) и кнопки для создания, удаления и управления слоями. Динамика переключения между слоями (точнее, изменения их видимости) описывается средствами встроенного языка. Чтобы создать и использовать в форме справочника закладки, следует использовать следующие процедуры модуля формы справочника.

Предопределенная процедура *ПриОткрытии()* – обеспечивает формирование первоначального варианта окна диалога. В ней должно выполняться создание элементов списка типа «Закладки». Для этого существуют специальные методы, входящие в контекст модуля формы:

*Форма.ИспользоватьЗакладки(1)*

*Форма.ЗакладкиДобавитьЗначение(Значение, “Строка символов”)*, где Значение – номер слоя по порядку, Строка символов – наименование слоя.

*Форма.ИспользоватьСлой(“ИмяСлоя”, Режим)*,

где “ИмяСлоя” – задает слой, который следует использовать в окне диалога, Режим – определяет режим отображения слоев (0 – скрыть слой; 1- показать слой; 2 – показать слой, скрыв все остальные).

Предопределенная процедура *ПриВыбореЗакладки(Ном, Значение)* описывает динамику ведения диалога при переключении пользователя с одной закладки на другую (с помощью операторов условного выполнения *Если Значение = 1 Тогда...Иначе...КонецЕсли; Продолжить* – передает управление в начало цикла). Завершением процедуры является ключевое слово *КонецПроцедуры*.

#### **1.4 Конструкции встроенного языка**

Доступ к реквизитам *Справочника* из документа происходит при помощи переменной со ссылкой на объект типа *Справочник*. Объект создается функцией *СоздатьОбъект*, ссылка на который присваивается переменной. При создании переменной со ссылкой на объект типа *Справочник* в качестве имени справочника должно выступать полное наименование конкретного вида справочника, как он объявлен в конфигураторе. Например:

*Спр1= СоздатьОбъект(“Справочник.Сотрудники”);*

Объект, созданный при помощи функции *СоздатьОбъект*, изначально не спозиционирован, т.е. не указывает на конкретный элемент справочника. Позиционирование элемента справочника осуществляется при помощи методов *НайтиЭлемент, НайтиПоКоду, ПолучитьЭлемент* и т.п.



## **Методы агрегатного типа данных Справочник.**

**ВыбратьЭлементы (режим)** – открыть выборку элементов справочника. Режим 0 – выбирать элементы без учета иерархии; 1 – с учетом иерархии.

**ПолучитьЭлемент()** – получить из выборки следующий элемент справочника. Метод используется для организации цикла поиска по справочнику (ключевые слова: *Пока ... Цикл ... КонецЦикла*).

**ЭтоГруппа()** – проверяет принадлежность группе.

**ТекущийЭлемент()** – получает значение элемента справочника. Данный метод используется, если элемент справочника нужно присвоить какому-либо реквизиту. Выборка элемента из справочника осуществляется либо по коду, либо по наименованию на этапе создания справочника.

**Получить(<Дата>)** – получает значение периодического реквизита справочника на дату или документ, где <Дата> - ДатаДок. Метод можно использовать только для периодических реквизитов справочника.

Например:

КФ = СоздатьОбъект("Справочник.Коеффициенты");

КФ.ВыбратьЭлементы(0);

Пока КФ.ПолучитьЭлемент() = 1 Цикл

    НоваяСтрока();

    Перем1 = КФ.ТекущийЭлемент();

    Перем2 = КФ.Реквизит.Получить(ДатаДок);

КонецЦикла;

## **Предопределенные процедуры Модуля документа**

Процедура **ОбработкаПроведения()** – не является встроенной процедурой языка. Для нее определено только название и синтаксис. Тело процедуры должно быть написано разработчиком конфигурации в Модуле документа.

## **Методы документа**

Доступ к атрибутам, вызов методов конкретного документа происходит при помощи переменной со ссылкой на объект типа *Документ*. Например:

*Док* = СоздатьОбъект("Документ.БольшичныйЛист");

Атрибут *ДатаДок* – дата документа.

Метод **ВыбратьДокументы** – открывает выборку документов.

Метод **ПолучитьДокумент** – выбирает следующий документ в последовательности выборки, открытой при помощи метода **ВыбратьДокументы**.

Метод **НоваяСтрока()** – добавляет новую строку с очередным.

Метод **ВыбратьСтроки()** – открыть выборку строк многострочной части документа. Возвращает: 1 - если действие выполнено, и в выборке есть хотя бы одна строка; 0 - если действие не выполнено, или в выборке нет ни одной строки.

Метод **ПолучитьСтроку()** – получить из выборки следующую строку табличной части документа. Возвращает: 1 - если строка выбрана, 0 - иначе.

Метод **ТекущийДокумент** – возвращает значение позиционированного текущего документа (в целом, как объекта). Например, если нужно документ присвоить какому-либо реквизиту:

*ТекДок* = *Док.ТекущийДокумент()*;

Метод *ЗагрузитьТабличнуюЧасть(ТаблЗнач)* – загружает многострочную часть документа из таблицы значений.

Метод *ВыгрузитьТабличнуюЧасть* – выгружает многострочную часть документа в таблицу значений.

*УстановитьРеквизитСправочника* (<ЭлементСправочника>, <НазваниеРеквизита>, <Значение>, <Дата Установки>)

**Назначение:** Записать значение периодического реквизита справочника с привязкой к проведению документа.

**Параметры:**

<ЭлементСправочника> - элемент справочника, в который будет запись;

<НазваниеРеквизита> - название периодического реквизита справочника (например, "Оклад");

<Значение> - новое значение периодического реквизита;

<ДатаУстановки> - дата установки нового значения периодического реквизита (Формат *ДатаДок*).

**Замечание:** Метод доступен только в Модуле документа в предопределенной процедуре *ОбработкаПроведения*.

*ВвестиЧисло*(<Число>, <Подсказка>, <Длина>, <Точность>) – вызов диалога для ввода числа.

**Параметры:**

<Число> - имя переменной, объявленной в модуле, для приема вводимого значения;

<Подсказка> - текст заголовка окна диалога ввода;

<Длина> - длина вводимого числа;

<Точность> - число знаков после десятичной точки;

Например:

Процедура Подбор ()

  ВыбратьСтроки();

Пока ПолучитьСтроку()>0 Цикл

  Если ВвестиЧисло(Кол, "Введите Количество",5,1)=1 Тогда

    Сумма=Цена\*Кол;

  КонецЕсли;

  КонецЦикла;

КонецПроцедуры

порядковым номером в многострочную часть документа.

## **1.5 Администрирование**

В составе конфигуратора имеются средства настройки пользовательских интерфейсов по категориям пользователей и средства управления правами пользователей.

### **1.5.1 Редактирование интерфейсов**

Под индивидуальным пользовательским интерфейсом понимается состав команд меню и кнопок панели инструментов, используемых в пользовательском режиме.

Различают два типа пользовательских интерфейсов: родительский и подчиненный. Формирование набора команд и кнопок панели инструментов выполняется только для родительских интерфейсов. Подчиненный интерфейс при создании наследует состав команд меню и кнопок родительского интерфейса. Все действия по формированию иерархии пользовательских

интерфейсов выполняются в окне конфигурации на вкладке *Интерфейсы*. В исходном состоянии эта вкладка пуста, по умолчанию предполагается, что это интерфейс администратора системы. На его основе должны формироваться интерфейсы всех других категорий пользователей. Для создания нового пользовательского интерфейса необходимо щелкнуть на кнопке *Новый элемент*, выбрать команду *Новый интерфейс*, с помощью элементов управления панели свойств интерфейса установить его основные атрибуты. Формирование интерфейсов выполняется с помощью специальных инструментов – Редактора меню и Редактора панелей инструментов.

Процесс создания меню облегчает вызов Конструктора меню (п. Действия). Окно Конструктора меню содержит два списка. Список Колонки включает перечень всех типов объектов метаданных. Наличие флажка возле типа объекта означает, что соответствующее меню будет представлено в окне. В списке Пункты отображается перечень команд меню, выбранного в списке Колонки.

### 1.5.2 Описание прав пользователей

Формирование наборов пользовательских прав выполняется в окне конфигуратора на вкладке *Права*. Чтобы создать новый набор прав следует выбрать команду *Действия – Новый*, указать его название, далее двойным щелчком вызвать *Редактор пользовательских прав*, в котором установить набор разрешенных операций над конфигурацией в целом и над отдельными объектами метаданных.

### 1.5.3 Создание списка пользователей

Список пользователей вызывается на экран командой *Администрирование – Пользователи*. Создание нового пользователя выполняется по команде *Действия – Новый*. Закладка *Атрибуты* содержит управляющие элементы для присвоения пользователю краткого и полного имени и указания рабочего каталога. Рабочий каталог используется для размещения индивидуальных файлов конфигурации и различных временных и служебных файлов, создаваемых системой 1С:Предприятие (указанный рабочий каталог должен существовать на диске). Закладка *Роль* служит для выбора наименования пользовательского интерфейса и набора прав, которые будут поставлены в соответствие имени пользователя. Для установки пароля для пользователя следует его выделить и выполнить команду *Действия – Изменить пароль*.

## 2 Задание для выполнения

При создании объектов метаданных в режиме конфигурирования рекомендуется максимально использовать Синтаксис-Помощник. Результаты выполнения заданий продемонстрировать преподавателю в пользовательском режиме.

2.1 Создать новую информационную базу и назвать ее Учебная. В ее свойствах установить флажок *разрешить непосредственное удаление объектов*.

2.2 В режиме конфигурирования новой информационной базы создать следующие виды справочников:

### Подразделения

Поле	Тип	Кол-во
Код	число	5
Подразделение	строка	25

Редактировать в форме списка.

### Список должностей

Поле	Тип	Кол-во
Код	число	5
Должность	строка	30

Редактировать в форме списка.

### Сотрудники

Поле	Тип	Кол-во	Точность	Периодичность
Код	число	5		
ФИО	строка	30		
Оклад	число	10	2	Да
Дата приема	дата			Да
Дата увольнения	дата			Да
Подразделение	Справ.Подразделения			Да
Должность	Справ.Должность			Да
Адрес	строка	30		Да
Телефон	строка	10		Да
Номер док	строка	10		
Дата выдачи	дата			
Кем выдан	строка	60		

Для реквизита Оклад установить флажок – *Изменять документами*.  
Справочник – многоуровневый, количество уровней – 2. Редактировать в форме элемента. Предусмотреть 2 формы списка: *Для выбора* и *Основная*. Форма списка *Для выбора* включает поля: *код, ФИО, должность*. Форма списка *Основная* включает *дерево групп* и поля: *код, ФИО, должность*. Справочник должен иметь 3 закладки: *Основная, Адресные данные* и *Паспортные данные*. По закладке *Адресные данные* устанавливается слой формы *Адрес*, который включает поля *Адрес* и *телефон*. По закладке *Паспортные данные* устанавливается слой формы *Паспорт*, который включает поля: *Номер док., Дата выдачи, Кем выдан*.

С помощью конструктора печати в форме списка *Основная* создать печатный аналог списка *Справочники* (по нажатию кнопки).

### 2.3 Создать следующие константы:

Код	Тип
Наименование организации	строка
Юридический адрес	строка
Телефон	строка
Дата регистрации	дата
Директор	Справочник.Сотрудники
Главный бухгалтер	Справочник.Сотрудники

2.4 Создать 2 пользовательских интерфейса: родительский – для администратора и подчиненный – для секретаря. Интерфейс администратора должен включать в строку командного меню пункты *Справочники, Документы, Журналы*. В Меню *Справочники* через *разделитель* включить пункт *Константы*. Панель инструментов должна содержать кнопки открытия всех созданных документов.

2.5 Создать документ *Изменение окладов*. Документ должен проводиться и перепроводиться при записи. Обеспечить хранение документа в журнале *Приказы* (все последующие документы хранить там же).

Реквизиты шапки: *Номер* и *дата* документа (по умолчанию).

Реквизиты табличной части: *ФИО* – агрегатный тип *Справочник. Сотрудники*, *Оклад* – тип *число*.

Документ должен обеспечивать интерактивный ввод сотрудника, изменение его оклада и запись нового значения оклада в справочник. (Редактировать только модуль документа, использовать процедуру *ОбработкаПроведения()* и методы *ВыбратьСтроки()*, *ПолучитьСтроку()*, *УстановитьРеквизитСправочника*).

С помощью конструктора печати создать печатный аналог данного документа. В шапке печатной формы должны присутствовать реквизиты организации (Наименование, юридический адрес, ФИО директора, ФИО гл. бухгалтера, Телефон).

2.6 Создать документ *Изменение окладов2*, в котором вышеперечисленные действия выполняются по нажатию кнопки *Заполнить*. Для этой кнопки в модуле формы документа создать процедуру *Заполнить()* (использовать следующие методы *ВыбратьЭлементы*, *ПолучитьЭлемент*, *НоваяСтрока*, *ТекущийЭлемент*, *Получить*).

2.7 Создать справочник *КозффОкладов*. Реквизиты – *Код* (тип текстовый), *Козфф* (тип число, точность 5.2), *ФИО* (агрегатный тип *Справочник.Сотрудники*). В него должны быть включены коэффициенты пересчета окладов по каждому сотруднику. Редактировать в форме списка. Создать документ *Пересчет окладов*. Табличная часть включает реквизиты: *ФИО* (агрегатный тип *Справочник.Сотрудники*), *Оклад* (число), *Козфф* (число), *НовыйОклад* (число), *Код* (агрегатный тип *Справочник.КозффОкладов*). Предусмотреть заполнение граф *Оклад* и *Козфф* из справочников *Сотрудники* и *КозффОкладов* по нажатию кнопки в форме (например, *Заполнить*). По кнопке *Пересчет* вводятся новые значения коэффициентов в специальном окне ввода, пересчитываются оклады (заполняются графы *НовыйОклад* и *Козфф*), получается печатная форма (по кнопке *Печать*), и в результате выполнения документа новые значения окладов и коэффициентов заносятся в их справочники.

2.8 Сформировать 2 набора пользовательских прав – для администратора и секретаря. Для секретаря установить только права на чтение, а применительно к справочнику *Сотрудники* – чтение и ввод нового элемента. Создать списки пользователей в соответствии с их правами и паролями. Продемонстрировать в пользовательском режиме.

### **3. Контрольные вопросы**

3.1 Понятие типовой конфигурации, основные режимы работы.

3.2 Средства и возможности администрирования.

3.3. Основные объекты метаданных.

3.4 Назначение Редактора форм и его состав.

3.5. Формы представления объектов метаданных.

### **4 Содержание отчета**

4.1 Цель работы.

4.2. Тексты алгоритмов на встроенном языке.

4.3. Выводы по работе.

**Лабораторная работа № 2**  
**Реализация бухгалтерской задачи**  
**«Расчеты по краткосрочным кредитам и займам»**  
**(часть 1)**

Цель работы: Получение практических навыков по перенастройке существующей типовой конфигурации. Ознакомиться с объектами метаданных «План счетов», «Виды субконто», «Перечисление», «Операция» и «Проводка».

## **1 Краткие теоретические сведения**

Ведение бухгалтерского учета в системе 1С:Предприятие обеспечивают объекты метаданных «Планы Счетов», «Виды Субконто», «Операция» и «Проводка». Эти объекты тесно связаны между собой.

В соответствии с компонентной организацией системы 1С:Предприятие, возможности ведения бухгалтерского учета в системе появляются после установки компоненты «Бухгалтерский учет».

### **1.1 Планы счетов**

*Планом счетов* называется совокупность счетов, предназначенных для группировки информации о хозяйственной деятельности предприятия. Информация, накапливаемая на таких счетах, позволяет получить полную картину состояния средств предприятия в денежном выражении (синтетический учет).

Планы счетов в системе 1С:Предприятие поддерживают многоуровневую иерархию «счет - субсчета». Редактирование планов счетов может выполняться как в Конфигураторе, так и в пользовательском режиме. Для просмотра или выбора счета план счетов всегда выдается в виде таблицы. При редактировании плана счетов в Конфигураторе доступны все операции в пределах заданных для планов счетов свойств:

- ввод новых счетов и субсчетов;
- удаление счетов и субсчетов;
- включение валютного и количественного учета на счете (субсчете) путем задания соответствующих признаков;
- редактирование признака «Активный — Пассивный»;
- настройка аналитического учета.

Для ввода нового счета (субсчета) в план счетов следует нажать клавишу Insert или выбрать пункт «Новая строка» в контекстном меню плана счетов. В появившейся новой строке заполнить колонку «Код» и нажать клавишу Enter. Если в план счетов вводится новый счет не последнего уровня, будет выдан запрос «Счет будет иметь субсчета?», на который необходимо ответить «Да» или «Нет», т.к. счета в плане счетов делятся на 2 вида — *группы счетов* и *собственно счета*. Различие между ними простое: группы счетов имеют субсчета, а «собственно счета» не имеют субсчетов. Если счет будет иметь субсчета (то есть вводится группа счетов), то такой счет нельзя будет указать в качестве корреспондирующего при вводе проводок — необходимо будет указывать субсчета этого счета. И наоборот: если счет не будет иметь субсчетов, его можно указывать при вводе проводок.

### 1.3 Аналитический учет

Информация о средствах предприятия, которая накапливается на счетах бухгалтерского учета, имеет обобщающий характер. Для получения детальной информации о наличии конкретных хозяйственных средств (например, какого-нибудь материала или основного средства) необходима организация аналитического учета по хозяйственным средствам. В этом случае общие суммы на счетах, субсчетах разбиваются на более мелкие — стоимости конкретных хозяйственных средств.

В системе 1С:Предприятие аналитический учет организуется при помощи специального механизма *субконто*. Термином “субконто” могут быть обозначены любые объекты аналитического учета: основные средства, нематериальные активы, материалы, организации, подотчетные лица, договоры. *Видом субконто* называется множество однотипных объектов аналитического учета. Аналитический учет можно вести для любого счета или субсчета и включать до 5-ти различных разрезов аналитики. Объекты аналитического учета могут представлять собой элементы справочников, документы или произвольные числа, даты и строки.

### 1.4 Операции и проводки

Операцией считается любое хозяйственное действие, способное вызвать изменение состояния средств предприятия. Для ввода информации о хозяйственных операциях в системе 1С:Предприятие используется документ специального вида — “Операция”. Он позволяет занести в систему информацию о хозяйственной операции и зафиксировать дату и время совершения операции, сумму и содержание операции. Операция может быть автоматически сформирована на основе Документа (объекта данных типа “Документ”). Порядок такой генерации определяется в Конфигураторе средствами встроенного языка. Операция, введенная автоматически, всегда связана с породившим ее документом: если документ необходимо отредактировать, то при его проведении операция будет сформирована заново; при удалении документа будет удалена и операция документа.

Операция может вводиться пользователем “вручную”. В этом случае пользователь сам заполняет все реквизиты операции.

Чтобы введенная операция вызвала изменение в бухгалтерских итогах, такая операция должна содержать *проводки*. Ввод бухгалтерских проводок в системе 1С:Предприятие реализуется при помощи объекта метаданных “Проводка”. Особенность ввода проводок состоит в том, что проводки всегда вводятся в составе операции. Проводки операции составляют группу и при любых изменениях в реквизитах операции или в самих проводках всегда “держатся” вместе.

Если операция автоматически формируется документом, то, помимо заполнения реквизитов операции, документ может автоматически сформировать необходимые бухгалтерские проводки. Порядок генерации проводок также определяется средствами встроенного языка.

### 1.5 Пример формирования операции на основе Документа (объекта данных типа “Документ”).

Для примера возьмем документ из типовой конфигурации “Перемещение основного средства”. Данный документ производит перемещение кон-

кретного основного средства с одного места хранения на другое. Модуль формы документа обеспечивает выбор данного основного средства и пути его перемещения. Таким образом, на момент проведения документа мы имеем все необходимые данные для построения правильных проводок. Рассмотрим фрагмент модуля проведения (модуля документа).

```
Операция.НоваяПроводка();
Операция.СодержаниеПроводки = "Перемещение ОС";
Операция.НомерЖурнала = "ОС";
Операция.Дебет.Счет = СчетПоКоду("01.1");
Операция.Дебет.ОсновныеСредства = ОС;
Операция.Дебет.МестаХранения = МестоХранения2;
Операция.НомерЖурнала = "ОС";
Операция.Кредит.Счет = СчетПоКоду("01.1");
Операция.Кредит.ОсновныеСредства = ОС;
Операция.Кредит.МестаХранения = МестоХранения1;
Операция.Сумма = БалансоваяСтоимость;
Операция.Записать();
```

КонецПроцедуры

Формируется новая операция, в которой будет существовать одна проводка. Атрибут *Операция* предназначен для обращения к бухгалтерской операции документа.

Метод *НоваяПроводка()* создает новую проводку для текущей операции. Метод *Записать()* выполняет запись новой или измененной операции.

## 2 Задание для выполнения

Требуется создать следующие информационные объекты:

2.1 Справочник «Агенты» – структурное подразделение или отв. лицо (сотрудник), несущее ответственность по заключению и выполнению договора займа. Реквизиты справочника:

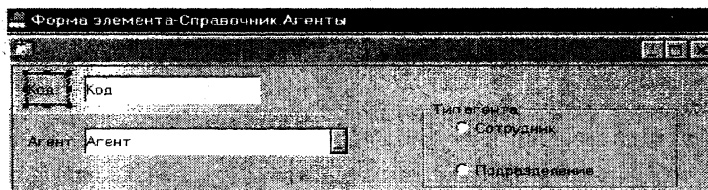
*код;*

*наименование;*

*агент (тип значения – Справочник неопределенного вида);*

*флаг (тип значения – число, длина - 1).*

Флаг используется в качестве идентификатора первого элемента в группе переключателей для конкретизации типа реквизита Агент). Элемент диалога «Переключатель» может принимать числовое значение от 0 до числа, равного количеству элементов в группе: 0 – ни один элемент в группе не выбран; 1 – выбран первый элемент (это элемент, помеченный как «Первый в группе»); 2 – выбран второй элемент в группе и т.д.





Модуль формы справочника должен содержать процедуру переключения агентов (Например Процедура *ПриВыборе()*) – в поле *Формула* элемента диалога типа «Переключатель» с идентификатором *Флаг*. Использовать системную процедуру - *НазначитьВид* и метод *НеИзменятьВид*.

**2.2 В рабочем плане счетов на счете 66 «Расчеты по кредитам и займам»** предусмотреть ведение аналитического учета в трех независимых разрезах: агентов, контрагентов и договоров займа. Для этого необходимо внести счет 66 в план счетов (групповой). Счет 66 должен иметь три аналитики (*Субконто*): справочник «Сведения о заимодавцах», справочник «Сведения об агентах», документ «Сведения о заключенных договорах займа» (предварительно создать указанные виды субконто). Предусмотреть на счете 66 два субсчета Расчеты в рублях и Расчеты в валюте с соответствующими признаками.

**2.3 Предусмотреть ввод и хранение в журнале *ДоговорыЗайма*** документа *ДоговорЗайма* (документ, являющийся основанием совершения операций займа), включающего следующие реквизиты:

- номер договора;
- дата договора;
- агент (*Справочник Агенты*);
- заимодавец (*Справочник Контрагенты*);
- дата начала действия договора;
- сумма договора;
- валюта договора (*Справочник Валюты*);
- срок договора (в днях);
- дата окончания договора (рассчитывается автоматически);
- процент годовых с суммы договора, выплачиваемый заимодавцу при возврате денежных средств по окончании срока договора;
- досрочный процент, выплачиваемый заимодавцу при возврате денежных средств при досрочном завершении договора.

The screenshot shows a window titled "ДоговорЗайма - ?" with a standard Windows-style title bar. The form contains the following fields and values:

№ договора	<input type="text"/>	Дата договора	01.02.04
Агент	Отдел маркетинга	Заимодавец	ООО "Искра"
Дата начала действия договора	01.02.04	Сумма договора	500.00
Срок договора	300	Дата окончания договора	27.11.04
Процент в рублях	10	Досрочный процент	5
Валюта договора	USD		

At the bottom of the form, there are three buttons: "OK", "Закрыть", and "Печать".

2.4 Создать документ *ПоступлениеЗайма*. Реквизиты табличной части:  
*Заимодавец* (*Справочник.Контрагенты*);  
*Агент* (*Справочник.Агенты*);  
*Договор* (*Документ.ДоговорЗайма*);  
*БанковскийСчет* (*Справочник.БанковскиеСчета*);  
*Валюта* (*Справочник.Валюты*);  
*Сумма* (*число*).

В результате проведения документа формируется запись в дебет счета 51 (или 52 в случае валюты) и кредит счета 66 (соответствующий суб-счет) на сумму полученного займа.

НомерДок: [ ] 0 ДатаДок: [ ]

N	Заимодавец	Договор	Валюта
	Агент	БанкСчет	Сумма

OK      Закрыть

## 2.2 Реализовать автоматизацию выполнения следующих операций (в пользовательском режиме):

- 1) Установить рабочую дату 01.02.04 (меню Сервис – Параметры).
- 2) Заполнить справочник *Валюты* (а именно, установить курс \$ по отношению к рублю на рабочую дату);
- 3) Определить в качестве значения константы *ОсновнаяВалюта* – руб.
- 4) Заполнить справочники *Агенты* и *Контрагенты* (по две записи).
- 5) Ввести 2 первичных документа *Договор\_займа* в базу данных (дата начала договора – 01.02.04, срок договора – 365 дней, в первом документе валюта – рубли, во втором документе - \$).
- 6) Ввести документ *ПоступлениеЗайма*, подтверждающий поступление сумм займа на расчетный счет от указанных заимодавцев (соответственно в рублях и в валюте). Убедиться в проведении данного документа, просмотрев Журнал проводок и Журнал Операций.

## 3 Содержание отчета

3.1 Цель работы.

3.2 Тексты программных модулей.

3.3 Схема связи созданных объектов метаданных.

**Лабораторная работа № 3**  
**Реализация бухгалтерской задачи**  
**«Расчеты по краткосрочным кредитам и займам»**  
**(часть 2)**

**Цель работы:** Развитие постановки задачи, изложенной в лабораторной работе № 2, разработка документа «Закрытие договора займа» с заданными свойствами.

### 1 Краткие теоретические сведения

Для отражения в бухгалтерском учете информации о движении средств в системе 1С:Предприятие используются объекты *Операции* и *Проводки*. Объект *Операция* используется для формирования и анализа проводок, формируемых документом. Для этого у агрегатного объекта *Документ* существует атрибут *Операция* типа «Операция», который обеспечивает доступ к операции данного документа. При этом, для таких документов в режиме Конструктора должен быть установлен признак «*Бухгалтерский учет*». Операции, связанные с документами, могут записываться в Журнал Операций и в Журнал Проводок только при записи или при проведении этих документов. Изменение реквизитов самой операции может производиться в модуле формы документа, и при записи они автоматически будут записаны в операцию, принадлежащую документу. В предопределенной процедуре *ОбработкаПроведения* для записи операции следует вызвать метод операции *Записать*.

#### **Предопределенная процедура модуля формы документа ВводНаОсновании.**

*Синтаксис: ВводНаОсновании(ДокументОснование)*

Вызов этой процедуры происходит неявно в журналах документов при выборе пункта меню Действия – Ввести на основании. В этот момент система подставляет фактическое значение параметра *ДокументОснование*, содержащее документ, на котором находился курсор в момент выполнения данной команды.

Данная процедура может использоваться для установки значений реквизитов нового документа, вводимого на основании другого документа.

Например:

*Процедура ВводНаОсновании(ДокОснование)*

*Если ДокОснование.Вид()= "ПоступлениеМатериалов" Тогда*  
*НоваяСтрока();*

*Реквизит 1=ДокОснование.Реквизит 1;*

...

*КонецЕсли;*

*КонецПроцедуры*

Для проверки выполнения процедуры следует в пользовательском режиме открыть журнал документов, в котором находится *ДокументОснование*, выделить его и выполнить пункт меню Действия – Ввести на основании.

Для вычисления процентов и сумм следует воспользоваться системной математической функцией *Окр* – округлить число. Она возвращает числовое значение результата округления до числа знаков дробной части. Например:

*Сумма = Окр(Сумма\*Процент, 2)*

## 2 Задание для выполнения.

Необходимо внести дополнения в типовую конфигурацию, использованную и доработанную в лабораторной работе № 2.

### 2.1 Создать Документ "ЗакрытиеЗайма" (Закрытие договора займа) с заданными свойствами.

Заголовочная часть документа должна содержать два реквизита:

*Номер;*

*Дата.*

В табличную часть включаются сведения о заимодавцах в разрезе заключенных с ними договоров займа. Реквизиты табличной части:

*Заимодавец (Справочник Контрагенты);*

*Договор (Документ Договор\_Займа);*

*Валюта (Справочник Валюты);*

*Процент (число);*

*Сумма в валюте (число);*

*Рублевое покрытие (число)*

Процент начисляется следующим образом:

- если дата окончания договора займа равна дате создаваемого документа, то  $\text{Процент} = \text{ПроцентГодовых} / 100 * \text{Сумма договора}$ ;

- если дата создаваемого документа меньше даты окончания договора займа (т.е. досрочная выплата займа), то

$\text{Процент} = \text{ДосрочныйПроцент} / 100 / 365 * \text{Сумма договора} * \text{Колич\_Дней}$   
где Количество дней = Дата документа – Дата начала договора займа.

Для получения текущего курса валюты используется метод Валюта.Курс.Получить(ДатаДок). Например:

$\text{Сумма} = \text{Сумма} * \text{Валюта.Курс.Получить(ДатаДок)}$ ;

**Создать документ при помощи:  
механизма ввода на основании;  
интерактивного ввода.**

Необходимо предусмотреть сохранение и проведение документа.

По закрытому договору формируются проводки:

Дебет 91, Кредит 66 (соответствующий субсчет) на сумму начисленного процента;

Дебет 66 (соответствующий субсчет), Кредит 76 (соответствующий субсчет) на сумму займа.

Следует создать бухгалтерский счет 91 «Прочие доходы и расходы» (не групповой), имеющий одну аналитику «ПрибылиУбытки» (перечисление).

Для проведения документа использовать Атрибут документа «Операция».

### 2.2 В пользовательском режиме:

1) ввести в информационную базу два документа «Закрытие займа»:

- один с датой, соответствующей дате окончания договора «Договора займа»;
- второй с более ранней датой, чем дата окончания договора «Договора займа».

2) Сравнить процент начисления.

3) В Журнале Операций и в Журнале проводок просмотреть созданные данными документами проводки.

### Предполагаемый фрагмент Журнала Проводок.

Дебет	Субконто д	Кредит	Субконто к	Вал.	Кол-во	Бал. сумма	Сумма	ИЖ
				Курс	Ед. проводки			
1	Прочие операционные	66.1	ОДО Фармоза Сидоров К.Н. Договор займа 1				20,548	
					Начислен процент			ЭК
2	66.1	ОДО Фармоза Сидоров К.Н. Договор займа 1	76.1	ОДО Фармоза < >		Сумма займа	1,000,000	ЭК
3	91	Прочие операционные	66.2	СП Металл Иванов И.И. Договор займа 3	USD 2160 0000	Начислен процент	8,00	17,280 ЭК
4	66.2	СП Металл Иванов И.И. Договор займа 3	76.2	СП Металл < >	USD 2160 0000	Сумма займа	400,22	864,475 ЭК

### 3 Содержание отчета

#### 3.1 Цель работы.

#### 3.2 Модули формы документа и обработки документа.

## Лабораторная работа №4 Разработка отчетов в системе 1С:Предприятие

Цель работы: Ознакомиться с теоретическими сведениями и получить практические навыки по разработке отчетов в системе 1С:Предприятие, знакомство с конструктором макетов отчетов.

### 1 Краткие теоретические сведения

Любая система автоматизации учета только тогда выполняет свои функции, когда она имеет средства обработки накопленной в системе информации и получения сводных данных в удобном для просмотра и анализа виде. Для решения подобных задач в системе автоматизации учета существует подсистема формирования отчетных документов (или выходных документов).

#### 1.1 Отчеты и обработки в системе 1С:Предприятие

Для обработки информации в системе 1С:Предприятие используются объекты метаданных, называемые *Отчет* и *Обработка*. Объект метаданных этого типа представляет собой алгоритм на внутреннем языке системы 1С:Предприятие. К алгоритму “прикрепляется” диалог, при помощи которого, при необходимости, можно организовать ввод каких-либо параметров, влияющих на ход его алгоритма. Обычно алгоритм имеет и реализованное в табличном редакторе описание печатной формы, которое используется для вывода результатов выполнения алгоритма на экран и принтер. В некоторых случаях одна и та же таблица используется и для ввода параметров, и для вывода результатов.

*Отчеты* предназначены для получения информации, накопленной в системе и используемой для обобщения и формирования итоговых результатов в различных разрезах.

*Обработки* предназначены для выполнения различных действий над информацией, в результате чего информация, хранящаяся в системе, претерпевает какие-либо изменения.

С точки зрения настройки, эти два типа объектов метаданных совершенно одинаковы. Однако рекомендуется алгоритмы, выполняющие какие-либо изменения в информационной базе, размещать на ветви “Обработка” и назначать права на использование обработок только наиболее подготовленным пользователям системы.

#### 1.2 Управление списком отчетов

Вся работа с объектами метаданных типа *Отчеты* и *Обработка* ведется в окне *Конфигурация — Метаданные*. Свойства отчета редактируются в палитре свойств *Свойства Отчета*. Приемы редактирования свойств отчета совпадают с общими приемами редактирования свойств объекта метаданных, изложенными в работе №4. Фактически, объект метаданных типа *Отчет (Обработка)* представляет собой только форму: модуль формы является алгоритмом построения отчета, диалог позволяет задавать параметры формирования отчета, а таблица служит для выдачи результатов построения отчетов в требуемом виде. У объектов метаданных типа *Обработка* таблица чаще всего отсутствует, а для отчетов, напротив, может заменять собой диалог. В процессе работы с отчетами в поль-

зовательском режиме существует возможность “сохранить настройку” — запомнить текущие значения реквизитов диалога отчета.

### 1.3 Внешние отчеты (обработки)

Внешним отчетом (обработкой) в системе 1С: Предприятие называется отчет (обработка), хранящиеся вне конфигурации, в отдельном файле внешнего отчета (обработки). Внешний отчет служит для решения тех же задач, что и объекты метаданных типа *Отчет* или *Обработка*.

Внешние отчеты (обработки) не обладают в полной мере всеми свойствами, присущими объектам метаданных. Так, внешние отчеты (обработки) не имеют идентификаторов, и обращение к ним, например, при создании пользовательского интерфейса, выполняется по имени файла внешнего отчета (обработки).

Основное преимущество внешнего отчета (обработки) - возможность его проектирования и отладки в процессе работы системы 1С: Предприятие. В этом случае разработка и отладка отчета (обработки) значительно ускоряются: редактирование и сохранение внешнего отчета (обработки) выполняется в режиме Конфигуратора, без сохранения конфигурации в целом, а запуск — в режиме “1С:Предприятие”. Для выполнения внешнего отчета (обработка) загружается при помощи меню “Файл” и работает так же, как и любой другой отчет конфигурации.

Любой объект метаданных типа *Отчет* или *Обработка* может быть скопирован в файл внешнего отчета (обработки) и наоборот — форма объекта метаданных может быть заменена формой внешнего отчета (обработки). Для этого необходимо открыть для редактирования форму отчета и в меню “Файл” главного меню Конфигуратора выбрать пункт “Сохранить как...”. В стандартном диалоге сохранения файла, следует выбрать тип файла *Внешний отчет (\*.ert)* и указать имя для сохраняемого внешнего отчета (обработки). По умолчанию внешний отчет (обработка) предлагается сохранить в подкаталоге ExtForms, который автоматически создается в каталоге с информационной базой. Редактирование внешнего отчета (обработки) выполняется в Конфигураторе.

*Добавление внешнего отчета в структуру метаданных*: Существующий внешний отчет (обработка) может быть вставлен в структуру метаданных как новый объект метаданных типа “Отчет” или “Обработка”. Для этого необходимо в структуре метаданных выделить наименование любого объекта метаданных типа “Отчет” или “Обработка” и использовать пункт “Вставить внешний отчет (обработку)” контекстного меню объекта метаданных. В выданном на экран стандартном диалоге открытия файла необходимо выбрать тип файла “Внешний отчет (обработка) (\*.ert)” и указать имя файла внешнего отчета, который требуется вставить в структуру метаданных. В результате этих действий в дереве метаданных появится новый отчет (или обработка). Новому объекту метаданных будет присвоен условный идентификатор, состоящий из слова “Новый” и числа — порядкового номера нового объекта метаданных.

### 1.4 Конструктор макета отчета

Назначение Конструктора макета отчета - автоматическое создание шаблона печатной формы объекта метаданных. В результате работы Конструктора для объекта метаданных будет создан табличный доку-

мент с заданными секциями, а в модуль формы объекта метаданных будут внесены команды для их вывода в отчет (печатную форму).

#### *Запуск Конструктора макета отчета.*

Конструктор макета отчета можно запустить двумя способами. Он может быть запущен индивидуально, либо в процессе работы Конструктора отчетов (при создании отчета с применением Конструктора отчетов этот Конструктор на одном из этапов создания отчета предложит вызвать Конструктор макета отчета).

При редактировании формы объекта метаданных Конструктор макета отчета можно вызвать, выбрав пункт "Макет отчета" из меню "Конструкторы" главного меню программы. Перед запуском Конструктора макета отчетов следует открыть ту форму объекта метаданных, из которой должна строиться создаваемая печатная форма, например, форму элемента справочника или форму списка.

При запуске Конструктора макета отчетов появляется диалог "Новая процедура", предлагающий ввести имя процедуры, которая будет выполнять построение печатной формы. По умолчанию это имя "Сформировать". Имя процедуры можно изменить. Если указанное имя процедуры совпадает с именем одной из процедур, уже присутствующих в модуле формы (например, в данном случае, в справочнике валют уже была процедура "ПостроитьОтчет"), то при нажатии кнопки "ОК" было выдано сообщение ("Процедура ПостроитьОтчет уже существует"). Сообщение выдается также при вводе названия, недопустимого в качестве имени процедуры.

#### *Выбор вида отчета*

Если совпадения имен не произошло, при нажатии "ОК" откроется диалог "Конструктор макета отчета", в котором нужно указать заголовок отчета. По умолчанию заголовок совпадает с названием процедуры. Заголовок будет выдаваться в верхней части печатной формы, поэтому рекомендуется отредактировать его таким образом, чтобы заголовок отражал содержимое создаваемой печатной формы.

Затем следует выбрать принципиальную схему построения отчета:

- отчет состоит из отдельных значений;
- отчет содержит циклы по строкам;
- отчет содержит циклы по строкам и по столбцам;
- отчет содержит только заголовки.

В любой из отчетов можно также включить диаграмму, установив флажок "Включить диаграмму".

После выбора принципиальной схемы построения отчета и нажатия на кнопку *Далее* появляется диалог "Конструктор макета отчетов: Параметры". В зависимости от выбранного типа отчета Конструктор будет запрашивать в этом диалоге различный набор параметров, необходимых для создания макета.

*Отчет состоит из отдельных значений.* Отчет из отдельных значений используется для печати реквизитов, существующих в единственном числе. Например, отчет такого типа удобно использовать для печати формы элемента справочника или реквизитов шапки документа. В поле "Количество выражений" нужно ввести число, равное тому количеству значений, которое требуется получить. По умолчанию это число равно трем.



*Отчет содержит циклы по строкам.* Такой вид отчета удобно использовать для объектов метаданных, имеющих табличную часть: документов с табличной частью, журналов документов, форм списков справочников.

В диалоге "Конструктор макета отчета: Параметры" в поле "Количество циклов по строкам" нужно ввести число, значение которого равно необходимому количеству циклов (под "циклами" понимаются операторы встроенного языка Пока...Цикл...КонецЦикла). По умолчанию это число равно единице, то есть вложенные циклы отсутствуют. В поле "Количество колонок" следует ввести число, значение которого равно необходимому количеству колонок. По умолчанию это число равно трем.

*Отчет содержит циклы по строкам и столбцам.* Такой вид отчета используется, если заранее не известно ни число строк, ни число столбцов. Например, это может быть шахматная ведомость, или, отчет по реализации товаров, представляющий собой таблицу, строки которой соответствуют товарам, а столбцы - контрагентам, которые их приобрели. В диалоге "Конструктор макета отчета: Параметры" необходимо ввести числа, определяющие следующие параметры макета отчета. "Количество циклов по строкам" и "Количество циклов по колонкам" по умолчанию равны 1, то есть отсутствуют вложенные циклы. Параметр "Количество фиксированных колонок слева" задает количество постоянных колонок отчета, которые будут выводиться до выполнения цикла по колонкам. Например, это могут быть колонки для вывода кода товара и его наименования. "Количество фиксированных колонок справа" задает количество колонок, которые должны выводиться после выполнения цикла по колонкам. Например, такими фиксированными колонками могут быть колонки для вывода итоговых значений количества и суммы приобретенного товара.

*Отчет содержит только заголовок.* Этот вид отчета имеет наиболее общее применение и применим для быстрого создания "заготовки" будущего отчета. Такой вид отчета можно использовать, например, если требуется отчет в виде диаграммы (диаграмму можно вставить в любой из видов отчета).

#### *Завершение работы Конструктора.*

На завершающем этапе работы Конструктора задается порядок вызова процедуры формирования печатной формы и режим открытия готовой печатной формы.

- *Группа Вызов процедуры* позволяет задать порядок вызова процедуры формирования отчета.
- *Не вставлять* - диалог формы объекта метаданных не будет содержать кнопки для построения печатной формы;
- *Новая кнопка* - в диалоге появится новая кнопка, надпись на ней по умолчанию совпадает с именем процедуры, определенным при запуске Конструктора;
- *Выбрать кнопку* - из выпадающего списка можно выбрать название одной из кнопок, уже существующих в диалоге.

Если установить флажок *Режим "только просмотр"*, созданная печатная форма будет открываться только для просмотра, и для редактирования печатной формы пользователю необходимо будет отключить этот режим.

## 1.5 Результаты работы Конструктора

В результате работы Конструктора макета отчета в форме объекта метаданных появляется новый лист типа “Таблица” с именем процедуры, определенным при запуске Конструктора.

Таблица содержит секцию “Заголовков” для вывода заголовка отчета, указанного в процессе работы Конструктора. Состав прочих секций отчета зависит от выбранного типа отчета.

*Отчет состоит из отдельных значений.* В этом случае созданная таблица состоит из одной секции — “Шапка”, и содержит заготовки для вывода значений. В каждой строке с заготовками следует в ячейку “Текст” ввести подпись к выводимому значению, а в ячейку “Выражение” - выражение для вычисления выводимого значения. В результате работы конструктора в модуль формы добавлена процедура для вывода отчета с указанным при работе конструктора именем.

*Отчет содержит циклы по строкам.* Таблица для отчета этого типа содержит секции для вывода шапки и подвала отчета, а также столько секций “Строка...”, сколько циклов по строкам было указано при работе Конструктора.

*Отчет содержит циклы по строкам и столбцам.* Для отчета такого типа Конструктор помещает в табличную часть вертикальные и горизонтальные секции. Горизонтальные секции включают секции для вывода заголовка отчета, шапки колонок отчета, подвала отчета, а также столько секций “Строка...”, сколько циклов по строкам было указано при работе конструктора. Вертикальные секции включают секции для вывода фиксированных колонок слева и справа, а также столько секций “Колонка...”, сколько циклов по колонкам было указано при работе Конструктора. Имя одной из них, “Сформировать”, было задано при работе Конструктора. Процедура содержит команды создания и вывода таблицы, а также цикл по строкам. Внутри цикла по строкам вызывается процедура Сформировать\_Колонки (имя процедуры сформировано автоматически), которая “отвечает” за цикл по колонкам. В обоих циклах не определены условия циклов и Конструктор напоминает об этом при помощи подсказки. Условия выполнения циклов необходимо ввести вручную.

## 2 Задание для выполнения

Обеспечить формирование следующего отчета.

1) Сведения о договорах займа, завершенных до истечения установленного срока в период с . . . по . . .

Период, за который формируется отчет, может быть произвольным и задается в диалоге пользователем.

**Сведения о договорах займа, завершенных до истечения установленного срока в период с . . . по . . .**

Займодавец	Агент	Договор	Дата начала договора	Срок займа по договору (в днях)	Фактическая дата завершения договора	Фактический срок займа (в днях)	Сумма займа	Начислен процент
------------	-------	---------	----------------------	---------------------------------	--------------------------------------	---------------------------------	-------------	------------------

**3 Контрольные вопросы**

- 3.1 Что такое отчет, что такое обработка?
- 3.2 Порядок создания отчета. Конструктор Макета отчета.
- 3.3 Виды отчетов при использовании Конструктора Макета отчета.
- 3.4 Внешние отчеты, порядок их создания.

**4 Содержание отчета**

- 4.1 Цель работы.
- 4.2. Тексты алгоритмов на встроенном языке.
- 4.3. Выводы по работе

Учебное издание

Составители: Гладышева Елена Львовна  
Селезнёв Пётр Владимирович

## **Методические указания**

к выполнению лабораторных работ по курсу  
**«Программные средства систем обработки данных»**  
для студентов специальности 53.01.02

**«Автоматизированные системы обработки информации»**

**Часть 2**

Ответственный за выпуск: Гладышева Е.Л.

Редактор: Строкач Т.В.

Корректор: Никитчик Е.В.

Компьютерная верстка: Боровикова Е.А.

---

Подписано к печати 8.11.2004 г. Формат 60\*84<sup>1/16</sup>. Бумага писчая.  
Гарнитура Arial. Усл. п. л. 1,6. Уч. изд. 1,75. Заказ № 1098. Тираж 100 экз.  
Отпечатано на ризографе Учреждения образования «Брестский государствен-  
ственный технический университет». 224017, г. Брест, ул. Московская. 267.