




«САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ»

Кафедра информационных систем и вычислительной техники

Допущены
к проведению занятий в 2018-2019 уч. году
Заведующий кафедрой

 **Е.Б. Мазаков**
«26» апреля 2018 г.

УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

для курсового проектирования по учебной дисциплине

«БАЗЫ ДАННЫХ»

Направление подготовки:

09.03.01. «Информатика и вычислительная техника»

Профиль программы:

Автоматизированные системы обработки информации и управления

Разработал: *доцент Копейкин М.В.*

*Обсуждены и одобрены на заседании кафедры
Протокол № 9 от 26 апреля 2018 г.*

САНКТ-ПЕТЕРБУРГ
2018

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Санкт-Петербургский горный университет

БАЗЫ ДАННЫХ

*Методические указания по курсовому проектированию для студентов
бакалавриата направления подготовки 09.03.01*

Санкт-Петербург

2017

УДК 681.3

Базы данных: Методические указания по курсовому проектированию / Национальный минерально-сырьевой университет «Горный». Сост.: М.В. Копейкин, В.В. Спиридонов, Е.О. Шумова. - СПб.: Изд-во , 2017. – 152 с.

Методические указания разработаны в соответствии с федеральным государственным образовательным стандартом высшего профессионального образования и рабочей программой.

Дисциплина посвящена изучению вопросов функциональной и структурной организации баз данных. В основных ее разделах изучаются принципы построения, функционирования и оценки характеристик баз данных и СУБД, отдельных ее компонент и блоков.

Курсовой проект посвящен практическому изучению вопросов организации баз данных и систем управления базами данных.

Методические указания предназначены для студентов направления 09.03.01. «Информатика и вычислительная техника»

Научный редактор проф. И.В. Иванова

© Национальный минерально-сырьевой университет «Горный», 2017

Введение

Курсовой проект выполняется на четвертом курсе. Исходными данными для проектирования является информация предметной области определяемая студентом самостоятельно по согласованию с преподавателем.

1. ОБЩИЕ ПОЛОЖЕНИЯ

Практическое изучение дисциплины “Базы данных” завершается выполнением курсового проекта. Лабораторные занятия позволили закрепить полученные на лекциях знания и выработать необходимые навыки для решения практических задач.

Цель курсового проекта - расширить и углубить эти навыки, научить студента самостоятельно проектировать программные комплексы на основе концепций баз данных, отражающих работу сложных систем (проводить анализ работы сложных систем, разбивать задачу на комплекс подзадач, составлять для них алгоритмы и программы, объединяя все в единый программный комплекс).

Основная особенность курсового проекта заключается в том, что в ее рамках решаются задачи, близкие к производственной деятельности студента. Поэтому содержательная часть проектируемой информационной системы не регламентируется и выбирается студентом самостоятельно.

Курсовой проект может служить основой для дипломного проектирования. В этом случае структура работы остается, однако все вопросы прорабатываются глубже.

При выполнении курсового проекта студент должен руководствоваться указаниями по оформлению [1], а также:

- заданием на курсовой проект;
- методическими указаниями;

- указаниями руководителя;
- учебными пособиями и другой рекомендованной литературой.

Курсовая работа должна продемонстрировать способность студента к самостоятельному и творческому использованию полученных знаний.

Окончательно оформленная и подготовленная к сдаче работа должна содержать:

- титульный лист;
- оглавление;
- задание на курсовой проект;
- расчетно-пояснительную записку;
- приложение.

На титульном листе указывается:

- наименование и тема проекта;
- дисциплина ("Базы данных");
- данные о студенте (Ф. И. О. ,шифр, группа);
- преподаватель (Ф. И. О.);
- оценка;
- дата представления.

В приложении приводятся тексты программ и протоколы их работы (результаты решения тестовых задач).

При написании пояснительной записки желательно в максимальной степени использовать принтер ПЭВМ.

Студенту разрешается (по согласованию с преподавателем) подготовить и оформить пояснительную записку на дискетке с выполнением указанных выше требований (без предоставления распечатанной расчетно-пояснительной записки).

2. ТЕМАТИКА КУРСОВОГО ПРОЕКТА

Тематика подразделяется: на типовую и выбираемую студентом самостоятельно. В типовом проекте предлагается разработать или выбрать

комплекс моделей структур данных и программ для автоматизированного рабочего места (АРМ) информационной модели предметной области, функционирующей в рамках объектно-реляционной системы управления базами данных – ORD (**O**bject **R**elation **D**ataFrame **D**ynamic structure).

Во втором случае могут быть использованы СУБД типа: INTERBASE, FOXBASE, ALASKA, ORACLE, ACSESS, INFORMIX и многие другие, поддерживающие реляционные принципы в структурах отображаемой информации.

В случае типового проекта прикладная часть предметной области для разрабатываемой информационной подсистемы выбирается студентом из множества типовых ситуаций (фреймов), предоставляемых системой ORD и адаптацией их под свои условия или студент может выбирать тематику самостоятельно из других областей, характеризующих специфику его предметной области.

Во втором случае основанием выбора служат особенностями производственной деятельности студента, однако все требования указанные для типового варианта также должны быть реализованы.

В обоих случаях проектировщик должен продемонстрировать работоспособность разработанной подсистемы на вычислительной технике, имеющейся на кафедре.

Каждое задание на курсовой проект уточняется с преподавателем.

3. ЗАДАНИЕ НА ТИПОВОЙ КУРСОВОЙ ПРОЕКТ

Необходимо разработать (выбрать) комплекс моделей структур данных для реализации АРМ оператора на базе системы распределенной обработки информации реального времени - системы ORD. Область действия (интересов) оператора (кадры, финансы, расчет заработной платы, складское хозяйство, аптека, музей, стадион, больница и т.д.) выбирается студентом самостоятельно.

В пояснительной записке должны быть отражены следующие вопросы:

- техническое задание;
- разработка (уточнение типового меню) управляющей части АРМ;
- разработка (выбор) структур моделей баз данных для АРМ;
- проектирование (выбор) прикладных программ обработки хранимой информации;
- тестирование, доказательство работоспособности в условиях динамического изменения предметной области, для которой реализуется проект;
- использование стандартных средств системы, обеспечивающих работоспособность информационной модели в условиях развития предметной области: генератора незапланированных запросов (SQL или QBE), генератора печати, архивация информации, циркулирующей в предметной области.

Большинство существующих СУБД предполагают централизованную обработку хранимой информации, что накладывает ограничения не только на процесс проектирования информационной модели в целом, но и на будущее ее развитие. Поэтому одной из целей данного курсового проекта является получение навыков проектирования не только централизованных информационных систем, но и систем распределенной обработки информации, например, на базе системы ORD, позволяющей проектировать информационные системы со следующими свойствами:

- повышенной гибкостью, модульностью, реактивностью и надежностью;
- открытой архитектурой и, как следствие этого, повышенной мобильностью и адаптируемостью под требования заказчика с учетом распределенной обработки информации.

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Далее рассматривается содержание вопросов, излагаемых в пояснительной записке.

4.1. Техническое задание (ТЗ)

ТЗ отражает требования, предъявленные пользователями - заказчиками к разрабатываемой системе. Оно представляет собой описание совокупности характеристик, которым должна удовлетворять создаваемая система, в том числе по функциональному, программному и математическому, техническому, информационному и организационному обеспечению. Таким образом, ТЗ отвечает на вопросы, какой должна быть проектируемая система и что должно дать ее внедрение для улучшения работы экономического объекта.

Согласно ГОСТ 24.201-79 ТЗ включает следующие разделы:

- введение;
- характеристика объекта;
- назначение проектируемой системы;
- основные требования к системе;
- технико-экономические показатели;
- состав, содержание и организация работ по созданию системы;
- порядок приемки системы;

В первом разделе приводятся полное наименование системы, наименование организаций-участников проектирования, сроки начала и окончания работ. Во втором разделе дается описание состава объекта управления, его производственная структура, характеристики входных и выходных информационных потоков. Третий раздел ТЗ содержит описание назначения системы, основы ее создания, критерии эффективности функционирования объекта в условиях автоматизации, перечень выполняемых функций, описание новой структуры в условиях автоматизации.

Содержание остальных разделов понятно из их наименования.

В курсовом проекте не обязательна тщательная проработка всех разделов, так как студент описывает упрощенный вариант объекта управления и соответствующего ему автоматизированного рабочего места.

Однако если работа служит основой дипломного проекта, то в дальнейшем необходимо развить все разделы ТЗ.

Учитывая, что разрабатываемая информационная подсистема является только частью автоматизированной системы управления предприятием, необходимо уже на начальном этапе разработки технического задания (и инфологического этапа проектирования моделей) выбрать среди множества существующих систем управления базами данных СУБД, позволяющую в полной мере удовлетворить требования настоящего и будущих периодов проектирования информационной модели Вашего предприятия.

Выбор базовой СУБД является очень ответственной задачей, определяющей в конечном итоге информационное развитие Вашего предприятия, и является самостоятельной задачей, рассматривать которую необходимо, если курсовой проект служит основой дипломного проекта.

Студентам для курсового проектирования рекомендуется воспользоваться системой ORD, т.к. она предоставляет методику создания единой интегрированной информационной модели предметной области любого профиля, сохраняя (при небольших затратах) разработанное прикладное программное обеспечение при развитии (наращивании) предметной области.

Система может функционировать как на одной машине, так и на сети ЭВМ, причем в последнем случае, она обладает рядом уникальных возможностей, например: - возможностью параллельной работы одной и той же информационной подсистемы в различных узлах сети, без нарушения целостности и достоверности хранимых данных;

- возможностью работать как в одно-серверном, так и много серверном варианте, используя при этом в качестве серверов любые ЭВМ сети;

- возможностью управлять распределением обработки, информации не нарушая целостности и работоспособности всей информационной системы в целом и т.д.

Для целей обучения важно, что в системе сразу предоставляется набор АРМов, учитывающих хозяйственную деятельность некоторого предприятия (с точки зрения движения в нем материальных, трудовых ресурсов и финансовых средств и т.д.). Например: основные средства ("О"), материальные ценности ("М"), учет труда и начисление заработной платы ("Т"), учет финансово-расчетных операций ("F"), учет затрат на производство и себестоимость продукции ("J"), учет и реализация готовой продукции ("G"), баланс предприятия ("В") и любые другие необходимые в данной предметной области подсистемы, встраиваемые разработчиком в оболочку целевой СУБД.

Если студент выбирает в качестве типового проекта какой-то существующий АРМ, то ему необходимо адаптировать его под свою предметную область и провести необходимое тестирование.

Ниже рассмотрен пример курсового проекта, если студент выбирает информационную подсистему, не зафиксированную в стандартной поставке ORD.

Целью создания АРМ является автоматизация сбора и обработки информации, циркулирующей в объекте управления. При разработке подсистемы следует учитывать, что планируемый АРМ является частью общего АСУ предприятия и, в конечном итоге, разработанная подсистема будет функционировать в сетевом окружении (на нескольких машинах, как какая-то часть единой интегрированной информационной модели предметной области).

Суть фиксируемой ситуации в информационной модели, рассмотрим на фрагменте небольшого АРМ для склада мебельной фурнитуры. На складе хранятся разные материалы и комплектующие, необходимые изготовителям мебели: ткани для обтяжки мягкой мебели, гвозди, ручки, скобы, колесики и много-много всякой другой мелочи и не очень.

Владельцы фирмы (назовем ее "МеДина Лтд"), в состав которой входит склад, закупают все эти товары у оптовых поставщиков или производителей, а

затем используют их для производства мебели в мебельных цехах своей же фирмы или продают их мелкими партиями или даже в розницу своим клиентам. При этом им приходится решать определенные учетные и управленческие задачи, в общем, характерные для складского дела, но имеющие некоторые особенности, связанные, в основном, с видами хранимых на складе товаров.

В качестве программного обеспечения для реализации подобной задачи вполне можно использовать средства известных программных систем типа 1С, Галактика, Парус и т.д. Однако некоторые особенности работы с такими товарами как рулоны тканей и специфика их учета требуют заметных усилий по адаптации подобных систем к данному случаю.

Классическая схема разработки бизнес приложений предполагает, что на начальном этапе, в той или иной форме, описываются бизнес процессы фирмы, для которой будет создаваться программное обеспечение.

Как следует из сказанного выше, основных процессов в фирме “МеДина Лтд” три. Это:

- закупка и складирование товаров и комплектующих,
- производство мебели и
- продажа произведенной продукции и комплектующих (в том числе и покупных).

Собственно говоря, эти три бизнес процесса являются своего рода стандартом для всех фирм, имеющих свое небольшое производство.

В примере курсового проекта иллюстрируются (частично) первый и третий процессы, т.е., закупка, складирование и продажа товаров и продукции.

Пусть в процессе инфологического этапа проектирования модели выделены три основных документа (формы) в которых отражается *приход*, *расход* и *учет* товаров используемых фирмой: НАКЛАДНАЯ ПРИХОДА и РАСХОДА и ТОВАРНАЯ КАРТОКА (карточку для простоты разделим на две: для тканей и штучных товаров).

Естественно для заполнения НАКЛАДНОЙ ПРИХОДА и РАСХОДА потребуется дополнительная справочная информация, подлежащая хранению в базе: ТОВАР, ОРГАНИЗАЦИИ (клиенты, контрагенты, поставщики и пр.), ПОДРАЗДЕЛЕНИЯ фирмы, СОТРУДНИКИ и БАНКИ и многое другое потому, что материалы приходится измерять и учитывать. Их особенности, с этих точек зрения, отражаются ВИДАМИ ПРИХОДА, ГРУППАМИ МАТЕРИАЛОВ и ТОВАРОВ, ЕДИНИЦАМИ ИЗМЕРЕНИЯ и некоторыми другими особенностями (например, ВАЛЮТА). Перечисления (или списки) всех этих видов, групп и единиц в системе ORD обычно относят к категории справочников.

Естественно, в проектируемой подсистеме, не обойтись и без программирования не только для формирования всевозможных отчетов (запросов) по накопленной информации (для этого полезен язык SQL или QBE), но и обычного программирования, связанного, например, с формированием прайс-листов (на основе анализа продаж).

АРМ, как обычно, помимо справочников, документов и процедур, должен включать в себя и некоторые сервисные утилиты (защита, архивация и т. д.).

Большого для АРМ и не надо. Конечно, любой знающий студент спросит: “А где же все остальное: финансовые документы, счета-фактуры, журналы, шахматки и прочее? Где проводки, гроссбух (он же главная книга), баланс, отчетные формы, расчеты налогов и все другие не менее полезные и необходимые вещи?” Конечно, все это должно быть, но это не дипломный проект. ORD – это, прежде всего, определенная идеология и инструмент построения баз данных, ознакомление с которыми и является основным назначением данного пособия.

4.2. Разработка управляющей части программного комплекса

Цель этого раздела состоит в том, чтобы на основании содержательного анализа предметной области и состава функциональных задач разложить их на частные подзадачи. При этом удобно использовать стратегию проектирования, которая в литературе носит название "сверху вниз", а также "нисходящая" или "пошаговое совершенствование" [2]. Удобнее всего эту стратегию реализовать

в виде системы меню. Каждая строчка меню предлагает обращение либо к меню следующего уровня, либо к решению определенной задачи. Т.е. система меню должна позволять по своим элементам определить не только назначение самого АРМ, но и раскрыть состав и функциональное назначение его составляющих (см. Приложение Б, рис. Б10).

Для нашего примера набором таких меню может быть (рис. 1).

Каждый пункт меню, в свою очередь, может содержать необходимую группу подменю, раскрывающую суть данного пункта. Например, пункт "Информация" может содержать текстовую справку о назначении данного АРМ и особенностях его использования в общей группе АРМов, обслуживающих данное предприятие.

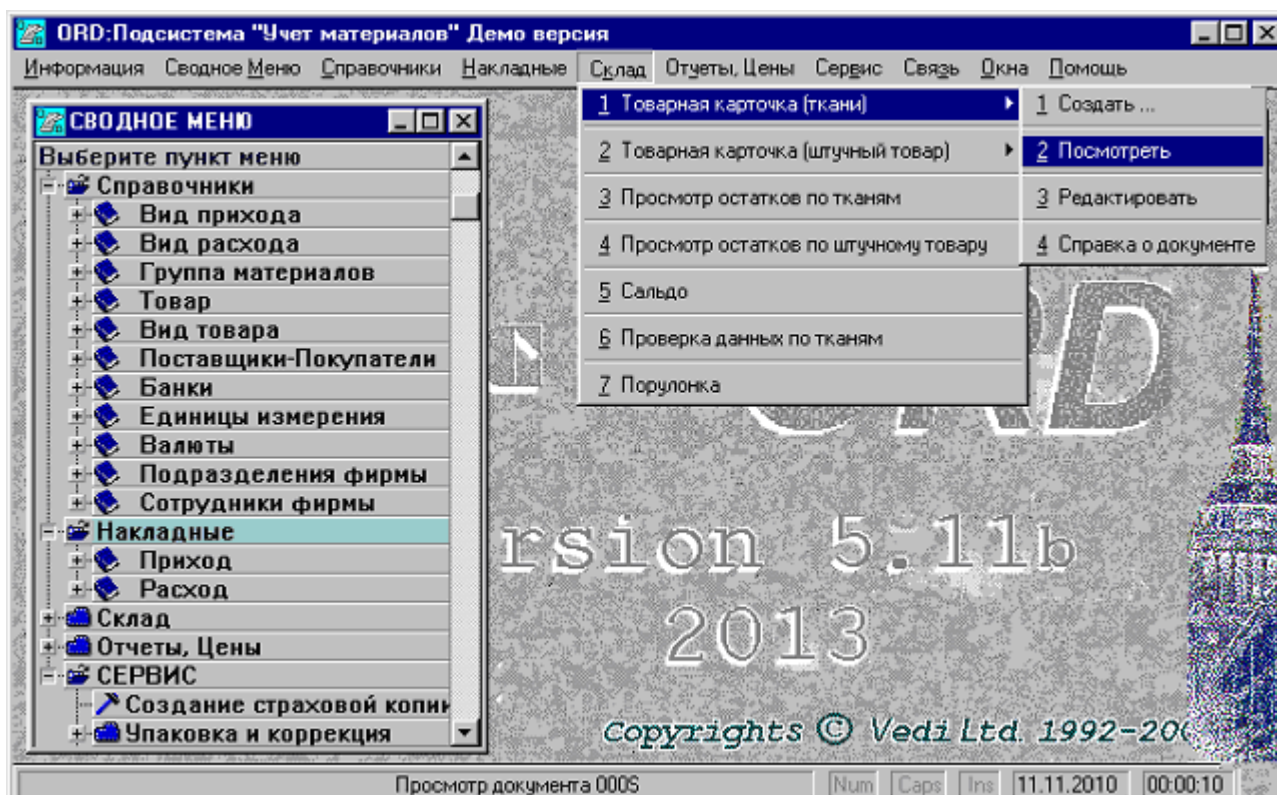


Рис. 1 Возможные варианты меню

Пункт "Связь", в свою очередь, может содержать подчиненное меню следующего состава:

Связь

1. Связь с поставщиками по модему.
2. Связь с удаленным сервером.

3. Выход.

Подчеркнем, что расположение информации по уровням меню и их наименование должно помогать пользователю АРМ определиться, откуда появляется данная информация и кем и как она используется. Так, например, анализ пунктов меню "Товарная карточка ткани" (основное информационное содержание подсистемы) позволяет сделать следующее заключение. Данный документ (рис. 15) может корректироваться пользователем, если он обладает для этого соответствующими правами. Причем, именно последовательность расположения элементов меню и должна характеризовать ситуацию предметной области, которую необходимо отразить в информационной модели. (Если наименования пункта меню не достаточно, для выражения сути происходящего, то необходимо обеспечить соответствующий Help данного пункта. Вызов данного Help, будет всегда обеспечиваться системой при нажатии клавиши F1.)

Пункт "Отчеты Цены" головного меню предназначен для построения необходимых графиков и отчетов по подсистеме. Пункт 7 головного меню "Сервис" содержит следующее подменю:

1. Создание страховой копии
2. Чистка баз данных
3. Архивация документов подсистемы
4. Работа с архивом
5. Поиск по запросам

Каждый пункт приведенного выше меню и пункта Связь имеет свое многоуровневое подменю, назначение, и состав которых приводится в технической документации системы и необходимость в их разработке отпадает, т.к. все функции указанных пунктов меню поставляются с системой ORD.

Писать функцию для формирования любого пункта меню (как это принято в большинстве оболочек для создания подобных АРМ) тоже нет необходимости. При создании документов система автоматически формирует

пункт меню, где отражается доступ к данному объекту системы. Данный процесс был рассмотрен подробно в одной из лабораторных работ. Для DOS варианта системы указанные вопросы рассмотрены в [2].

Практически все необходимые Вам функции представлены в **открытых** библиотеках системы ORD. Help системы и техническая документация по системе содержит множество примеров и иллюстраций по способам использования данных функций.

Напомним, что система ORD состоит из двух основных частей:

- системы управления распределенными базами данных ORD.exe;
- конструктора создания модели предметной области Db.exe.

Указанные части могут функционировать как независимо друг от друга, так и являться единым целым.

Перед прочтением данного пособия система должна быть инсталлирована. Необходимый материал предоставляется на кафедре или находится по адресу WWW.ORD.COM.RU.

СУБД ORD запускается на исполнение модулем t.exe или ORD.exe, который находится в папке информационной подсистемы.

Система инструментальной оболочки создания модели предметной области (конструктор) запускается на исполнение модулем Db.exe, который может находиться в любой директории (см. командный файл d.bat или S.bat). Все необходимые командные файлы и исходные тексты программ приведены в текущей папке, куда инсталлировалась система. Запуск исполняемого модуля в любой информационной подсистеме осуществляется посредством T.exe (Windows) и C.exe (DOS).

Модуль t.exe формируется с помощью командного файла 3.bat. Все Ваши новые (или модифицированные) программы, которые Вы хотите использовать в оболочке ORD должны быть скомпилированы с помощью командного файла Cm.bat и прикомпонованы с помощью 3.bat.

Любая Ваша программа может быть реализована на любом известном Вам языке программирования и вызвана на исполнение из оболочки системы ORD в виде файла *.com или *.exe. В любой момент работы системы Вам доступна контекстная помощь по клавише F1 или помощь по вопросам проектирования информационной модели - клавиша F12.

4.3. Основные определения, используемые в системе

Перед ознакомлением с основными теоретическими положениями системы ORD, рекомендуем установить систему на Вашей машине и ознакомиться с системой, запустив на исполнение модуль ORD.exe.

Отметим, что Вам предоставлена возможность, ознакомиться с функционированием системы, как на реальной информационной модели промышленного предприятия, так и версия системы, подробно описывающая пример построения Вашего курсового проекта для части предметной области, получившей условное название "Учет материалов" (информация указанной подсистемы находится в папке SUBA).

Структура распределения информации по папкам приведена на рисунке 37.

В процессе ознакомления с системой Вам доступна вся информация по методологии проектирования распределенных информационных систем в среде ORD, которая может быть использована для других СУБД реляционного типа.

Поэтому, здесь не приводится полное описание методики и методов проектирования и описания моделей, используемых в системе.

Также не полностью раскрыты определения и некоторые понятия, используемые в системе ORD. В случае возникновения вопросов по некоторым терминам рекомендуем обратиться к соответствующему поставляемому техническому руководству.

Для лучшего понимания излагаемого материала приведем одну из форм (документов), используемых в учебном проекте, а именно документ СОТРУДНИКИ ФИРМЫ (рис. 2), в котором представлена информация, характеризующая людей работающих на фирме.

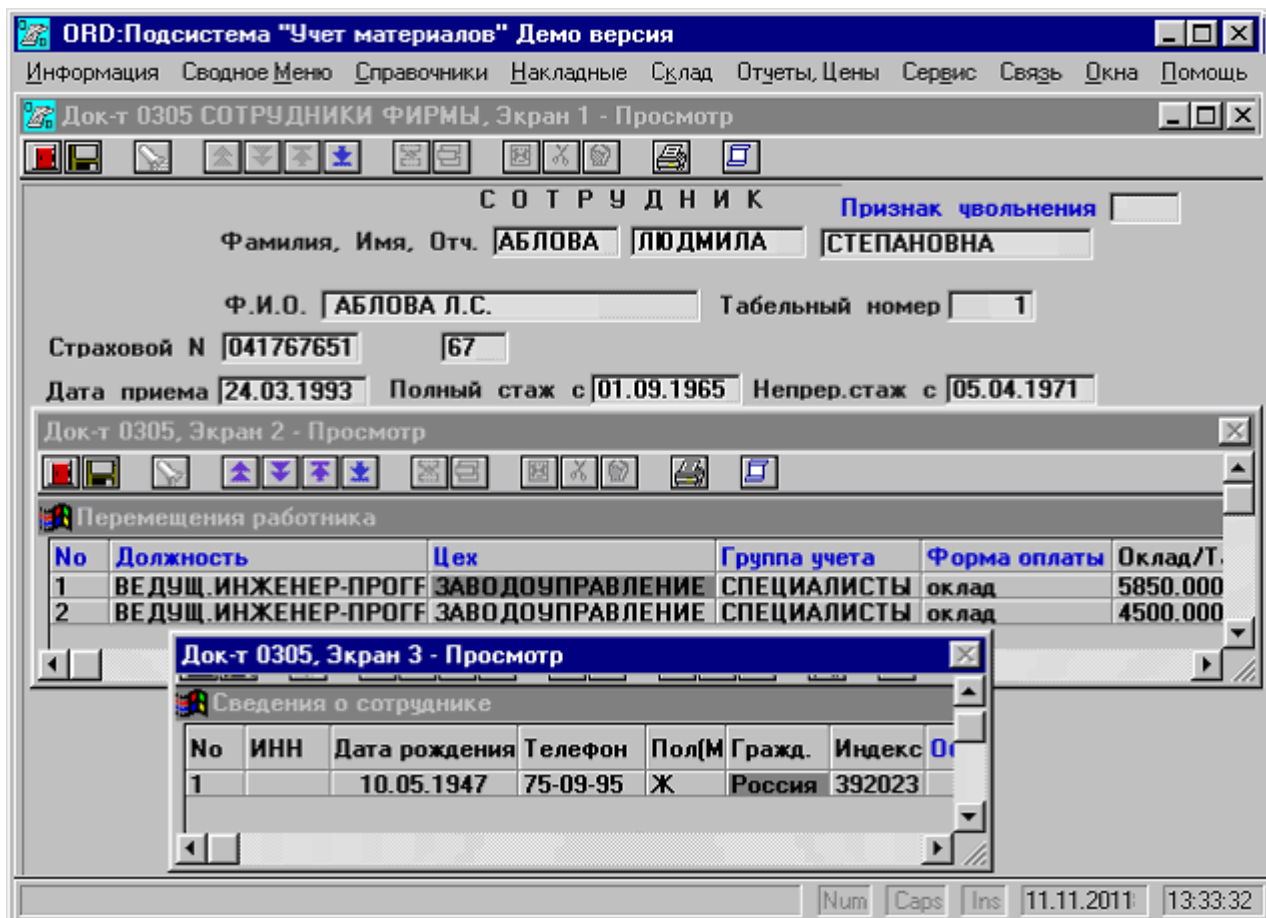


Рис. 2 Изображение внешней модели на экране

Представленная форма на рис. 2 является отображением инфологической модели (ILM) на концептуальную модель системы [3]

На концептуальном уровне ILM отображается в универсальный набор формальных моделей. В *ORD* таким набором моделей является набор фреймов. Фрейм – типовая иерархически упорядоченная структура, необходимая для представления объекта (класса объектов) и/или связи (класса связей) (т.е. структура данных для представления некоторых стандартных (типовых) ситуаций объективной действительности). Фрейм ассоциируется в ILM с понятием иерархической сущности (см. [6], пункт 1.2) с изменяемой (динамической) структурой. При этом объект (или связь), использованный в иерархической сущности, в свою очередь, может являться фреймом.

Основной постулат подхода - любая ситуация в предметной области (ILM), подлежащая отображению в системе, может быть представлена в виде размеченного графа с изменяемой структурой [6].

Для уникальной идентификации понятия отображаемого в ILM и соответствующей ей концептуальной модели, используется понятие объектного идентификатора (TDOK) для каждого используемого класса объектов.

Объектные идентификаторы классов имеют уникальные имена, позволяющие однозначно различать классы используемых объектов в системе и устанавливать связи между ними.

Каждому объекту класса система присваивает уникальное значение объектного идентификатора данного класса - системный номер экземпляра (Ndok). Таким образом, системный номер (Ndok) однозначно идентифицирует объект (экземпляр данного класса) в рамках данного класса объекта (TDOK).

Основным объектом, с которым оперирует система и ее пользователи, является понятие документа (форма). Например, для формы изображенной на рис. 2 система автоматически (в момент фиксации ситуации в системных справочниках) может присвоить идентификатор TDOK="0305", который будет зафиксирован в справочнике с именем СИМ (списке типов документов предметной области).

Типом документа, или просто документом (D), называется класс объектов системы, имеющий свою информационную структуру и уникальное значение системного номера типа документа (TDOK). Другими словами у каждого типа документа D имеется свой объектный идентификатор, с уникальным именем TDOK, соответствующим данному типу документа

Обратите внимание TDOK -это уникальное символьное имя, используемое, например, в пользовательских функциях и доступно программам в виде глобальной переменной с именем TDOK.

Различные документы, относящиеся к одному типу, называются экземплярами документа. Экземпляр документа (DE) данного типа называется некоторое подмножество множества данных, хранящихся в системе, выделенное по имени и значению объектного идентификатора данного типа документа. Уникальное значение объектного идентификатора каждому экземпляру документа СИСТЕМА присваивает автоматически. Это значение позволяет системе различать экземпляры документов одного и того же типа, обеспечивая, тем самым, существование каждого экземпляра, как целостного объекта.

Уникальное значение объектного идентификатора экземпляра документа будет называться системным номером экземпляра документа (Ndok), которое ассоциируется с понятием значения собственного якоря в ILM [6]. Так, например, для экземпляра типа документа TDOK="0305" с именем "Сотрудники фирмы" будет присвоенное уникальное значение Ndok, доступное только системе и ее администратору в виде глобальной переменной с именем Ndok.

Подчеркнем, что понятие Ndok может быть эквивалентно понятию реляционного ключа только, если его значение формируется *только* системой и недоступно для модификации для пользователей и программистов системы.

Любой документ системы является аналогом (моделью) некоторого оригинала (понятия) предметной области и предназначен для описания внутреннего строения оригинала (структурная модель), его поведения (функциональная модель) и строения и поведения оригинала одновременно (структурно-функциональная модель).

Аналогами оригинала служат:

- реально существующие объекты (понятия) инфологического уровня;
- искусственно созданные объекты (понятия) инфологического уровня;
- математические или логико-математические конструкции, используемые в предметной области для описания принципов функционирования либо представления структуры какого-либо понятия (языковой системы) используемого в предметной области.

Таким образом, модель (документ) должна отражать и воспроизводить те реальные и существенные черты натурального объекта, которые являются наиболее важными для предметной области. Обычно предметная область не представляет собой четко отграниченных друг от друга объектов и их групп. Объекты и их группы "переливаются" друг в друга, причем переход из одной группы объектов в другую происходит не скачкообразно, через четко

очерченные границы, а путем плавного скольжения. Формальной основой для отображения оригинала предметной области в систему ORD является теория фреймов и теория нечетких множеств, которые наделяют структурно-функциональную модель оригинала, присущим ему свойством нерегулярности структуры.

Как и любая другая модель - понятие "документ" имеет внутреннюю структуру (структурную модель) одним из основных свойств (носителем) которой, является понятие - "СТАТУС ЧАСТИ" (для концептуальной и внешней модели и "СТАТУС ФАЙЛА" для физической модели), которое определяет правила причинно событийного появления информации в структуре документа. На данном этапе понятия "СТАТУС ФАЙЛА" и "СТАТУС ЧАСТИ" можно считать эквивалентными, но учитывать что часть это логическое понятие (часть является элементом структуры экранного образа документа и представляется как нечто логико-семантическое целое [6, пункт 2.1.1.] в документе), а файл физическое.

Само понятие файл определяется стандартно, как поименованная структура данных (стандартное определение, принятое в языках программирования). Каждому типу документа D соответствует множество $F(D)$ файлов, предназначенных для хранения информации всех экземпляров документа данного типа. Если файл F принадлежит множеству $F(D)$, то будем говорить, что F является файлом документа типа D .

По роли файла в структурно-функциональной модели документа, файлы подразделяют на основные ("O", "S", "X", "x"), внутренние ("V", "v"), внешние ("B") и классификаторы ("K", "k", "Q", "q", "H", "h"). Формальное определение этих понятий приведено в техническом руководстве, здесь же неформально можно сказать, что основной файл документа содержит однозначные характеристики присущие только данному экземпляру данного типа документа.

Внутренний файл предназначен для отображения многозначных собственных характеристик данного типа документа. Он не может использоваться в других типах документов как основной и внутренний. Но,

однако, в определенных условиях, запись внутреннего файла данного типа документа может быть 'порождена' внешним файлом другого типа документа (тогда он получает статус 'внутреннего - согласованного' "v"). Обычно такая ситуация наблюдается тогда, когда несколько объектов вступают в связь и один из объектов наследует свойства другого в результате такой связи.

Внутренний и основной файл документа предназначены, прежде всего, для фиксации свойств, присущих данному типу документа в момент его 'рождения' в информационной модели.

Особую роль в системе играют классификационные файлы, которые позволяют связывать несколько понятий в новое понятие системы. Файл F, принадлежащий F(D), называется КЛАССИФИКАТОРОМ документа типа D, если его записи могут создаваться вне связи с процессом заполнения экземпляров документа типа D. При этом в системе различают три типа классификаторов - простые ("K"), сложные ("Q") и модельные ("H") классификаторы.

Основное назначение "K" и "Q" классификаторов - это установление областей допустимых значений домена для какого-либо атрибута (свойства) документа. Например, типичным случаем простого классификатора является требование, что областью допустимых значений для свойства "пол" объекта СЛУЖАЩИЙ является область значений {мужской, женский}.

Если, раскрытие области допустимых значений какого-либо свойства требует некоторого контекста для данного значения, то для означивания таких характеристик используют понятие сложного классификатора ("Q"). Например, если в объекте необходимо зафиксировать такое свойство как значение "Минимальная ставка", без указания даты, когда данная минимальная ставка была установлена, то для установления данного факта необходимо использовать "Q" классификатор со свойствами ("Минимальная ставка", "Дата установления").

Заметим, что в отличие от простого классификатора "Q" классификатор всегда содержит не менее двух полей и обычно является документом системы, т.е. обычно представлен в пункте меню "Справочники подсистемы".

Особую роль играет в системе модельный классификатор ("Н") это файл (группа файлов), содержащий произвольное число полей, которые могут одновременно прокручиваться на экране. Модельный классификатор является самостоятельным документом в системе, что позволяет поддерживать в системе структуры объектов любой сложности. В отличие от классификаторов типа "К" и "Q", модельный классификатор является частью документа (определение понятия части документа см. в [6, пункт 2.2.1]) и используется для установления связи между понятиями (документами системы). При этом связь модельного классификатора с другим документом может "ограничиваться" специальным семантическим кодам (поддерживаемым системой), что позволяет учитывать всевозможные нюансы семантики предметной области, при вступлении в связь различных объектов предметной области и учитывать ситуации совместимости данных хранимых в документах системы.

Понятие статуса "Внешнего файла" тесно связано с процессом функционирования информационной модели в целом и правилами порождения новых экземпляров различных типов документов. Пусть новая запись в основном файле документа типа D_j автоматически создается в процессе создания экземпляра документа типа D_i . Тогда документ типа D_j называется СИСТЕМНЫМ документом, порождаемым документом типа D_i . Тогда, ниже приведенное определение поясняет понятие "Внешнего файла". Файл F , принадлежащий $F(D_i)$ и являющийся основным для некоторого системного документа D_j , порождаемого документом D_i , называется ВНЕШНИМ.

Из определения следует, что внешний файл может иметь только одну запись, относящуюся к экземпляру документа, и содержит данные из документа другого типа. Наличие внешнего файла позволяет автоматически создавать документ D_j в процессе заполнения документа D_i .

Связь между различными типами статусов файлов (определяет правила формирования документа как объекта системы) осуществляется с помощью специальных средств системы на основе понятия ЯКОРЯ, которое имеет множество типов (например, структурные, функциональные, валентные, и т.д.). Структурные якоря бывают двух типов собственные якоря и внесенные якоря.

Каждый файл F, используемый в системе, имеет в своей структуре поле с уникальным для системы именем A (первое по порядку в структуре), которое выступает в роли объектного идентификатора для записей этого и только этого файла F. Такое поле называется СОБСТВЕННЫМ ЯКОРЕМ (СЯ) файла F.

Поле с тем же именем A, имеющееся в другом файле G, называется в системе ВНЕСЕННЫМ ЯКОРЕМ (ВЯ) файла G.

Приведенные понятия используются для отображения инфологических объектов предметной области в моделях системы ORD, и не следует их путать с понятиями первичного и внешнего ключа традиционной реляционной модели.

Описание моделей ORD хранятся в системных таблицах с именами SIIM, SIUEK, SIAGF, SIAOF, описание которых (как и всех других файлов), хранится в справочнике системы с именем SIORD и их содержимое всегда Вам доступно, что делает процесс проектирования информационной модели открытым, объяснимым и контролируемым не только Вами, но и самой системой.

Содержимое совокупности указанных файлов позволяют системе контролировать структурно-функциональную как отдельного документа системы, так и структурно-функциональную модель проектируемой инфологической модели в целом.

SIIM используется для описания доступных данной подсистеме документов системы и общих правил их функционирования (прав доступа, привилегии при совместной обработке, обработка документа в географически разнесенных узлах системы и т.д.).

SIORD используется для описания состава и структуры всех файлов (в том числе и индексов) доступных информационной подсистеме, то, что принято называть описанием внутренней модели.

SIUEK описывает внешние модели (экранные образы) документов, указанных в таблице SIIM. Здесь же указываются правила и ограничения присущие каждому элементу внешней модели (в том числе хранимые процедуры и триггеры).

SIAGF (логика документа) предназначен для описания структурно-функциональной модели документа (формы) и правил построения и заполнения частей документов, которое находит свое отражение в понятии маршрута документа (см. [6], параграф 2.1.3. Маршрут формы (документа)).

SIAOF (список файлов документа) используется системой для контроля описания структуры документа и создания архивных, резервных и журнальных файлов отката при аппаратных и программных сбоях в системе.

В системе разрешены модели документов, в которых реализуются следующие виды связей между частями (рис. 3 а, б, в)

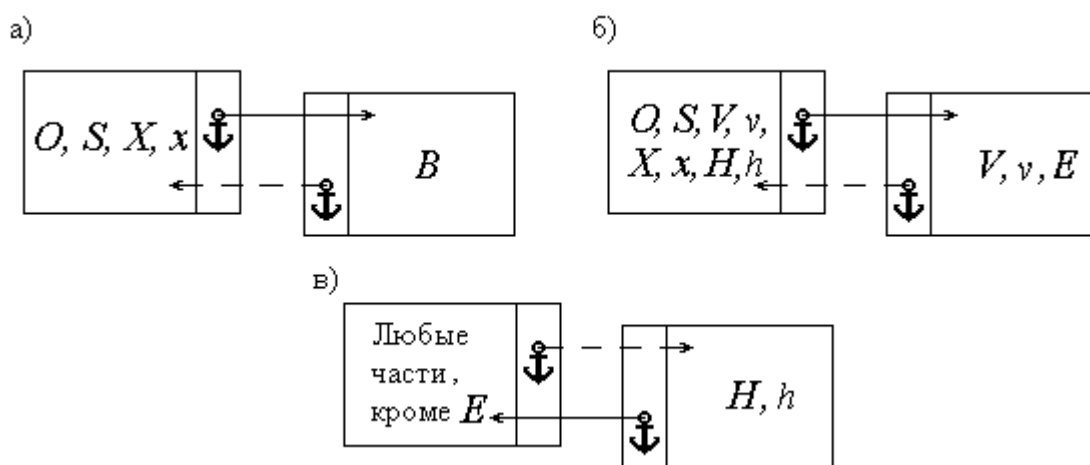


Рис. 3. Допустимые способы связей частей в форме

Как видно из рисунка связи между частями формы осуществляются с помощью якорей (значения которых формируются системой автоматически). Среди этих связей имеются основные (обязательные), необходимые для работы системы, и вспомогательные, устанавливаемые разработчиком документа для своих целей.

Причем, сплошной линией обозначены обязательные связи, а пунктирной – вспомогательные связи. Вертикальной чертой отделен собственный якорь части, изображенный соответствующим значком.

Все обязательные связи система устанавливает автоматически, вспомогательные связи указываются разработчиком при формировании ПЛМ и автоматически отслеживаются системой.

Согласно рис. 3 (а) внешняя часть может подключаться к основной части любого типа (O, S, X, x). При этом СЯ (собственный якорь) основной части, обязательно вносится во внешнюю часть. Перенос СЯ внешней части в часть, к которой она подключена, не обязателен.

В соответствии с рис. 3 (б) внутренняя часть может подключаться к основной, внутренней и части типа *H*. При этом обязательно, чтобы во внутреннюю часть вносился СЯ той части, к которой она подключена. Перенос СЯ внутренней части в часть, к которой она подключена, не обязателен.

Рис. 3. (в) показывает, что классификатор может подключаться к части любого типа с обязательным переносом ее СЯ в часть, к которой она подключена. Занесение СЯ части, к которой подключен классификатор, в сам классификатор не обязателен.

Приведенная кодировка используется для описания информационно-логической структуры определенного класса документа и позволяет отразить в системе структурные и функциональные связи, присутствующие в данной предметной области. Связи типа “К” и “Q” на рисунке не представлены, т.к. они являются элементами частей (кроме текстовых частей типа ‘E’)

Если приведенная структура ИЛМ (для рисунка 1) Вам будет понятна (рис. 4), то следует переходить к разработке моделей ИЛМ, для проектируемой подсистемы. На рис. 1 в части 3 “Сведения о работнике” может находиться информация о документах удостоверяющих личность либо сведения о гражданстве и т.д. На рисунке 4 не указаны классификаторы типа “К” и “Q”, которые определены на атрибутах *должность, цех, группа учета, форма оплаты, пол*, чтобы подчеркнуть независимость этих объектов от объекта СОТРУДНИКИ ФИРМЫ.



Рис. 4. Граф ИЛМ для формы представленной на рис.1

4.4. Разработка структуры документов предметной области

Когда выявлены все подзадачи, и определено меню подсистемы становится ясным, какого типа операции с данными следует выполнять для

решения функциональных задач. Поэтому можно приступить к разработке структур документов необходимых для реализации Вашего проекта. Все вопросы, связанные с описанием, созданием, модификацией физических файлов, где будет храниться информация о модели предметной области, будут автоматически отслеживаться системой.

Выполнение данного этапа подразумевает, что Вы ознакомились с методологией проектирования распределенных информационных систем в среде ORD. Необходимая документация находится на Вашей машине, если система инсталлирована. В папке SUBS (сама система), находится всевозможные Help, где представлена совокупность математических и физических моделей, позволяющих Вам построить (отобразить) инфологическую модель Вашей предметной области в концептуальную модель системы ORD.

Учитывая, что пример курсового проекта для фрагмента предметной области "Учет материалов" у Вас имеется, ниже рассмотрены только основные моменты проектирования некоторых документов подсистемы, принимая во внимание, что другие документы строятся аналогично.

Пусть в качестве формального имени Вашей подсистемы будет использован символ "L" и необходимо отобразить в этой подсистеме два документа (две модели) для существующих в предметной области понятий СОТРУДНИКИ ФИРМЫ и НАКЛАДНАЯ ПРИХОДА.

Чтобы создать новую информационную систему достаточно запустить инструментальную оболочку Db.exe (командный файл D.bat в любой из папок любой информационной подсистемы) и нажать кнопку F2 – Создать (рис. 5).

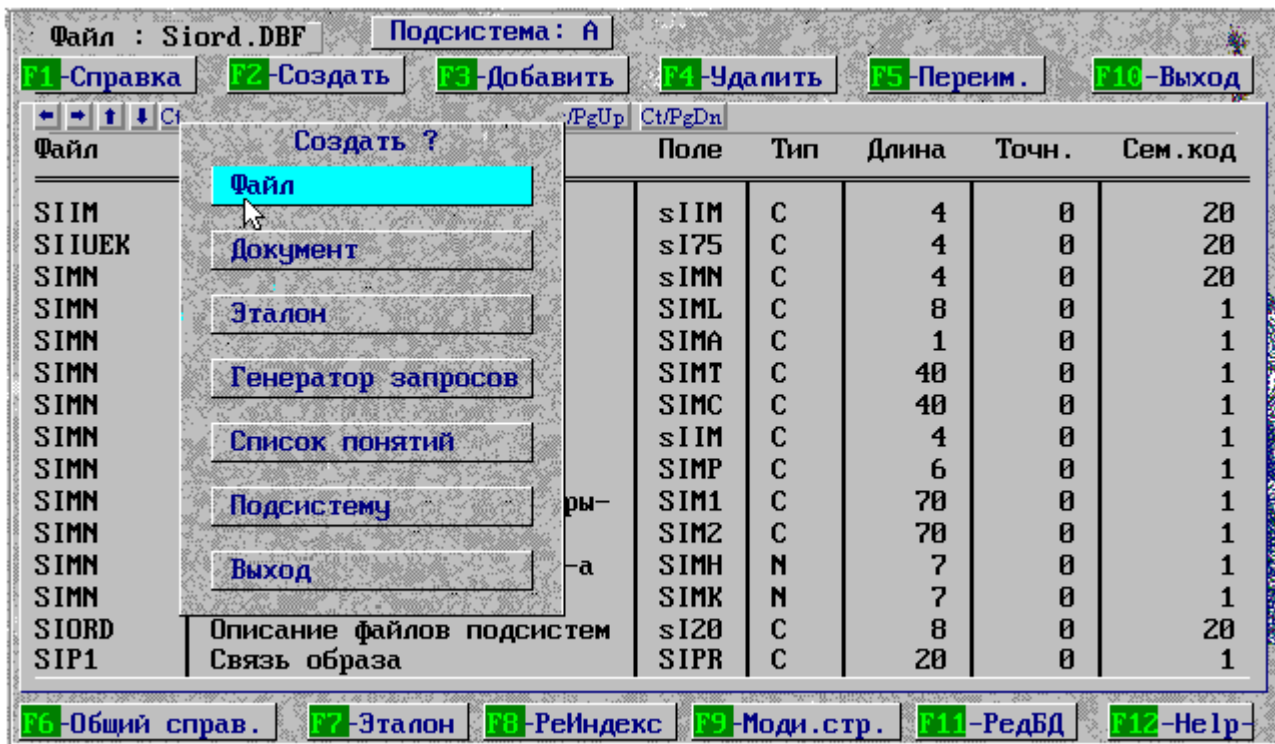


Рис. 5. Окно инструментальной оболочки Db.exe

Установив курсор на пункт меню “Подсистему”, и выбрав его, Вы прикажете системе создать новую информационную подсистему в папке SUBL. В этой папке будут созданы соответствующие поддиректории, структура которых приведена на рисунке 37.

В таблице SIIM (вновь созданной подсистемы рис.6) автоматически будут прописаны встроенные документы-утилиты для реализации пункта “Сервис” (рис. 1) Вашей подсистемы.

В папке PRG_WIN будет находиться исполняемый модуль T.exe, и исходный текст головной программы T.prg, и командные файлы Cm.bat (компиляция) и 3.bat (компоновка), которые могут Вам потребоваться, если Вы будете расширять возможности системы.

После создания своей подсистемы следует перейти в папку ... \SUBL \PRG_WIN \ и запустить в ней инструментальную оболочку (d.bat) для отображения ИЛМ на концептуальную внешнюю и внутреннюю модели системы.

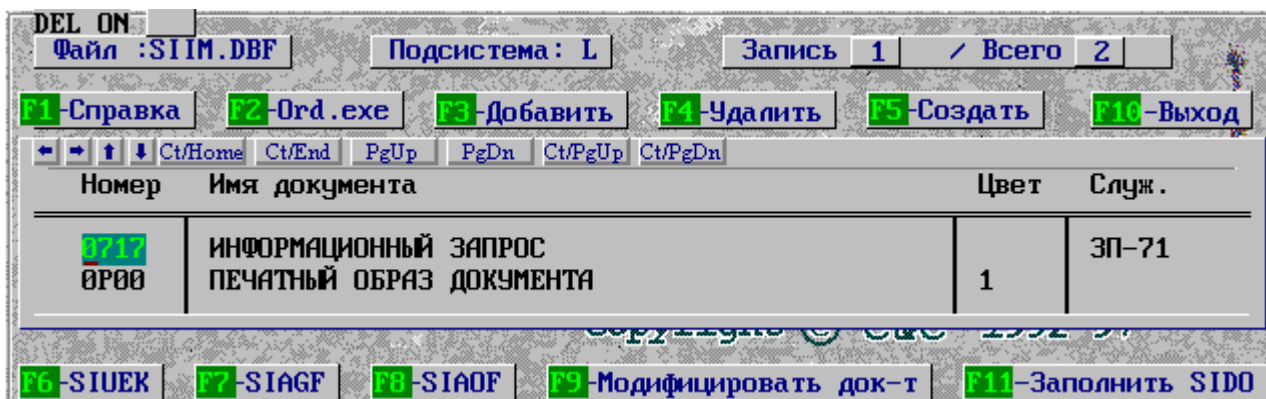


Рис. 6. Таблица SIIM после создания подсистемы “L”

Все указанные преобразования система выполнит автоматически, если разработанная ILM не будет противоречить рис. 3.

Пусть в Вашей подсистеме существует понятие инфологического уровня СОТРУДНИКИ ФИРМЫ, и Вы приняли решение использовать его для своих целей. Самое простое решение – это скопировать (см. лаб. работу) его из той подсистемы, где эта форма формировалась впервые, и модифицировать его под свои условия. Однако, учитывая учебный характер излагаемого материала, рассмотрим более подробно процесс построения модели ILM документа TDOK="0305" и связь этой модели с другими моделями системы.

Пусть заказчика интересуют следующие атрибуты, относящиеся по его представлению к понятию СЛУЖАЩИЙ: *Фамилия, Имя, Отчество, Страховой номер, Дата рождения, Дата приема, Пол, Цех, Должность, Оклад*. Заметим, что если данное отношение будет представлено в базе данных в виде одной таблицы, то в процессе сопровождения такой базы могут возникнуть серьезные неприятности, рассмотренные в [5, 6].

Ваша задача разработать ILM в соответствии с требованиями, которые предъявляются к подобным моделям в системах объектно-реляционного типа, т.е. модель должна быть структурирована таким образом, чтобы она не противоречила концептам этой модели и в максимально возможной степени позволяла обеспечить принцип независимости данных.

Первое, что требует ILM это разделение всех атрибутов, выделенных для описания понятия предметной области, на однозначные и многозначные. При

этом следует системно использовать такие понятия как: **объект, свойство объекта, связь, свойство связей, класс объектов (связей)** и понятие семантического кода рассмотренные в [6]. Только учитывая все эти требования как систему, можно достаточно адекватно (с учетом прогнозирования развития) отобразить реальную ситуацию в ИМ. Если при этом концептуальные ограничения, указанные на рис. 3 будут выполняться, следует принять решение о реализации ИМ в документ системы.

Анализ понятия СЛУЖАЩИЙ позволяет с большой уверенностью отнести такие атрибуты как *Фамилия, Имя, Отчество, Страховой номер, Дата рождения, Дата приема, Пол* к однозначным собственным свойствам (семантический код для таких атрибутов равен единице) объекта СЛУЖАЩИЙ, т.к значения этих атрибутов устойчиво во времени и практически не модифицируется.

Рассматривая атрибуты: *Цех, Должность, Оклад* можно также отнести их к однозначным характеристикам, т.к. в данный момент времени конкретный служащий работает именно в данной фирме на определенной должности и имеет один оклад. Если же предположить, что служащий может по совместительству работать в другом отделе фирмы, или требуется хранить историю изменения окладов служащего и переход его с одной должности на другую - требование однозначности будет нарушено.

Если приведенные выше аргументы являются недостаточными для выделения указанных атрибутов в отдельную часть документа, и утверждается, например, что в фирме всего один отдел и заказчика не интересует история изменения окладов служащих, следует еще раз обратиться к рабочим определениям: **{объект, свойство объекта, связь, свойство связей, класс объектов (связей)}** [6] для окончательного принятия решения о включении такого документа в систему.

Если не будет найдено ни каких противоречий и при анализе этих атрибутов по их принадлежности к определенной категории, то следует обратиться к понятию семантической близости атрибутов [6] этой группы к

элементам из: {*Фамилия, Имя, Отчество, Страховой номер, Дата рождения, Дата приема, Пол*}.

Понятие семантической близости можно трактовать следующим образом: проекция по любому сочетанию атрибутов, входящих в часть документа должна нести объяснимую нагрузку и обладать свойством симметричности.

Например, трудно объяснить проекцию на атрибуты [Цех, Пол], хотя оба атрибута относятся к понятию СОТРУДНИК. Но сам по себе ЦЕХ (самостоятельный объект анализируемой предметной области) не может иметь в качестве собственного свойства атрибут пол (конечно, если под полом не понимают – деревянный, бетонный).

Если все эти аргументы (они все не формальны) не заставляют Вас выделить атрибуты в отдельную часть, необходимо еще раз прочитать определения фрейма [6] и принять окончательное решение.

Таким образом, данная ситуация может быть представлена следующими видами ИЛМ (рис. 7):

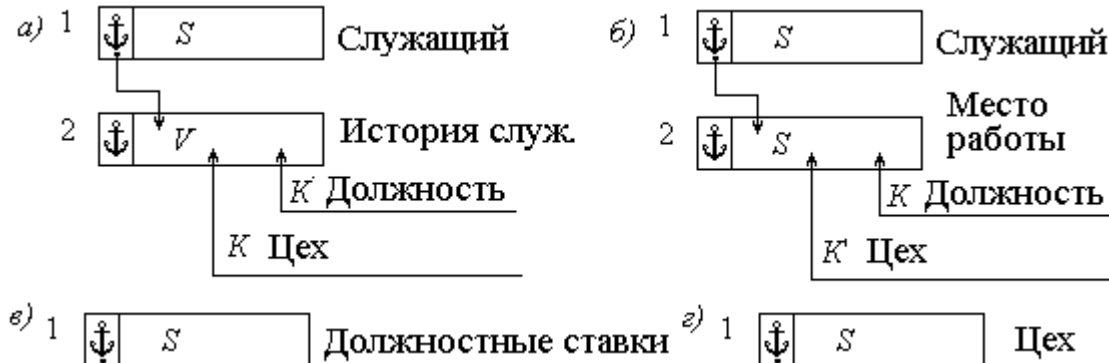


Рис. 7. Виды ИЛМ для ситуации "Служащий фирмы"

С точки зрения методики, принятой в системе ORD, модель ИЛМ, приведенная на рис. 7 (а) наиболее полно отвечает концептам системы. При этом объекты, указанные на рисунках 7 (в) и 7 (г), должны быть представлены как самостоятельные объекты (документы) инфологического уровня. Следует учесть, что если даже в фирме всего один отдел и заказчика не интересует смена должностей и окладов сотрудника, необходимо остановиться на модели, указанной на рис. 7 (а). Для указания же подобного ограничения в предметной

области следует воспользоваться правилом, указывающим, что часть 2 может иметь только один слой.

Чтобы отобразить ILM (рис. 7 а) в системе необходимо добавить в таблицу SIIM (рис. 6) запись следующими характеристиками: Номер=0305 и Имя документа=Сотрудники фирмы. Далее, нажав клавишу **F5** – **Создать**, войти в визуальный построитель модели и отобразить модель, например, в виде указанном на рис. 2. Например, на рис. 8 представлен один из таких вариантов.

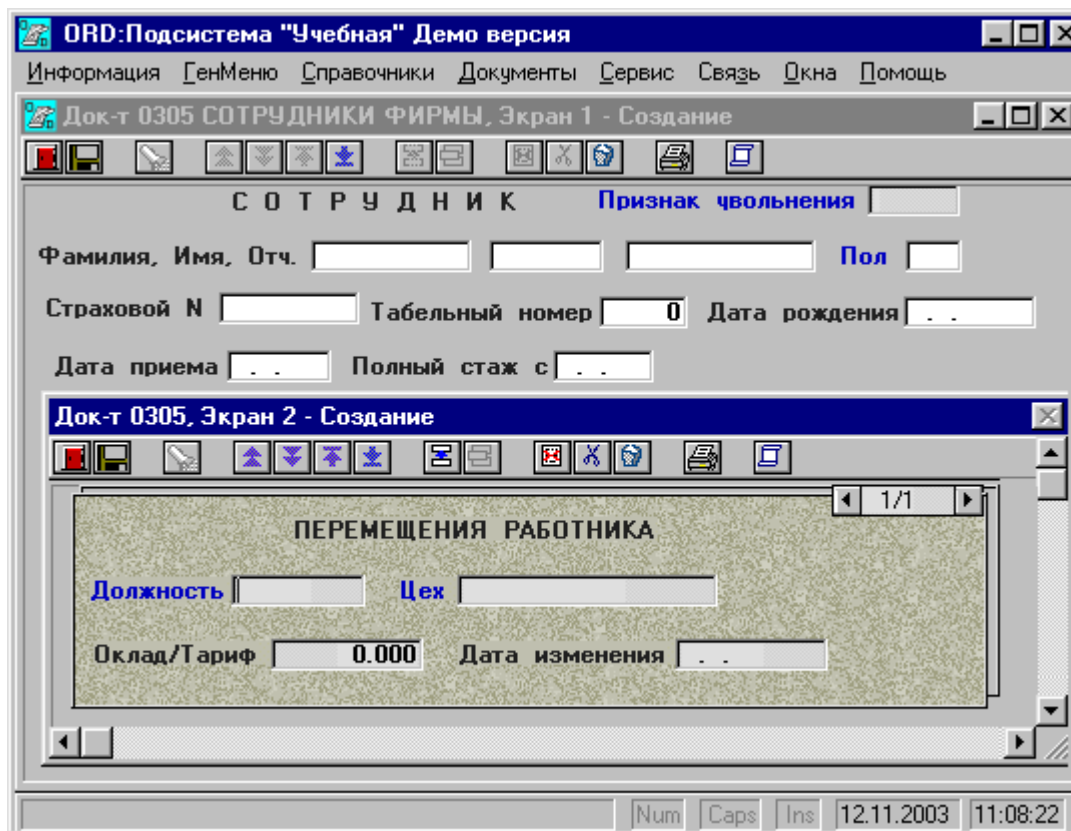


Рис. 8. Отображение ILM (рис. 7 а) на экране дисплея

Естественно, что модель представленная на рисунке 8 будет автоматически отображена в системных таблицах (SIUEK, SIAGF и SIAOF), где будут зафиксированы концептуальная, внешняя и внутренняя модель (каждому GET полю в таблице (файле) будет соответствовать столбец с уникальным именем, сформированным системой) созданного документа. В системном меню (таблица SIMN) для созданной формы будет указана соответствующая строка, которая будет представлена пунктом меню, как показано на рисунке 1.

Для заказчика содержимое и структура этих моделей безразлична, но для разработчика может оказаться полезным знать как ‘устроены’ ‘эти модели, так

как разработчик с помощью этих моделей получает доступ к хранимой информации.

В этих моделях описываются не только обычные характеристики GET полей (тип, длина, точность), но и фиксируются правила поддержки целостности данных.

Поддержка целостности данных включает в средства задания ограничений, которые вводятся с целью защиты базы данных от нарушения согласованности сохраняемых в ней данных. К таким типам ограничений целостности обычно относят:

- обязательные данные;
- ограничения для доменов атрибутов;
- целостность сущностей;
- ссылочная целостность;
- бизнес правила (требования данного предприятия).

В системе ORD все указанные типы ограничений реализуются в соответствии со стандартом ISO на язык SQL, однако их поддержка и правила описания отличаются от указанного стандарта, т.к. в качестве базового языка поиска данных используется визуальный язык QBE (запрос по образцу) [8] и некоторые из указанных типов ограничений расширены, чтобы обеспечить максимально понятный и удобный интерфейс для разработчика информационных систем.

Описание и назначение приведенных видов ограничений приведено практически во всех руководствах связанных с СУБД и некоторые из них Вам уже хорошо известны. Так, например, ограничения для доменов атрибутов есть не что иное, как некоторый набор допустимых значений данного атрибута в данной части документа и реализуется в системе с помощью механизма простых (“К”) классификаторов.

Для примера приведенного на рисунке 8 в качестве таких атрибутов могут быть указаны: *Пол* и *Признак увольнения*. При этом областью значений поля “Пол” могут быть использованы два допустимых значения: {Мужской,

Женский}, а для поля “Признак увольнения” элементы из множества: {Да, Нет} или {пробел, “1”}. При этом символ “1” будет обозначать, что сотрудник уволен, а его отсутствие, что сотрудник продолжает числиться на фирме.

В то же время, если на фирме зафиксирован сотрудник, то он должен относиться к категории работающих или уволенных, это приводит к понятию обязательности данных. Для задания ограничений подобного типа обычно используется спецификатор NOT NULL. Например, если поле (столбец) Должность в части 2 (рис. 8) обязательно должно иметь значение, то необходимо использовать ограничение на обязательность данных.

Обычно приведенное ограничение прописываются в операторах CREATE языка описания данных [4] и имеют вид: Должность CHAR (15) NOT NULL

Учитывая, что в системе разрешается не только набор простых плоских таблиц, описание ее моделей требует пояснения.

Нетрудно заметить, что расположение атрибутов и полей формы на экране может иметь произвольное значение (например, каждый атрибут находится в определенной строке формы и начинается с определенной позиции (столбца экрана)). Кроме того, каждый элемент формы может определять не только его назначение в форме, но и определять значение другого атрибута или целой части в целом или права доступа. Все эти моменты и отмечаются автоматически в моделях системы в таблицах, имена которых начинаются с символов “SI”.

На рис. 9 представлен фрагмент внешней модели (таблица SIUEK) для формы, представленной на рис. 8. Первая строка описания любой формы в таблице SIUEK начинается со служебной строки, с символом управления “M” в которой в столбце ‘Файл’ система прописывает имя физического файла (в примере LA01), в котором будет храниться информация документа. Первое поле (тип CHAR) указанного файла обычно используется для хранения собственного якоря экземпляра данного типа документа (в примере TDOK=0305) и совпадает с именем файла.

Сист.	№ образа	Имя атрибута на экране	Упр.	Y	X	Файл	Поле	Y	X	Появ.	Тип	Длина	Точн.
8700	UL305001	Ся Служащего	M	0	0	LA01	LA01	0	0	0	C	4	0
8702	UL305001	СОТРУДНИК	Z	1	15			0	0	0			0
8704	UL305001	Признак увольнения	K	1	45	LA01	LA0L	1	54	0	C	1	0

Подсистема L Файл SIUEK.DBF Запись 1462 / Всего 1733

Рис. 9. Фрагмент внешней модели формы TDOК=0305

Вторая строка с символом управления “Z” (имя столбца Упр.) указывает, что на экране в первой строке окна формы с позиции 15 (столбца окна X) будет выведен заголовок СОТРУДНИК (см. рис. 8).

Третья строка указывает, что значение атрибута *Признак увольнения* должно выбираться из списка (классификатора типа “k”) допустимых для этого атрибута значений (без права модификации этих значений). При этом атрибут должен выводиться в 1-ой строке с 45-ой позиции, а его значения с 54-ой позиции.

Примечание. Символ, указанный в столбце Упр. таблицы SIUEK (рис. 9) относится к понятию “Статуса поля” (см. приложение).

Аналогичным образом системой прописываются и другие строки указанные на рисунке 8. Естественно в зависимости от роли строки, статуса поля и семантического кода определяется и типы ограничений целостности для каждого поля формы.

Если, находясь в окне внешней модели (рис. 9) нажать клавишу F7-SIAGF, то система раскроет описание логической модели (рис. 10) формы представленной на рисунке 8 (эту модель иногда называют структурно-логической). В этой таблице, каждой части ИЛМ соответствует одна строка, в которой указываются границы ‘Начала’ и ‘Конца’ описания части в таблице SIUEK (рис. 9), логика перехода из части в часть (‘Переход’, ‘Отказ’), структура графа ИЛМ, статус и название части и экран (страница) на котором располагается часть. Столбцы Спец.1 и Спец.2 используются для сложных

блоковых (рекурсивных) моделей ИЛМ и указания специфических ограничений целостности либо документа в целом, либо какой-то ее части.

N	Часть	Начало	Конец	Пережд	Отказ	Вершина	Статус	Экран	Спец.1	Спец.2	Название части
0305	1	8700	8720	2	99	0	S	1			Сотрудник
0305	2	8721	8744	98	98	1	V	1			Перемещения

Рис. 10. Отображение ИЛМ в таблице SIAGF

Как указывалось, в момент преобразования ИЛМ система создает хранимые таблицы (латинские имена этих таблиц стандартизованы), перечень которых фиксируется в системной таблице SIAOF (рис.11).

Тип документа	Файл документа	Статус файла	Номер образа	Номер части
SIIM	SI20	SI59	SI52	SI60
0305	LA01	O	UL305001	1
0305	LA03	V	UL305002	2
0305	LAOL	k	UL305001	1
0305	LTAR	K	UL305001	1
0305	LGTA	k	UL305002	2
0305	LT05	Q	UL305002	2

1	LA01	Служащий	к Пол LAOL
	LTAR	КПризнак увольнения	
2	LA03	История служ.	к Должность LGTA
	Q Цех LT05		

Рис. 11. Отображение ИЛМ (б) в таблице SIAOF (а)

Отметим, что на рисунке 11 показана Q-связь, которая обычно используется как контекстная связка [6]. В отличие от “К” классификатора, Q-классификатор должен быть оформлен как документ системы.

Примечание. Новые значения в К и Н-классификаторы можно вводить как заранее, так и в процессе заполнения документа, для которого они понадобились, все зависит от правил указанных для этих объектов. Например, символ “к” (в описании в Siuek) указывает, что домен значений для данного атрибута сформирован заранее, а символ “К” допускает его расширение в данном типе документа.

На этом процесс “встраивания” ИЛМ в структуру базы данных можно считать завершенным, а требования ограничений целостности автоматически будут отслеживаться системой.

Так, например, требование целостности сущностей, трактуемое в реляционных системах как отсутствие пустых значений в первичном (PRIMARY) ключе таблицы не может быть нарушено по определению собственного якоря, который присутствует в каждом экземпляре объекта (формы).

Выполнение требований ссылочной целостности, означающее, что если поле внешнего ключа содержит некоторое значение, то оно обязательно должно ссылаться на существующую корректную строку в родительской таблице (см. [7], стр. 489), является внутренним, структурным свойством формы и по определению не может быть нарушено.

Реализация требований специфичных для конкретного предприятия рассмотрим в параграфе, раскрывающем некоторые вопросы, связанные с программированием в системе ORD.

Следует отметить, что, в простейшем случае, построенная и отображенная модель в среду СУБД совпадает с набором нормализованных таблиц классической реляционной модели и для рассмотренного примера (рис. 8) может состоять из следующего набора таблиц (рис. 12), в которые для иллюстрации занесены некоторые данные для двух сотрудников.

Заметим, что каждая таблица на рисунке 12 имеет якорный столбец, который всегда будет физически первым в хранимой таблице. Расположение других столбцов произвольно. Для удобства и наглядности столбцы в таблицах показаны в том порядке, в котором они встречаются на рисунке 8. Так, например, для GET поля *Признак увольнения* в таблице соответствует столбец с именем "LTAR" (рис. 12 а). GET полю с атрибутом *Фамилия* соответствует столбец "LA0F". Если столбец таблицы классификационный, то за ним всегда следует служебное поле наследования [3], начинающееся с символа "Z". Если таблица является самостоятельным объектом в системе, то в ней присутствует столбец маршрута [6]. Поля таблицы, начинающиеся с "Z" и маршрутные поля система генерирует автоматически.

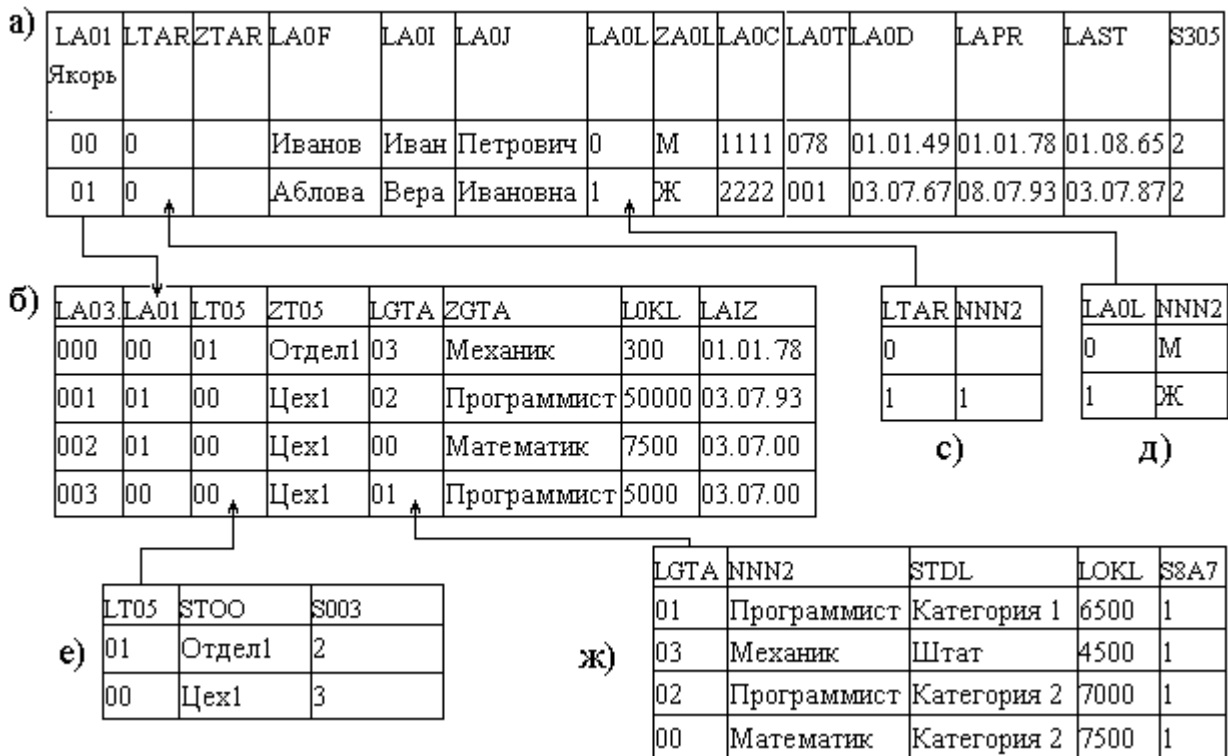


Рис. 12. Состояние отношений схемы модели

Как указывалось выше, одним из документов, используемых в фирме, является “Накладная прихода” (в системе его TDOК=0008), обобщенный вариант которого приведен на рисунке 13. Документ предназначен для регистрации поступления товаров на склад, как от внешних поставщиков, так и при внутренних передачах.

По своей структуре “Накладная прихода” похожа на типовые документы и отчеты, содержащие шапку, табличную часть и часть итогов. Однако табличная часть накладной (часть может иметь табличное представление (Shift\F1), если на экранном образе эта часть охвачена двойной рамкой) имеет более сложный вид и содержит вторую таблицу, вложенную в первую.

Это вызвано тем, что в основной табличной части накладной перечисляются полученные по ней материалы, а во вложенной табличной части, которая может содержать несколько строк для каждой из строк основной таблицы, записывается в какие именно единицы хранения на складе, приходится полученный товар. В частности, в нашей фирме “МеДина Лтд” каждая единица хранения получает некоторый номер, называемый ярлыком.

ORD:Подсистема "Учет материалов" Демо версия

Сводное Меню Справочники Накладные Склад Отчеты Сервис Связь Окна Помощь

Док-т 0008 Накладная прихода - Просмотр

Накладная N 1 Дата 21.05.2003 поставки Вид прихода Внешний

Поставщик ОАО "ТАЛВИС" Проведен - 1 1

Товары накладной

Группа	Ткань	Кол-во/длина	Товар	Арго 004-505
Тип единицы товара	Рулон	16.00 м.	На склад	Для тканей
Кол-во(факт)	4.00 единиц	Цена закупки	14.00 руб. за	м.
Всего(факт)	64.00 м.	Стоимость(факт)	896.00 руб.	
Всего(Док)	64.00 м	Стоимость(Док.)	896.00 руб.	

Маркировка товара

Ярлык 1-1Аг Фактическое кол-во/длина 16.00 м.

Итого факт 896.00 руб. Итого по док. 896.00 руб.

Рис. 13. Экранный образ документа TDOK=0008

В системе ORD такую структуру обозначают, как структуру, состоящую из четырех частей: основной части (шапки, головной), или части типа (статуса) "S", основной табличной части, или части типа V, вложенной табличной части, также имеющей тип V, и части-продолжения основной части, очевидно, того же самого типа "S".

Графически такую структуру можно представить так, как показано на рисунке [14]. На рисунке прямоугольниками изображены четыре части документа (вершины графа). В каждом прямоугольнике обозначен тип части. Кроме того, в левой стороне прямоугольника частей 1 - 3 помещено изображение якоря. Оно символически обозначает уникальный идентификатор каждой записи (слоя) части, автоматически формируемый системой. Как отмечалось (рис. 12), этот идентификатор физически хранится в первом поле каждой записи файла, и называется собственным якорем и используется для связи частей каждого экземпляра документа в единое целое. Если собственный якорь файла части используется в другом файле, то он получает статус

внесенного якоря (отметим, что может использоваться рекурсивное использование якоря, но подобные конструкции здесь не рассматриваются).

Система обеспечивает передачу внесенных якорей в соответствии с динамической структурой документа [6], что и позволяет трактовать подобные структуры как фреймы.



Рис. 14. Модель ILM документа TDOK=0008

В части 4 такого якоря нет, так как эта часть является продолжением первой части, и все ее данные, относящиеся к конкретному экземпляру документа, хранятся в той же записи того же файла, что и данные первой части.

В первой (основной части) указываются номер и дата накладной, а также поставщик и вид прихода. В этой же части указывается признак проводки накладной, заполняемый автоматически при учете поступления по накладной.

Во второй (многослойной - табличной) части перечисляются характеристики, количество, цена и стоимость товара. При этом поля общего количества товара и его стоимости заполняются автоматически (процедурами, встроенными в документ).

В третьей, многослойной части, вложенной во вторую часть, для каждого поступившего товара указываются единицы хранения (ярлыки, фактически, инвентарные номера), которыми товар будет обозначаться во время его пребывания на складе.

В четвертой части, являющейся продолжением первой части, записаны итоговые стоимостные данные по накладной.

Пусть система сформировала следующие уникальные имена для файлов документа: данные первой и четвертой частей хранятся в файле AA07.DBF, данные второй части – в файле AA0E.DBF, а третьей – в файле SAYY.DBF.

Примечание. В системе используется структурированный формат хранения данных известный в литературе по расширению - .DBF. В дальнейшем расширение в имени файла будет опускаться.

В общем случае, число файлов для хранения информации части не ограничено, однако для простоты будем считать, что каждая часть определена на одном файле. Отметим, что в части 4 нет якоря, так как эта часть является продолжением первой части, и все ее данные, относящиеся к конкретному экземпляру документа, хранятся в той же записи того же файла, что и данные первой части.

Рассмотрим, чем же руководствовался проектировщик модели базы данных, предложивший структуру модели, представленную на рисунке 14. Естественно был проведен тщательный анализ ситуации и выделены объекты и связи между ними на основании методики рассмотренной в [6].

На основании полученных результатов, в основной части (таблица AA07), содержится ровно по одной записи для каждого экземпляра накладной. В этой записи, соответствующей шапке накладной, записываются: *номер накладной, дата поступления товара, признак оприходования накладной, поставщик товара, вид прихода* (рис. 13), а также итоговые данные по накладной из четвертой части. В данном случае, часть 4 (продолжение) выделена из-за удобства восприятия накладной, т.к. ее данные должны сформироваться из значений атрибутов, находящихся выше в экранном образе.

В таблице AA0E основной табличной части может содержаться несколько записей для каждой накладной. Каждая запись соответствует одному виду поступившего по накладной товара. В ней записывается вид товара, который характеризуется рядом признаков, количество поступившего товара и

его стоимость. Причем, в отличие от твердой копии накладной, в ее электронный образ записывается и фактически принятое количество.

Наконец, в таблице SAYU (вложенной табличной части) может содержаться несколько записей для каждого вида товара, т.е., для каждой записи основной табличной части. В записи этой таблицы фиксируется распределение поступившего товара на единицы хранения его на складе: ярлык (номер единицы хранения) и соответствующее ему количество товара.

Такую структуру уже достаточно сложно воспроизвести в стандартных средствах проектирования приложений. Хотя, конечно, можно обойти эти проблемы, например, создав два типа независимых таблиц: “Приход” и “Распределение”, связав их друг с другом, что, однако, нарушит логику отображения рассматриваемой ситуации.

Далее, попробуйте представить себе работу персонала фирмы “MeDina Лтд”, которая будет пользоваться данным приложением. По сути рассматриваемых бизнес процессов – это, прежде всего, менеджеры по закупкам и продажам, а также, почти что главная фигура: кладовщик (иногда говорят, менеджер склада).

Ввод “Накладной прихода” в базу данных при поступлении товара производится или непосредственно кладовщиком склада, принимающим товар, или совместно менеджером по закупкам и кладовщиком. Участие последнего обязательно, так как только кладовщик может заполнить часть накладной, описывающую распределение товара по единицам хранения на складе.

Собственно ввод данных осуществляется при вызове накладной из меню в режиме создания документа (рис. 1).

В приложениях ORD используются три режима работы с документом: Создание, Просмотр и Редактирование. Их назначение следует из названий, а особенности функционирования изложены в описании системы ORD .

В процессе ввода поля номера, даты оприходования, количества, цены и аналогичные им заполняются непосредственно вводом соответствующих значений. Атрибуты (названия) этих полей записаны черным цветом.

Поля *Вид прихода*, *Поставщик*, *Группа*, *Товар*, *Тип единицы* товара и *Склад* выбираются из списков имеющихся значений (классификаторов) Например, классификатор *Вид прихода* может состоять из следующего набора значений: {Внешний, Внутренний, Ввод остатков, Возврат}. Атрибуты этих полей на экране выделены ярко-синим цветом.

Все использованные в 'Накладной прихода' классификаторы, за исключением классификатора 'Тип единицы товара', являются классификаторами типа Q. Им соответствуют отдельные типы документов, вызываемые из пункта меню Справочники. Ввод новых значений и корректировка ранее введенных значений в этих классификаторах производится при вызове соответствующего документа. Классификатор 'Тип единицы товара' имеет тип "К", и ввод в него новых значений и корректировка ранее введенных производятся непосредственно в самом классификаторе при вызове его из документа 'Группы товаров'.

Наконец, поля типа: *Всего*, *Стоимость*, *Итого* атрибуты, которых записаны темно-синим цветом, являются полями **процедурного** типа. Их значения не вводятся, а вычисляются и заносятся автоматически процедурами (системными или пользовательскими), подключенными к документу.

В нашем АРМ заполнение накладной, свидетельствующее о поступлении товара на склад, автоматически вызывает выполнение еще одной процедуры: создание или корректировку ТОВАРНОЙ КАРТОЧКИ (TDOK=000S). Этот документ, сохранившийся в складском учете с давних времен, настолько привычен для персонала, что заставить отказаться от него практически невозможно.

Но нет худа без добра. ТОВАРНАЯ КАРТОЧКА очень показательна для демонстрации возможностей имеющегося в ORD модельного классификатора. Так как приход товара изначально отражается в НАКЛАДНОЙ ПРИХОДА, то в ТОВАРНОЙ КАРТОЧКЕ фигурируют не только данные о поступившем количестве товара, но и атрибуты самой накладной. Естественно, они либо должны дублироваться (что потенциально содержит возможность внесения

противоречий в базу данных) либо отображаться в карточке непосредственно из накладной (что дублирование и возможность противоречия исключают). Модельный классификатор в ORD и позволяет реализовать такое отображение, высвечивая непосредственно атрибуты накладной в документе ТОВАРНАЯ КАРТОЧКА ТКАНИ. Образ и модель этого типа документов приведена ниже на рисунке 15.

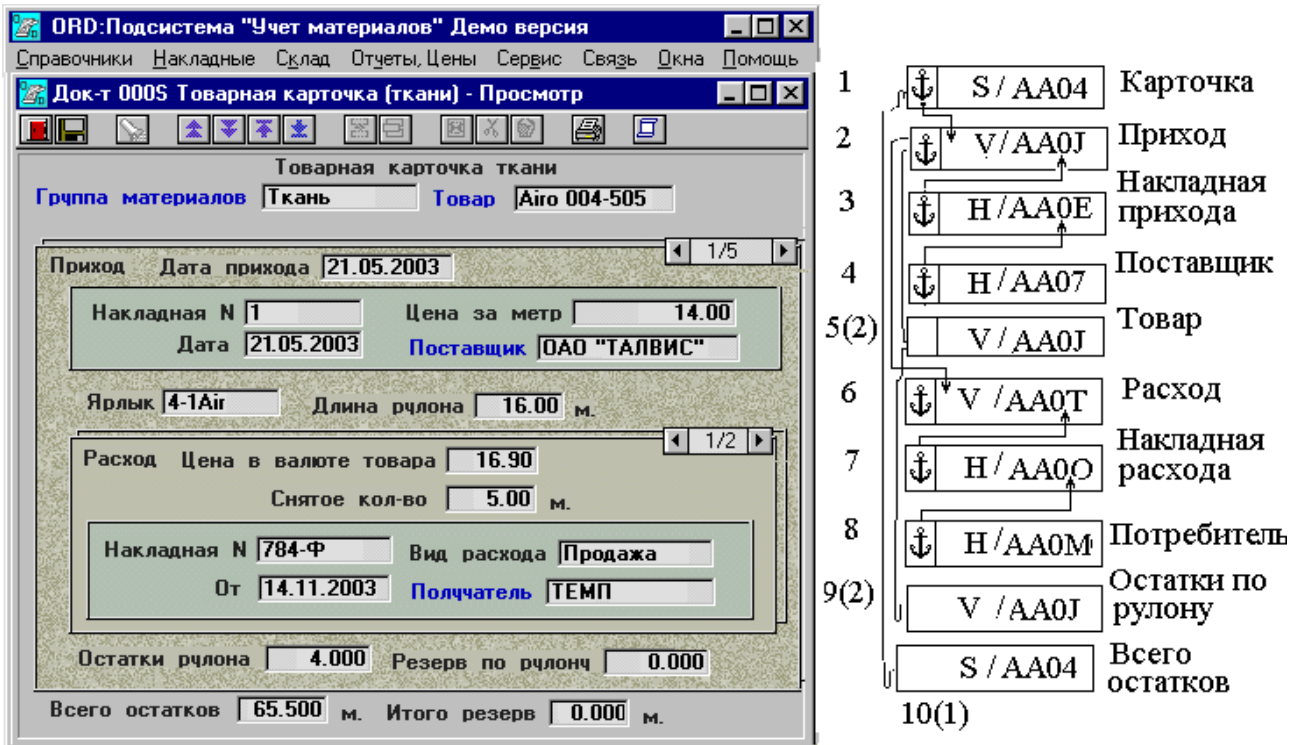


Рис. 15. Модель ИЛМ и образ документа TDOK=000S

В этом документе каждая запись таблицы AA0J соответствует приходу некоторого количества товара (для ткани – это обычно один рулон, или как говорили в старину – одна штука ткани). НАКЛАДНАЯ ПРИХОДА, в которой имеется первоначальная запись об этом приходе, не копируется в карточку, а непосредственно “привязана” к этой записи в структуре документа. Поэтому в нем и будут видны сведения о дате накладной, ее номере, закупочной цене товара и поставщике непосредственно из таблиц AA07 и AA0E, входящих в структуру ‘Накладной прихода’. Эти файлы, как видно из рисунка 15 имеют в ‘Товарной карточке’ статус “Н”, т.е., статус модельного классификатора. (Аналогичным образом включаются в карточку и таблицы расходных документов AA0M и AA0O, о которых говорится ниже.)

Справедливости ради, надо заметить, что использованную структуру товарной карточки можно несколько упростить, исключив из нее одну-две части. Однако на данном этапе не ставилась такая задача.

Вот почти и все с регистрацией поступления закупленных товаров.

Есть, правда, одна деталь. Приход товара на склад, вообще говоря, может иметь различную “природу”. Это и приход от поставщика, это и возврат нереализованного товара из подразделений фирмы, это и поступление на склад изготовленной в собственных цехах фирмы продукции, это и (техническая сторона приложения) начальный ввод остатков. Чтобы не плодить лишних документов (вспомним знаменитый принцип бритвы Оккама), использована только одна форма накладной, в которой имеется поле типа прихода, позволяющее различать перечисленные ситуации.

Более того, поскольку источник поступающего товара в этих случаях также различен по своему виду, выбранный вид прихода определяет, какой из классификаторов: “Организации” или “Подразделения фирмы”, будет использоваться для выбора источника. Этот, довольно мощный прием, реализованный в среде ORD, получил в ней название согласованных подменяемых классификаторов. Некоторую аналогию с ним можно видеть в используемых в ряде систем многоуровневых классификаторах.

С видом прихода связана еще одна очень важная особенность регистрации прихода товара. Дело в том, что в бухгалтерской практике, как минимум ежемесячно, подсчитывается сальдо сторонних организаций.

Для возможности автоматизации этой процедуры при наличии в системе разнообразных товарных и финансовых документов в приложениях ORD, связанных с подобным учетом, используется специальный реестр. Все документы, которые отображают перемещение материальных ценностей (в том числе финансов), между организацией – владельцем системы и ее контрагентами (поставщиками, покупателями, партнерами и пр.), фиксируются в этом реестре. При желании с указанным механизмом можно познакомиться на подсистеме, которая Вам предложена в качестве типовой.

Вид прихода, с которого началось обсуждение этого вопроса и используется для указания того, следует ли учитывать конкретный экземпляр документа в реестре. Поскольку приходные (и расходные) накладные могут описывать разные виды приходов, например, возврат материала из производственного цеха на склад, то сведения о таких накладных не должны попадать в реестр. Анализ видов прихода (и расхода) и позволяет выбрать такие накладные, сведения о которых, должны быть зафиксированы в реестре.

Другим важным бизнес процессом, осуществляемым в фирме “МеДина Лтд”, является продажа товаров (произведенных или закупленных) со склада фирмы. Понятно, что расход товара со склада также должен фиксироваться. Для этой цели служит тип документа НАКЛАДНАЯ РАСХОДА. Номер типа этого документа TDOK=000E.

Конечно, расход, также как и приход, должен отражаться и в ТОВАРНОЙ КАРТОЧКЕ.

Структура НАКЛАДНАЯ РАСХОДА, показанная на рисунке 16, похожа на структуру НАКЛАДНОЙ ПРИХОДА (рис. 13), но имеет небольшие отличия.

В первой части (таблица AA0M, рис. 17), содержится ровно по одной записи для каждого экземпляра накладной (аналогично таблице AA07 “Накладной прихода”). В этой записи, соответствующей шапке накладной, записываются номер накладной, дата передачи товара, вид расхода, признак проведения накладной, валюта накладной, получатель товара, а также итоговые данные по накладной.

Во второй части построенной на файле AA0O (аналогичному файлу AA0E ‘Накладной прихода’) основной табличной части может содержаться несколько записей для каждой накладной. Каждая запись соответствует одному виду передаваемого по накладной товара. В ней записывается группа товара, его наименование и склад, с которого выдается товар. Однако, в отличие от файла AA0E ‘Накладной прихода’ количество товара фиксируется не здесь.

ORD:Подсистема "Учет материалов" Демо версия

Сводное Меню Справочники Накладные Склад Отчеты, Цены Сервис Связь Окна Помощь

Док-т 000E Накладная расхода - Просмотр

Накладная N (Расход) 01 Вид расхода Продажа Дата 21.05.2003 Валюта док. \$
 Получатель ОАО "ТАЛВИС" 1-проведен 1

Состав накладной

Группа Ткань Товар Аіго 04-505 Склад Для тканей

Выбор на складе

Выбранный товар	Кол-во снятое	Остаток в рчлоне
Ярлык П-1Аіг	6.000	10.000
Текущие цены	Цена \$ 16.90	Сумма \$ 101.40
	Цена в вал.док. 16.90	Сумма . 101.40
	Цена 16.90	Сумма 101.40

Всего по материалч 27.50 м. Сумма за материал 472.15

Итого по накл. \$ 472.15 Оплачено \$ 0.00
 Оплачено рчб 0.00

Комментарий Долг \$ 0.00

Рис. 16. Образ документа TDOK=000E



Рис. 17. Модель ILM документа TDOK=000E

Количество товара ушедшего в реализацию фиксируется в части 4 (Итог по товару, рис. 17.), после выбора его со склада и фиксации его в части 3 (Подбор товара, рис. 17).

В файле AAQQ (аналогичному файлу SAYY ‘Накладной прихода’) вложенной табличной части может содержаться несколько записей для каждого вида товара, т.е., для каждой записи основной табличной части. В записи этого файла фиксируется, какие именно единицы хранения (ярлыки), в каком количестве и по какой цене выбыли со склада по данной накладной.

Отличие от ‘Накладной прихода’ состоит в дополнительной части продолжении второй части документа (на рис. 17 она обозначена 4(2)). Это вызвано тем, что при выписывании ‘Накладной расхода’ в начале подбираются реализуемые единицы хранения, а затем автоматически (в части 4) подсчитываются итоговые значения по продаваемому (или передаваемому) материалу.

Конечно, основные особенности регистрации расхода соответствуют операциям, выполняемым при приходе. Иначе говоря, накладная, одинаковая по форме, используется для разных типов передач товара со склада;

- регистрация расхода отображается автоматически в ‘Товарной карточке’;

- если вид расхода говорит о передаче товара вовне, то накладная автоматически фиксируется в реестре. В части 1 в классификаторе ‘Вид расхода’ указан диапазон возможных видов передач: {Продажа, Передача, Списание, Резерв}, при этом выбор одного из них определит для системы, какой из классификаторов ‘Получателя’ (рис. 17, часть 1, SFOR-TDOK=0102, ST05-TDOK=0104) может быть использован для выбранного вида расхода. На рис. 17 информация вида: SFOR-TDOK=0102 указывает, что в качестве классификатора может быть использован файл с именем SFOR, определенный в документе типа “0102”. TDOK=0102 соответствует документу с именем “Поставщики-Покупатели”, а TDOK=0104 документу “Подразделения фирмы” (рис. 1). Подобное правило совместного (связанного) использования объектов

системы указывается с помощью правил диапазона, которые подробно рассмотрены в [6].

Для нашего примера подобное правило фиксируется в ИЛМ следующим предложением для части 1 (рис. 18 а):

Q Получатель ({SFOR-TDOK=0102, ST05-TDOK=0104} DIAP SFOR TO AA5C *S

Для части 2, в которой выбор из “Группы” определяет “Товар”, а последний в свою очередь определяет “Склад” данное правило можно увидеть на рисунке 18 б.



Рис. 18. Модель ИЛМ документа TDOK=0008 и TDOK=000E

На рисунке 18 для простоты в правилах DIAP опущено служебное слово TDOK. Кроме того, показано, что если модель TDOK=000E связать с помощью Н-связи документом TDOK=0008, то отпадает необходимость в документе TDOK=000S (рис. 15 или рис. 19).

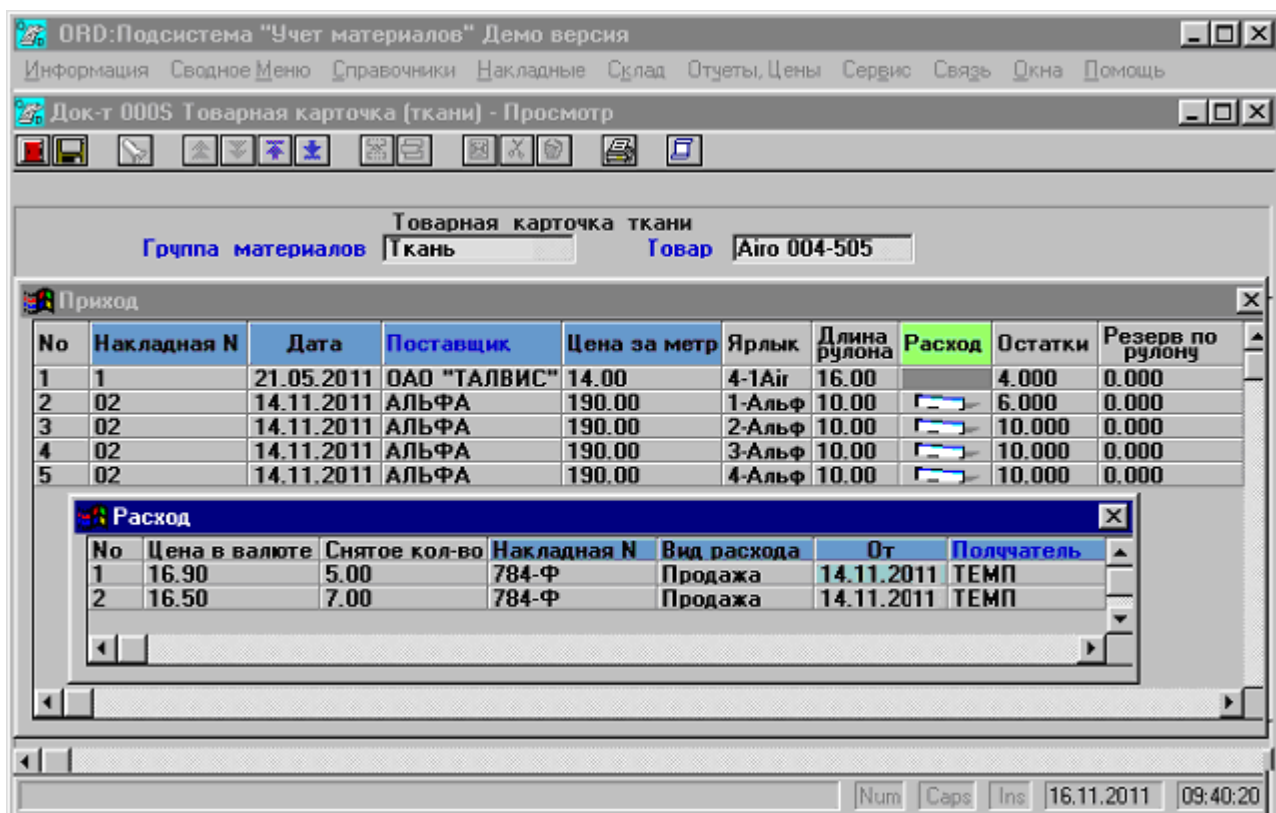


Рис. 19 Товарная карточка ткани Айго 004-505

Ну, вот, пожалуй, и все об основных процессах складского учета в нашей фирме “МеДина Лтд” и в типовом курсовом проекте. Теперь, наверное, есть смысл немного поговорить о программировании, которое должно присутствовать в любом проекте.

Обычно необходимость в программировании возникает в тех случаях, когда средств, предоставляемых системой, не достаточно для удовлетворения специфических требований заказчика.

Рассмотрим процесс ввода информации в документ “Накладная прихода” (рис. 13 или рис. 22) и поясним, в чем же проявляются особенности требований к АРМ со стороны заказчика.

После ввода номера накладной и даты, пользователь должен выбрать из списка *Q-Вид прихода* (таблица с именем AA5M) одно из значений: {Внешний, Внутренний, Ввод остатков, Возврат}. Выбранное значение из *Q-классификатора AA5M* определит область допустимых значений для *Q-классификатора Поставщик* (правило DIAP SFOR TO AA5M *S). При этом в качестве объектов связи могут выступать одна из сущностей, зафиксированных

в системе, с именами типов документов TDOK = "0102" (рис. 20 а, показывает подмножество контрагентов фирмы, определенных в системе в виде справочника ОРГАНИЗАЦИЯ, с головным файлом SFOR) и TDOK = "0104" (рис. 20 б, ПОДРАЗДЕЛЕНИЯ ФИРМЫ с головным файлом ST05).

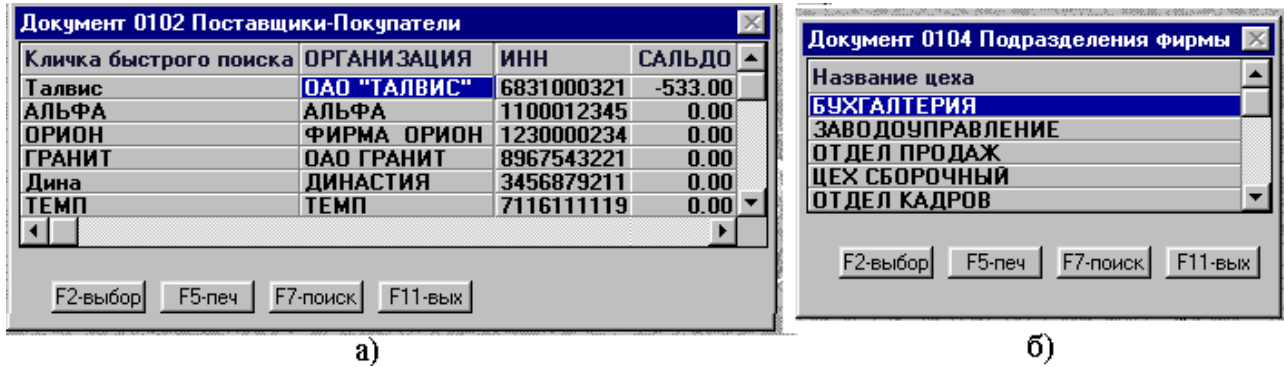


Рис. 20 Справочники подсистемы TDOK=0102 и TDOK=0104

Вторая часть типа "v" предназначена для отображения многозначных свойств накладной, т. е. для отображения "Состава накладной". Каждый слой части 2 может характеризоваться следующими атрибутами: *Группа товара, Наименование товара, Тип единицы товара, Кол-во\длина, Ед. измерения, На склад, Кол-во (факт) единиц товара, Цена закупки единицы, Всего факт, Стоимость фактическая, Всего по док-ту, Стоимость по док-ту.*

Процесс заполнения элементов второй части начинается с выбора элемента из Q-классификатора ГРУППА (TDOK = "0007", основной файл документа AA0F, рис. 21 а). Выбор какого-либо значения из экземпляров документа TDOK = "0007" (имя документа в меню "ГРУППЫ МАТЕРИАЛОВ") определит область допустимых значений для объекта TDOK=000G=ТОВАР (правило DIAP AA0F TO AA02 S, см. рис. 18) и частично определит область допустимых значений объекта СКЛАД (TDOK = 000F, рис. 20 в), на котором должен храниться полученный товар. Заметим, что здесь в качестве файла Q-классификатора, используется не основной файл документа TDOK = "0007", а внутренний файл AA64. Данное свойство уточнено символом "v" в предложении: 'Q На склад (AA64 – "0007 v") DIAP

AA64 TO AA02 S', рис. 18. Данная связка указывает, что товар с одним и тем же наименованием может храниться на разных складах.

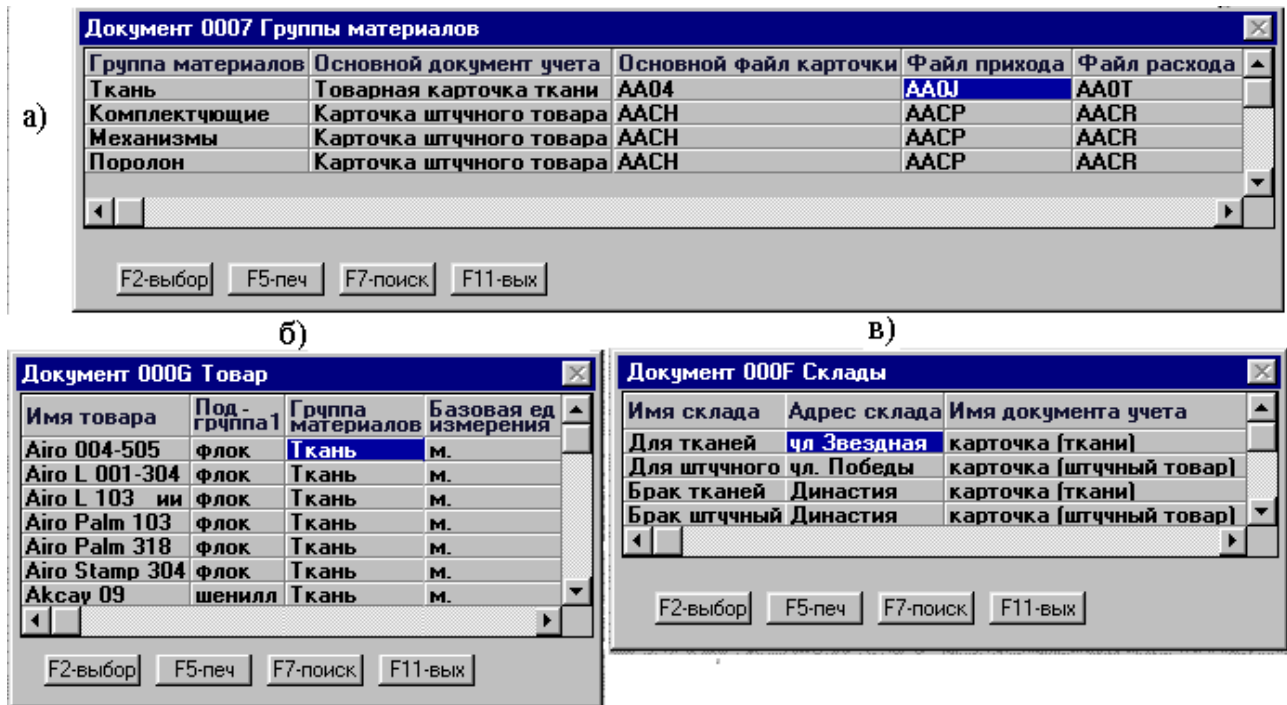


Рис. 21 Справочники подсистемы TDOK=0107, TDOK=010G, TDOK=000F

Выбрав необходимый товар, следует указать значение из К-классификатора “Тип единицы товара” ({Рулон, Ящик, Упаковка, Бокс, Штука}) и заполнить GET поле “Кол-во\длина”, при этом автоматически (с помощью встроенной функции) определится “Единица измерения” выбранного товара, которая будет отображена в других аналогичных полях экранной формы.

После уточнения СКЛАДА (рис. 20 в), необходимо указать значения полей *Кол-во (факт)* и *Цена закупки* за единицу товара. Все остальные поля система заполнит автоматически (с помощью встроенных системных функций), в том числе и все элементы слоев части 3 (рис. 13). Отметим, что значение, указываемое в поле *Кол-во (факт)* автоматически определяет число слоев в части 3.

Третья часть характеризует распределение каждой единицы пришедшего товара по складам и может состоять из следующих атрибутов: *Ярлык единицы*, *Фактическое кол\длина*. Хотя система автоматически формирует указанные поля, их значения открыты для редактирования. Это необходимо, прежде всего,

для поля *Фактическое кол\длина*, в котором в случае необходимости отмечается фактическое количество (размер, вес, длина) некоторой единицы товара, отличающееся от величины указанного в поле *Кол-во\длина*. Если, такая ситуация наблюдается (есть недостача или наоборот), то система естественно пересчитывает значения полей: *Всего (Док)*, *Стоимость (Док)* и *Итого по документу* (как и показано на рис. 22, когда при приемке товара кладовщиком обнаружилось что у рулона с маркером 4-1Air не хватает 6 метров).

Четвертая часть является итоговой (она продолжение части 1) и в ней содержатся атрибуты: *Итого фактически по накладной* и *Итого по документу*, значения указанных полей формируются автоматически (рис. 22).

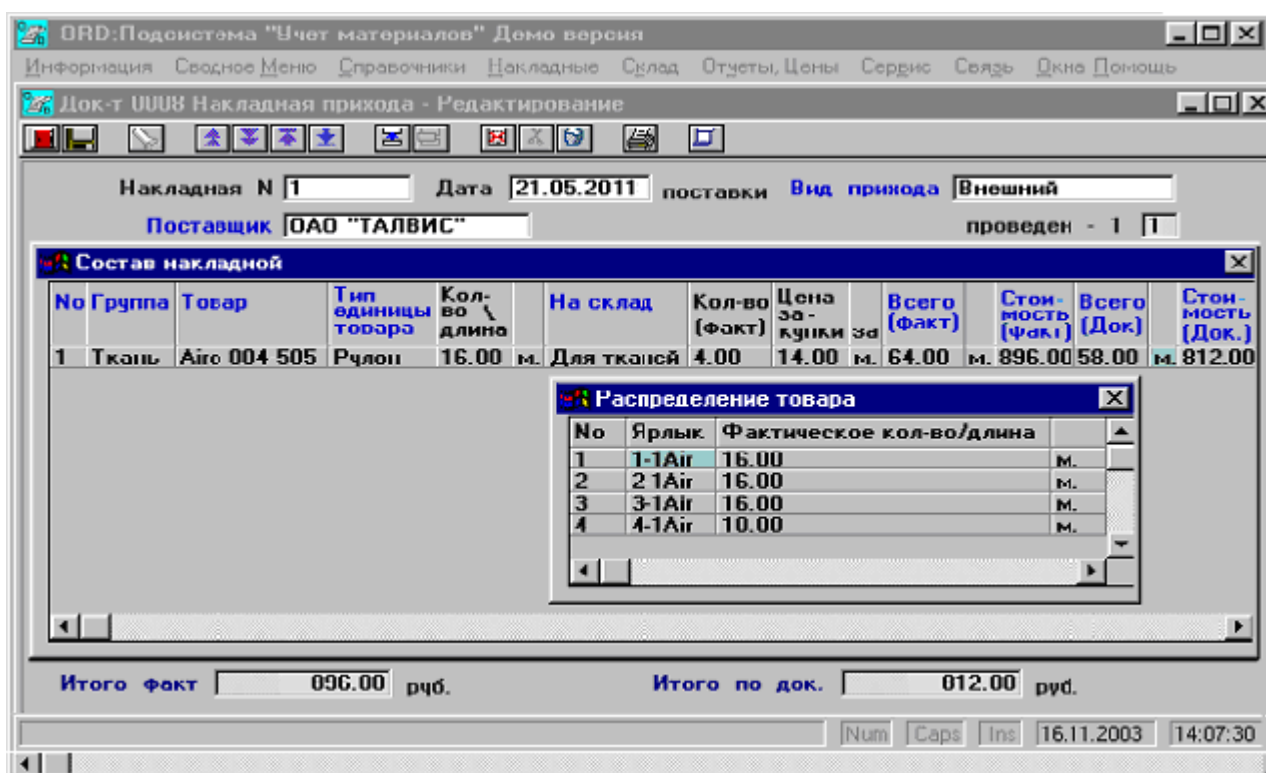


Рис. 22. Один из видов отображения на экране документа TDOK=0008

При описании процесса заполнения документа TDOK=000E, как указывалось, значения некоторых полей формируются автоматически системой. Это значит, что в описании модели хранящейся в таблице SIUEK данного типа документа указан вызов либо системной, либо пользовательской функции.

К системным функциям, встраиваемым в модель документа, обычно относят функции, которые обеспечивают функциональную целостность модели документа и системы в целом. К ним относятся: -

FUNIQ – целостность реляционного ключа (если он указан в объекте);

FIGA - контроль прав доступа и привилегий, регистрация процесса проведенного пользователем;

ORDRANGE – диапазон возможных значений для поля контролируемого функцией, в том числе и значение по умолчанию;

ORDDIAPZ – предусловие и Valid условие Get поля;

ORD_KONT - контроль действий производимых в части, в том числе вход и выход из части;

SUMM – арифметические операции с значениями Get полей документа.

Для отражения правил формирования элементарных составляющих экземпляров документа следует воспользоваться стандартными средствами системы ORD, либо использовать язык программирования целевой СУБД. Примеры написания пользовательских функций приведены в следующем параграфе. Здесь только отметим, что исходные тексты всех функций доступны разработчику и находятся в папке SUBS.

Рассмотрим пример использования системной функции SUMM в документе TDOK – 0008, которая может быть использована для формирования значений следующих полей (рис. 22):

Всего (факт)=Кол-во\длина * Кол-во (факт);

Стоимость (факт)=Всего (факт) * Цена закупки;

Всего (док)= SUM (Фактическое кол-во/длина);

Стоимость (док)= Всего (док) * Цена закупки;

Итого факт = Всего (факт) * Цена закупки;

Итого по док. = Всего (док) * Цена закупки

Отметьте, что первые четыре суммы относятся к каждой единице товара указываемой во второй части (рис. 13 или 22), а последние две суммы относятся целиком к документу. (Указанные расчеты система выполняет без

привлечения программиста, и встраивать функцию SUMM рекомендуется из визуального режима модификации образа документа).

Ниже приводится фрагмент таблицы SIUEK (рис. 23) в которой система автоматически сформировала описание вызова SUMM для вычисления значений:

Итого факт = Всего (факт) * Цена закупки (SUMM(aa4с,А));

Итого по док. = Всего (док) * Цена закупки (SUMM(aa59,А)),

которые находятся в четвертой части документа представленного на рис. 22.

N образа	Имя атрибута на экране	Координаты		Файл	Поле	Координаты поля		Поиск	Тип поля	Точность		
		Упр.	Y			X	Y			X	Длина	
UA008001	Продолж	X	0	0	AA07		1	13	0		1	
UA008001	Дубль Рез.ORD	X	0	0	AA07	AA2J	0	0	0	N	10	2
UA008001	Итого факт	P	27	2	AA07	aa2j	27	20	0	N	10	2
UA008001	SUMM(aa4с,А)	X	0	0			0	0	0		0	0
UA008001	(а)	X	0	0			0	0	0		0	0
UA008001	руб.	Z	27	30			0	0	0		0	0
UA008001	Дубль Рез.ORD	X	0	0	AA07	AA61	0	0	0	N	10	2
UA008001	Итого по док.	P	27	40	AA07	aa61	27	53	5	N	10	2
UA008001	SUMM(aa59,А)	X	0	0			0	0	0		0	0
UA008001	(а)	X	0	0			0	0	0		0	0
UA008001	руб.	Z	27	63			0	0	0		0	0

Рис. 23. Оформление системных функций в SIUEK

Более подробно на особенностях системной функции SUMM остановимся в следующем параграфе.

4.5. Проектирование программ в среде ORD

В приложениях ORD используется несколько видов хранимых процедур, вызов которых обязательно должен быть указан в таблице SIUEK. Система отличает такие объекты от других по наличию в поле Управления (Упр.) таблицы SIUEK одного из символов следующего множества: {D, U, P, A, W, Y

Процедуры могут не входить в структуру документов в приложениях, построенных в среде ORD. Однако они бывают полезными, когда необходимо выполнять скрыто для пользователя большие объемы работ по вычислениям значений, контролю ввода, определению контекстных особенностей обработки документа, ограничению доступа к информации и т.д.

В системе имеется большой набор типовых процедур, позволяющих выполнять многие действия, вообще не прибегая к программированию. В частности, к таким процедурам можно отнести подсчет итоговых сумм по накладным, арифметическую обработку групп полей, например, подсчет стоимости по количеству товара и его цене, и ряд других функций, перечень которых приведен в описании системы.

Здесь только отметим, что в зависимости от назначения функций они разделяются на следующие типы:

- "U" - стандартная (системная) функция с именем FUNIQ, предназначена для проверки уникальности значения, вводимого в GET поле;

- "P" - функция вызывается на исполнение при нажатии клавиши Enter на объекте, с которым связана функция;

- "A" - функция вызывается на исполнение при прохождении фокуса через объект, с которым связана функция;

- "W" - функция, исполняемая после нажатия пользователем клавиши Выход с сохранением", информация документа к моменту исполнения данной функции находится в буферах системы и не скинута на диск;

- "Y" - функция, исполняемая после записи информации документа в файлы (информация из буферов системы к моменту исполнения данного типа функции передана в файлы системы, и транзакция завершена успешно);

- "D" – функция, контролирующая введенное значение в Get поле.

В качестве простейшего примера использования функций пользователя рассмотрим, например, для документа представленного на рис. 2 контроль ввода информации в поле с именем атрибута *Дата приема* на предмет обязательного его 'означивания' в документе. Отметим, что обеспечить подобный контроль можно различными системными средствами, так, например, если среди свойств данного поля указать свойство NOT NULL – проблема разрешится автоматически.

Используем для решения поставленной задачи функцию типа “D” с именем ADIAST29. Функции подобного типа всегда должны располагаться во внешней модели документа сразу за полем, значения которого они должны контролировать. Ниже на рисунке 24 приведен фрагмент таблицы SIUEK содержащей описание функции с именем ADIAST29.

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9070	UA305001	Сотрудник	Z	1	30			0	0	0		0	0
9084	UA305001	Дата приема	G	10	25	LA01	Lapr	10	34	0	D	8	0
9085	UA305001	ADIAST29 Lapr	D	0	0			0	0	0		0	0

Рис. 24. Оформление пользовательских функций в SIUEK

Примечание. Если возникают вопросы по синтаксису любой функции, следует нажать клавишу F12, находясь в столбце Упр. (SI70) интересующей Вас строки. Каждый столбец таблицы имеет и уникальное латинское имя, поэтому иногда вместо русского наименования столбца используется его латинское обозначение.

Используя входной язык системы ORD, пользователь может написать следующий текст функции.

FUNCTION ADIAST29 (F1, F2, F3, F4, F5, F6)

```
//Система для функций типа "D" всегда передает следующий набор параметров:
//F1 - индекс массива Sary-Siuek (см. пояснения ниже, перед рисунком 26), где
//находится поле, контролируемое функцией;
//F2 - код последней нажатой клавиши при вызове функции;
//F3 - индекс слоя части (см. пояснения ниже), откуда вызвана функция;
//F4 – значение, введенное в GET поле пользователем;
//F5 - значение до редактирования;
//F6 - динамический индекс части, где находится редактируемое поле.
```

```
If F4=ctod(“)
```

```
/*Вывод сообщения на экран*/
```

```
mSay ("Значение в поле должно быть введено") //Вывод сообщения на экран
```

```
RETURN .f. //Возврат на повтор ввода информации в поле Даты приема
```

```
EndIf
```

```
/*Естественно, функция, использованная в одном документе, может использоваться и в других документах.*//
```

```
RETURN .t.
```


Примечание. Информация, находящаяся после символов "/" (или заключенная в символы /* */) относится к комментарию. Набор передаваемых параметров в функцию системой, зависит от типа используемой функции и описан в техническом руководстве. Функция обязана вернуть .t. (истину), если введенное значение удовлетворяет необходимым требованиям.

Действуя согласно приведенного текста, функция контролирует вводимые значения в поле (объект) и выводит сообщение, если оператор ошибается при вводе. Чтобы функция была доступна системе, она должна быть скомпилирована и скомпонована с системой (для этого предоставляются соответствующие командные (*.bat) файлы или указанные действия производятся из инструментальной оболочки Db.exe).

Строка "9085" (рис. 23) содержит вызов функции типа "D". Назначение функции должно быть понятно из приведенного текста функции. Если, после прочтения данной функции Вам не потребуется обращаться к техническому руководству системы, то считайте, что Вы научились программировать в системе ORD.

При написании любой программы в системе следует учитывать, что это система с открытой архитектурой, позволяющая динамически изменять любой из составляющих ее элементов. Например, могут динамически измениться не только правила взаимодействия понятий (документов или их частей) отображенных в системе, но и правила их рождения и существования. Может измениться не только структура документа, но и структура и состав элементов составляющих части этого документа. Кроме того, ORD это система сетевой распределенной обработки информации в реальном времени, допускающая синхронное функционирование одной и той же подсистемы в различных участках сети, с динамическим распределением хранимой информации по узлам сети. Для того чтобы Ваша программа была независима от проблем приведенных выше, достаточно придерживаться правил программирования, принятых в ORD и система сама будет корректно решать приведенные выше задачи.

Прежде всего, следует уяснить, что система может предоставить средства прямого физического доступа к хранимым таблицам (при наличии

соответствующих прав), минуя такие команды SQL как INSERT (добавить), UPDATE (модифицировать), DELETE (удалить), что требует от программиста определенных навыков и понимания архитектуры системы. Поэтому на данном этапе изучения, рекомендуется использовать *буферную технологию* доступа к хранимой информации, предоставляемой системой, что позволяет избежать всевозможных коллизий связанных с исполнением транзакций и их откатов.

Для каждой части документа, с которой в данный момент работает система, динамически формируется буфер, доступ к нему осуществляется с помощью рабочей таблицы, которая носит в системе название - МАССИВ ЧАСТИ. Таким образом, структура документа на уровне оперативной памяти представляется списком массивов (динамических) связанных в одно целое с помощью якорей. Все вопросы распределения памяти и свопинга автоматически разрешаются системой.

Для человека знакомого с понятием массива – это двумерная таблица, которую можно увидеть на рисунках 24, 23, 22, 12. Например, на рисунке 12 представлен набор массивов логически связанных между собой.

Массив части мало чем отличается от подобных массивов, за исключением того, что элементы некоторых столбцов могут хранить различные типы данных или являться ссылками на другой массив (объект). Важно другое, для обработки такого массива надо знать его имя, число строк и столбцов его составляющих.

Например, для документа изображенного на рис. 8, состав и содержимое массивов части в режиме редактирования для сотрудницы Аблова может иметь вид, приведенный на рисунке 25. На рис. 25 представлено содержимое массивов первой (рис. 25 а) и второй (рис. 25 б) части документа, в которых в реальности нет первых строк (где для удобства показаны имена полей (рис. 12) физических таблиц, а столбцы пронумерованы).

Именно на уровне подобных массивов и работает прикладной программист с хранимой информацией. Кроме столбцов с данными, массив части содержит ряд служебных столбцов, в которых фиксируется полезная и

доступная информация для программиста. Так, например, сразу за информационными столбцами размещается служебный столбец (например, на рис. 25 а) это столбец 16), в котором хранится признак модификации строки (слоя) массива части.

а)

LA01	LTARZ	TAR	LA0F	LA0I	LA0J	LA0L	ZA0L	LA0C	LA0T	LA0D	LAPR		LAST	S305					n
Якорь	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
01	0			Аблова	Вера	Ивановна	1	Ж	2222	001	03.07.67	08.07.93		03.07.87	2	"1"	-1		

б)

LA03	LA01				LGTA	ZGTA	LT05	ZT05	LOKL	LAIZ	12	13			n
1	2	3	4	5	6	7	8	9	10	11	12	13			
001	01				02	Программист	00	Цех1	50000	03.07.93	"1"	-1			
002	01				00	Математик	00	Цех1	7500	03.07.00	"1"				

Рис. 25. Буферные массивы системы.

Значения элементов этого столбца (принятые в системе) следующие:

- "0" – новая информация;
- "1" – слой из файла (не измененный);
- "2" – слой из файла и подвергался изменению;
- "3" – признак удаления записи взятой из файла;
- "4" – признак изменения новой информации.

Так, например, в режиме создания документа (для программиста доступна глобальная переменная с именем UREG, значение которой указывает на режим работы с данным экземпляром документа) значение рассматриваемого столбца будет "0", до тех пор, пока не изменится какой-либо элемент слоя. Если изменение элемента слоя происходит, то признак модификации слоя становится равным "4".

Следующие служебные столбцы редко используются программистом и их назначение описано в технической документации.

Если внимательно посмотреть на внешнюю модель (рис. 26) документа изображенного на рисунке 8, то нетрудно заметить что имена полей указанные в первых строках массивов частей (рис. 25) совпадают с именами полей, указанными в столбце с именем Поле на рисунке 26. В момент обращения к какому-либо документу система динамически отображает внешнюю модель в

системный массив с именем SARY. Поэтому между таблицей SIUEK и массивом SARY существует некоторое соответствие.

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9050	UA305001	Сотрудник	M			LA01	LA01				C	3	
9052	UA305001	Признак увольнения 1	K	1	45	LA01	LTAR	1	64		C	2	0
9054	UA305001	Фамилия, Имя, Отч.	G	3	5	LA01	LA0F	3	15	1	C	15	0
9056	UA305001		G	0	0	LA01	LA0I	3	27		C	12	0
9058	UA305001		G	0	0	LA01	LA0J		41		C	20	0
9060	UA305001	Пол 1	K	3	57	LA01	LA0L	3	65		C	1	0
9061	UA305001	Страховой номер	G	5	5	LA01	LA0C	5	17	3	C	8	0
9067	UA305001	Табельный номер	G	5	25	LA01	LA0T	5	39	2	C	5	0
9070	UA305001	Дата рождения	G	5	47	LA01	LA0D	5	69		D	8	0
9084	UA305001	Дата приема	G	10	25	LA01	Lapr	10	34	0	D	8	0
9085	UA305001	ADIAST29 Larp	D	0	0			0	0	0		0	0
9087	UA305001	Полный стаж	G	10	45	LA01	LAST	10	54	0	D	8	0
9090	UA305002	Перемещения	M	0	0	LA02	La02	0	0	0	C	3	0
9091	UA305002	*9050	N	0	0	LA02	La01	0	0	0	C	3	0
9092	UA305002	1gr+	R	12	5			20	78	0			
9093	UA305002	Перемещения сотрудни	Z	14	10			0	0	0			0
9094	UA305002	ка	Z	14	30			0	0	0			0
9095	UA305002	Должность 1	K	16	15	LA02	La02	16	24	0	C	20	0
9097	UA305002	Цех 1	K	16	37	LA02	La02	16	41	0	C	15	0
9098	UA305002	Оклад/Тариф	G	18	15	LA02	LOKL	18	24	0	N	10	2
909a	UA305002	Дата изменения	G	18	37	LA02	LAIZ	18	41	0	D	8	0

Рис. 26. Таблица SIUEK для документа TDOK=0305.

На рисунке 27 представлена логическая модель для документа изображенного на рисунке 8. Отметим, что в столбцах с именами SI75 и SC75 система указала границы частей документа из таблицы на рисунке 26. Каждый столбец таблицы SIAGF (кроме первого) имеет в оперативной памяти эквивалентное отображение в виде одномерного массива с именами: PSC60, PSI75, PSC65, PSC61 и т.д. Все эти массивы, как и массив SARY, доступны программисту в виде глобальных переменных.

N	Часть	Начало	Конец	Переход	Отказ	Вер-шт	Ста-тус	Эк-ран	Спец 1	Спец 2	Название части
SIIM	SC60	SI75	SC75	SC61	SC62	SC63	SI59	SC70	SC50	SC51	SI51
0305	1	9050	9087	2	99	0	S	1			Сотрудник
0305	2	9090	909a	98	99	1	V	1			Перемещения сотрудника

Рис. 27. Таблица SIAGF для документа TDOK=0305.

Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поис ск		Тип	Дли на	Точн ость
SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72		SI56	SI57	SI76
1	8	2	3	4	5	6	7	9	10	11	12	13

Сотрудник	M			LA01	LA01					C	3	
Признак увольнения	1 k	1	45	LA01	LTAR	1	64			C	2	0
Фамилия. Имя. Отч.	G	3	5	LA01	LA0F	3	15	1		C	15	0
	G	0	0	LA01	LA0I	3	27			C	12	0
	G	0	0	LA01	LA0J		41			C	20	0
Пол	1 k	3	57	LA01	LA0L	3	65			C	1	0
Страховой номер	G	5	5	LA01	LA0C	5	17	3		C	8	0
Табельный номер	G	5	25	LA01	LA0T	5	39	2		C	5	0
Дата рождения	G	5	47	LA01	LA0D	5	69			D	8	0
Дата приема	G	10	25	LA01	Lapr	10	34	0		D	8	0
ADIAST29 Lарг	D	0	0			0	0	0			0	0
Полный стаж	G	10	45	LA01	LAST	10	54	0		D	8	0
Перемещения	M	0	0	LA02	La02	0	0	0		C	3	0
*9050	N	0	0	LA02	La01	0	0	0		C	3	0
Igr+	R	12	5			20	78	0				
Перемещения сотрудни ка	Z	14	10			0	0	0				0
	Z	14	30			0	0	0				0
Должность	1 k	16	15	LA02	La02	16	24	0		C	20	0
Цех	1 k	16	37	LA02	La02	16	41	0		C	15	0
Оклад/Тариф	G	18	15	LA02	LOKL	18	24	0		N	10	2
Дата изменения	G	18	37	LA02	LAIZ	18	41	0		D	8	0

LA03	LA01			LGTA	ZGTA	LT05	ZT05	LOKL	LAIZ					n
1	2	3	4	5	6	7	8	9	10	11	12	13		

001	01			02	Программист	00	Цех1	50000	03.07.93	"1"	-1			
002	01			00	Математик	00	Цех1	7500	03.07.00	"1"				

LA01	LTAR	ZTAR	LA0F	LA0I	LA0J	LA0L	ZA0L	LA0C	LA0T	LA0D	LAPR		LAST	S305				n
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
01	0		Аблова	Вера	Ивановна	1	Ж	2222	001	03.07.67	08.07.93		03.07.87	2	"1"	-1		

Рис. 28. Массив SARY для документа TDOK=0305.

На рис. 28 приведен массив SARY (аналог SIUEK) и показано, что из 10-го столбца каждой первой строки части существует ссылка на массив соответствующей части.

Таким образом, в случае необходимости, можно динамически модифицировать любую модель документа (документов), не нарушая целостности системы.

Для каждого документа определены следующие массивы:

NP [] – число динамических частей в документе;

KP [] – число слоев в части;

KPP [] – число слоев в массиве части;

NPI [] – указатель слоя в части (индекс строки массива).

Для рассмотренного примера (например, в режиме просмотра) в момент раскрытия документа на экране (в плоском изображении) состав указанных массивов будет:

NP [1]=1 KP [1]=1 KPP [1]=1 NPI [1]=1

NP [2]=2 KP [2]=2 KPP [2]=1 NPI [2]=1

Для того чтобы связаться с любым массивом необходимой Вам части в программе необходимо указать следующий оператор, например для части с индексом P:

A: =Sary [Psi75 [Fsiagf+np [P]], 10]

Если известен статический номер части, например 2, то связь может быть осуществлена с помощью оператора:

A: =Sary [Psi75 [Fsiagf+2], 10]

Fsiagf - динамическая база в оперативной памяти данного экземпляра документа.

Номер служебного столбца, в котором находится признак модификации слоя части с индексом P, всегда хранится в первой строке части в седьмом столбце SARY. Получить значение признака модификации слоя можно оператором:

MODISLOI=A [npi [P], sary[Psi75 [Fsiagf+np[P]], 7]]

Рассмотрим пример использования функции типа A для формирования инициалов сотрудника. Пусть имя функции будет AFAMNAME. Например,

пусть по введенной информации Аблова Вера Ивановна необходимо сформировать строку вида Аблова В.И.

Для хранения сформированного значения используем поле с именем LINK (рис. 29, строка 9062), с символом управления S.

Примечание. Модификация внешней модели (добавление или удаление полей или частей документа и т.д.) производится с помощью визуальных средств системы утилиты Db.exe. Результат подобной модификации приведен на рисунке 30.

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9050	UA305001	Сотрудник	M			LA01	LA01				C	3	
9052	UA305001	Признак увольнения 1	k	1	45	LA01	LTAR	1	64		C	2	0
9054	UA305001	Фамилия, Имя, Отч.	G	3	5	LA01	LA0F	3	15	1	C	15	0
9056	UA305001		G	0	0	LA01	LA0I	3	27		C	12	0
9058	UA305001		G	0	0	LA01	LA0J		41		C	20	0
9060	UA305001	Вызов функции	A	0	0								0
9061	UA305001	AFAMNAME	X	0	0								0
9062	UA305001	Ф.И.О.	S	10	55	LA01	LinK	10	59		C	20	0

Рис. 29. Описание вызова функции типа А

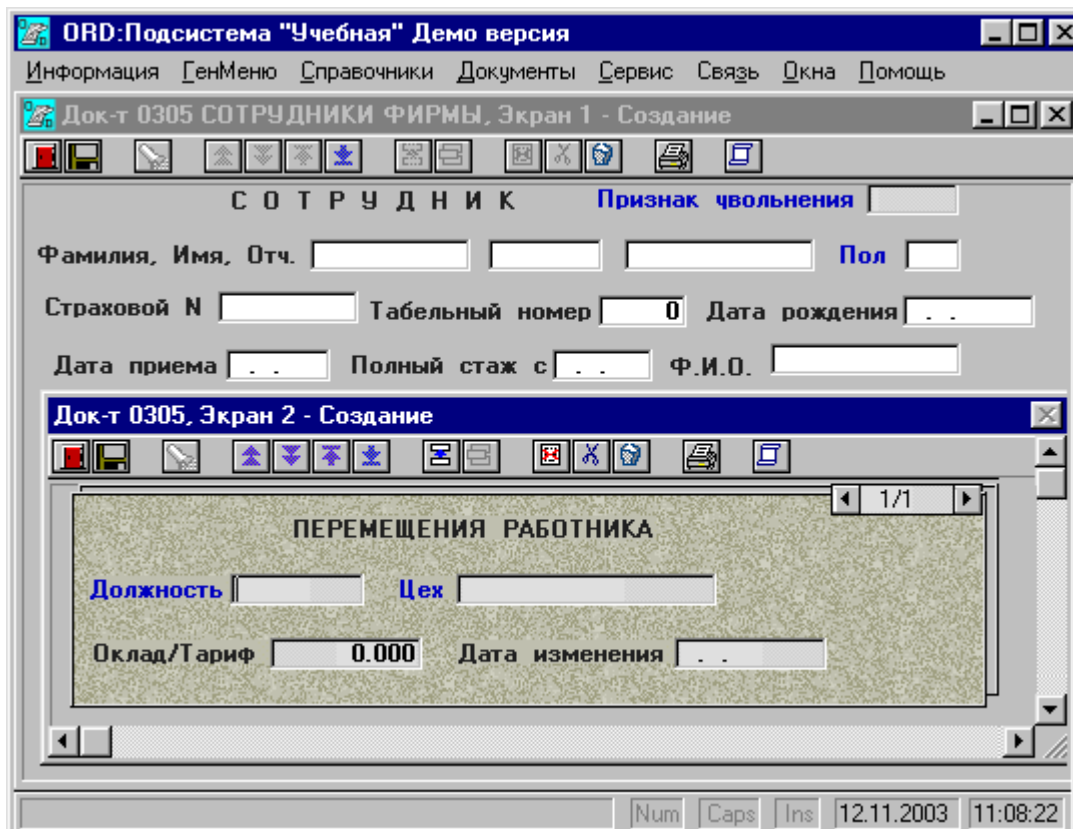


Рис. 30. Модифицированный образ документа из рис. 8

```

FUNCTION AFAMNAME(R, I, T, K, N, P) //
// В функцию системой передаются значения параметров:
//R - режим {"S", "G"}. R="S" - если документ вызван для просмотра UREG=2
//R="G" - документ вызван для создания UREG=3 или редактирования UREG=4
//I - индекс строки Sary, в котором находится функция AFAMNAME.
//T - символ ="A" (признак типа вызванной функции).
//K - код движения по экранному образу документа: K=1, если предыдущее
//движение было вниз; K=2, если предыдущее движение было вверх. Параметр
//K соответствует глобальной переменной KOD, которая указывает, какое
//предыдущее действие, выполнила система. Полное описание переменной
//KOD, можно найти в ORD0.hlp;
//N - индекс строки массива части (слоя части) на котором в данный момент
//вызвана функция (для однослойных частей N всегда равен 1).
//P - динамический индекс части, откуда вызвана данная функция
Local A, MODSLOI, I1

I1= Psi75 [Fsiagf+np [ p ] ]

      A: = Sary [I1, 10]

/*Приведенный оператор позволяют связаться с массивом части и обрабатывать
его как обычный двухмерный массив. Столбцам этого массива будут
соответствовать строки Sary (Siuek) данной части, а строкам этого массива
будут соответствовать слои части (поля записей файлов участвующие в
формировании образа части).*/
      MODSLOI=A [1, sary[ i1, 7 ] ]

IF MODSLOI $ [24] // Слой модифицировался
      Y=ascan (SARY, {|X| X[5]="LA0F"})//индекс атрибута "Фамилия" в SARY
      yi=ascan (SARY, {|X| X[5]='LA0I'}) //индекс атрибута "Имя"
      yo=ascan (SARY, {|X| X[5]='LA0J'}) //индекс атрибута "Отчество"
      yn=ascan (SARY, {|X| X[5]='LinK'}) //индекс атрибута "Ф.И.О."
      ysf=A [1, Y-I1+1] //значение атрибута "Фамилия"
      ysi=A [1, Yi-I1+1] //значение атрибута "Имя"
      yso=A [1, Yo-I1+1] //значение атрибута "Отчество"

      A [1, yn-I1+1]=alltrim(ysf)+left(ltrim(ysi), 1)+'.'+Left(ltrim(yso), 1)+'.'
/*запись фамилии и инициалов в массив части. Система выведет
полученное значение на экран автоматически, если использовать
следующий оператор SII_OSAY(p, i1, psc75 [Fsiagf+np [p]])*/
EndIF
RETURN .t.

Естественно представленное тело функции может быть представлено
одним оператором, но его будет трудно прочитать.

SARY[Psi75[Fsiagf+np[p]],10] [1, ascan (SARY,{|X| X[5]='LinK'})-;
Psi75[Fsiagf+np[p]]+1] = alltrim(SARY[Psi75[Fsiagf+np[p]],10][1, ascan (SARY,;

```



```
{|X| X[5]='LA0F'}) - Psi75 [Fsiagf+np[p]]+1) )+left(ltrim (SARY[Psi75[Fsiagf + ;
np[p]],10][1, ascan (SARY,{|X| X[5]='LA0F'})- Psi75[Fsiagf+np[p]]+1)),1)+;
'+left(ltrim(SARY[Psi75[Fsiagf+np[p]],10] [1, ascan (SARY,{|X| X[5]='LA0F'}) -;
Psi75[Fsiagf+np[p]]+1)),1)+'
```

В данной программе используются следующие функции системы ORD (полное описание указанных ниже функций приведено в техническом руководстве):

- ascan(<имя массива>, <выражение поиска>, [<индекс строки с которой надо начать поиск>], [<кол-во просматриваемых элементов>]).

Функция Ascan возвращает номер первого элемента, совпадающего со значением <выражение поиска>, или нуль, если ни один элемент не удовлетворяет поиску. Например:

```
Yn = ascan (SARY, {|X| X[5]='LinK', i1, 15}) //индекс атрибута "Ф.И.О."
```

В 5ой размерности массива SARY необходимо найти элемент 'LinK'. Поиск начать со строки i1 и просмотреть 15 элементов. (Заметим, что любая функция системы может быть заменена на Вашу функцию, которая выполняет подобные действия).

- alltrim(<символьное выражение>) – Удаляет ведущие и замыкающие пробелы в строке символов.

- ltrim(<строка символов>) - Удаляет начальные пробелы из символьной строки.

- left(<строка символов>,<длина подстроки>) - Определяет подстроку, начиная с первого символа в строке.

Таким образом, процесс программирования любой задачи в системе ORD сводится к выбору из библиотек системы необходимых функций и группировки их в необходимой последовательности. Естественно, что принцип открытой архитектуры распространяется и на библиотеки системы.

Отметим, что приведенная функция имеет существенный недостаток, так как она учитывает структуру документа, что не позволит ее использовать в других документах, и, кроме того, если данный документ будет преобразовываться в другой документ системы, она не будет там доступна.

Поэтому рекомендуется оформлять подобные функции независимыми от приложений. Например, если заменить имя рассмотренной функции, на имя ORD_A102, и оформить вызов функции, как показано на рис. 31 (строка 9061), то отпадают отмеченные выше проблемы и функция становится свойством объекта.

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9050	UA305001	Сотрудник	M			LA01	LA01				C	3	
9052	UA305001	Признак увольнения 1	K	1	45	LA01	LTAR	1	64		C	2	0
9054	UA305001	Фамилия. Имя. Отч.	G	3	5	LA01	LA0F	3	15	1	C	15	0
9056	UA305001		G	0	0	LA01	LA0I	3	27		C	12	0
9058	UA305001		G	0	0	LA01	LA0J		41		C	20	0
9060	UA305001	Вызов функции	A	0	0								0
9061	UA305001	ORD_A102(4,Link,LA0F,;	X	0	0								0
9062	UA305001	LA0I,LA0J)#YW.	X	0	0			0	0			0	0
9063	UA305001	Ф.И.О.	S	10	55	LA01	LinK	10	59		C	20	0

Рис. 31. Оформление “независимой” функции ORD_A102

Исходный текст функции, как и других, находится в папке SUBS, здесь его приведем, чтобы пояснить некоторые моменты оформления функции.

Если в описании функции (любого типа) указать имена полей (файлов) с которыми работают в теле функции, то такая функция получает статус свойства объекта. В этом случае в качестве параметра передается массив индексов элементов SARY, участвующих в операциях функции. Так, например, для рассматриваемого примера в функцию ORD_A102, будет передан следующий массив:

```

B[1]=ascan (SARY,{|X| X[5]='LinK'}) индекс атрибута "Ф.И.О."- поле результата
B[2]=ascan (SARY,{|X| X[5]='LA0F'}) //индекс атрибута "Фамилия" в SARY
B[3]=ascan (SARY,{|X| X[5]='LA0I'}) //индекс атрибута "Имя"
B[4]=ascan (SARY,{|X| X[5]='LA0J'}) //индекс атрибута "Отчество"

```

Если в строке 9062 указан символ '#', то значение следующего символа {Y,N} регламентирует вывод на экран результата функции. "Y" - выводить результат на экран, N - не выводить результат на экран.

Если следующим символом в рассматриваемой строке будет "W", то это обозначает использование этой функции и как функции типа "W", т.е. функция должна быть вызвана на исполнение перед записью информации на диск.

```

memvar sary,pSI75,pSC75,fSiagf,ureg,kod,np,npI,psi59,crash,KPP
FUNCTION ORD_A102 (R, i, T, K, n, p, B) //тип А
//B 13-ом столбце i+1-ой строки - число строк в функции
Local A,ysf,ysi,yso,i1
if kod = 11 .or. kod = 3 // Ctrl\F8
// идет удаление документа kod = 11 // if kod=3 выход без сохранения
    RETURN .t.
endif
if ureg<>3 //не просмотр
    i1=pSI75 [fSiagf+np[p]] //начало части в Sary
    A:=sary [i1,10] //массив части где находится функция ORD_A102
if T="A" // вызов как функции типа А
    ysf=A [n, B[ 2 ]-I1+1] //значение атрибута "Фамилия"
    ysi=A [n, B[ 3 ]-I1+1] //значение атрибута "Имя"
    yso=A [n, B[ 4 ]-I1+1] //значение атрибута "Отчество"
    A[n, B[1]-I1+1]=alltrim(ysf)+left(ltrim(ysi),1)+'.'+left(ltrim(yso),1)+'.'
    SII_OSA Y(p,i1,pSC75 [ fsiagf+np [ p ] ]) //вывод на экран
Else //вызов как функции типа W
    For j=1 to kpp[p] // по числу слоев в массиве части
        ysf=A [j, B[ 2 ]-I1+1] //значение атрибута "Фамилия"
        ysi=A [j, B[ 3 ]-I1+1] //значение атрибута "Имя"
        yso=A [j, B[ 4 ]-I1+1] //значение атрибута "Отчество"
        A[j, B[1]-I1+1]=alltrim(ysf)+left(ltrim(ysi),1)+'.'+left(ltrim(yso),1)+'.'
    Next
Endif
Return Nil

```

Рис. 32 Исходный текст программного модуля

Отметьте, что тело функции (рис. 32) оформлено так, что оно не зависит от структуры документа, в котором она используется. Кроме того, вызов функции как типа “W”, позволяет программисту еще раз проверить ввод информации и в случае необходимости запретить запись информации в базу данных. Для этих целей используется глобальная переменная с именем CRASH, если ее значение будет отлично от нуля, то система не допустит запись информации на физический носитель.

Заметьте, что проблемы связанные с совместной обработкой отсутствуют, независимо от того работаете ли Вы в сети или на одном компьютере.

4.5.1. Компиляция и передача пользовательских программ в систему

Для того чтобы проверить синтаксическую правильность написанных Вами программ необходимо их скомпилировать, используя следующий командный файл (Cm.bat):

```
CLS
SET INCLUDE=D:\Clip53a\INCLUDE
D:\Clip53a\BIN\clipper %1 /m/a/w/n > 1.txt
```

Рис. 33. Исходный текст командного файла

Примечание. В примерах используется диск D, имя которого при необходимости можно заменить.

Например, чтобы скомпилировать функцию AFAMNAME.prg необходимо в командной строке указать: cm AFAMNAME.prg. Если, в процессе компиляции будут найдены ошибки, их можно будет увидеть в текстовом файле 1.txt. В случае успешной компиляции функции ее имя необходимо указать в файле сценария компоновки ORD.lnk, который приведен ниже:

```
BLINKER INCREMENTAL OFF           //получение динамического оверлея
исполняемого

BLINKER CLIPPER SYMBOL OFF //модуля

file t           //головная системы
fi flp          //служебная

// меню подсистемы
file L0win       // головное меню
file L0all       // сводное меню
fi AFAMNAME     // новая программа

NOBELL // отключение звукового сигнала окончания процесса
// Первыми должны идти библиотеки ORDa
lib ordQ_win // Query
lib OrdK_win // классы
lib ordC_win // Си
lib ord0_win // система
lib ord1_win // система
```

```

lib ord2_win // система
lib ord_PRN // печать
lib ordP_win // пользовательские

LIB Five, FiveC, Objects
LIB WinApi, Clipper, Extend, DbfCDX, _dbfcdx
lib import, cwl

DEFBEGIN
    name          ORD_DBMS
    description    'ORD for Windows library'
    exetype       Windows
    code          preload moveable discardable
    data          preload moveable
    stacksize     8000
    heapsize      8192
    # heapsize    2048
    # Для NT и 2000 heapsize не менее 2048
DEFEND

```

Рис. 34. Исходный текст файла сценария

Для того, чтобы провести компоновку разработанной программы с системой, необходимо воспользоваться командным файлом 3.bat, текст которого приводится ниже:

```

set lib=d:\clip53a\lib;d:\fw192\lib
d:\clip53a\bin\blinker @ord nobell > 3
if not errorlevel 1 d:\clip53a\bin\crkpp t.exe
if not errorlevel 1 call 3r.bat

```

Рис. 35. Командный файл компоновки

Ошибки компоновки будут зафиксированы в текстовом файле 3. После исполнения 3.bat будет сформирован исполняемый модуль T.exe, и, запустив его, Вы можете увидеть на экране результат разработанной Вами программы. Система является полностью открытой, поэтому Вам предоставляются исходные тексты программ, на которых она работает.

Если же возникает необходимость получить DOS версию системы, то следует перейти в папку ... \PRG\ и выполнить компиляцию и компоновку модуля C.exe. Тексты Ваших программ (без какой-либо переделки) написанных под Windows будут работать и в DOS режиме. Имя исполняемого модуля под Windows имеет название t.exe и расположено в папке ... \PRG_ALS\ ... каждой

информационной подсистемы. Имя командного файла для компиляции Cm.bat, а для сборки A.bat.

4.5.2. Общие требования к именам программ, файлов и полей файлов в системе ORD

4.5.2.1. Структура имен программ

Имя прикладной программы (процедуры или функции) должно иметь следующий вид:

<имя подсистемы><имя программы>.prg,

где <имя подсистемы> - один символ, определяющий имя информационной подсистемы, в которой используется программа, например:

O - основные средства;

M - материальные ценности;

U - системная, (общие дублированные файлы системы);

S - зарезервировано;

<имя программы> - семь символов, если это процедура, используется в обработке документа (т.е. программа типа: A, P, W, Y, U).

Имена общесистемных программ имеют вид:

S <имя программы>.prg,

где S - указатель общесистемной программы.

4.5.2.2. Структура имен файлов .DBF

DBF файлы, используемые в системе ORD, подразделяются на:

- собственные файлы информационных подсистем;
- разделяемые файлы информационных подсистем;
- системные информационные файлы;
- общесистемные файлы;

- системные управляющие файлы.

Собственным файлом информационной подсистемы называется файл, используемый только в этой подсистеме.

Разделяемым файлом информационной подсистемы называется файл, используемый в одной или нескольких информационных подсистемах. Файл имеет единственный экземпляр, хранимый на одной из машин системы.

Системным информационным файлом называется файл, используемый несколькими или всеми информационными подсистемами, принадлежащий одной из них и допускающий хранение копий файла на машинах, где файл используется. Ответственность за соответствие копий оригиналу несет система.

Общесистемным файлом называется файл, не принадлежащий ни к одной из информационных подсистем. Используется только системой ORD.

Системными управляющими файлами называются файлы, описывающие состав и структуру документов и файлов, используемых в системе ORD, а также распределение подсистем и их файлов по сети.

Вид имени файла определяет его местоположение в директориях системы и правила его открытия в сетевом варианте системы. Например, собственный файл хранится только на одной машине, на которой расположена информационная подсистема, и всегда открывается в монопольном режиме, если только в SISOA.DBF (распределение подсистем по машинам сети. См. приложение 2) не дано специальное указание открыть его в разделяемом режиме.

Имя собственного файла информационной подсистемы имеет вид:

<имя подсистемы><имя файла в подсистеме>.DBF,

где <имя подсистемы> - один символ, определяющий к какой подсистеме относится файл (см. выше), имена файлов обычно формируются системой автоматически в момент визуального построения модели документа. Например, MCOU.DBF - собственный файл подсистемы "М". Если второй символ в имени собственного файла "_" (знак подчеркивания), то такой файл всегда открывается на текущей машине (той, с которой запущен .EXE файл).

Имя разделяемого файла информационной подсистемы имеет вид:

S<имя подсистемы><имя файла в подсистеме>.DBF,

где <имя подсистемы> - один символ, определяющий к какой подсистеме относится файл. Например, SFOR.DBF (файл "Организации") - разделяемый файл подсистемы "F", STRP.DBF ("Работники предприятия"), ST05.DBF ("Подразделения") - разделяемые файлы подсистемы "Т".

Имя системного информационного файла имеет вид:

U<имя подсистемы><имя файла в подсистеме>;

где <имя подсистемы> - один символ, определяющий имя подсистемы, к которой относится файл, <имя файла в подсистеме> три символа, которые вместе с литерой "U" однозначно идентифицируют файл. Примером такого файла является файл с именем UMCL.dbf, показывающий системе, что файл принадлежит подсистеме "М" и что система обязана следить за копиями этого файла.

Имя общесистемного файла имеет вид:

SS<имя файла>,

где <имя файла> - два символа, которые вместе с литерами "SS" однозначно идентифицируют файл.

4.5.2.3. Список системных управляющих файлов

Системные управляющие файлы имеют следующие имена:

SISOA.DBF	- "Адреса подсистем".
SIORD.DBF	- "Справочник файлов подсистемы".
SIDO.DBF	- "Документ - атрибут".
SIIM.DBF	- "Имена документов подсистемы".
SIAOF.DBF	- "Документ - список файлов".
SIUEK.DBF	- "Экранные образы документов".
SIAGF.DBF	- "Логическая модель док-та".

SU<номер машины>.DBF "Распределение U файлов по машинам".

SIMN.DBF - "Меню подсистемы".

SUUU.DBF- "Список машин системы".

SISF.DBF - "Общий справочник файлов системы".

SIPAT.DBF - "Путь файла".

SIFDBFV.DBF "Контроль доступности файла".

SIQIK.DBF - "Эталоны структур документов".

Последние четыре файла доступны только системе.

В общем случае структура имен файлов указывает на следующее: файлы, имена которых начинаются с "S" или "U" используются не менее чем двумя подсистемами. Если имя файла начинается с "S", то данный файл хранится только в одном экземпляре. Создавать и редактировать такой файл может только подсистема, к которой он относится. Первый символ "U" в имени файла указывает, что данный системный информационный файл имеет эталон, который хранится на какой-то машине системы и множество копий, которые могут храниться на машинах системы, использующих эти файлы. (Любые модификации файла с "U" на любой машине будут перенесены во все копии файлов и в эталон системой автоматически. Используются для уменьшения трафика сети). Распределение U файлов по машинам системы хранится в системных файлах с именами: SU<номер машины>.

"SI" - в имени файла обозначает, что копия структуры файла хранится на каждой машине (СОДЕРЖИМОЕ МОЖЕТ БЫТЬ РАЗЛИЧНО). "SU" - файл, имеющий копию в системе. Хранится на каждой машине (СОДЕРЖИМОЕ не МОЖЕТ БЫТЬ РАЗЛИЧНО!).

4.5.2.4. Имена полей .DBF файлов

Все имена полей .DBF файлов в системе состоят из четырех символов. Имена полей системных управляющих файлов устанавливаются системой.

Имена полей остальных файлов генерируются автоматически, но могут определяться разработчиком.

Например, в информационной подсистеме "Основные средства" имеется файл OMIS.DBF. Файл состоит из трех полей: OMIS (OMIS, OM02, OM03).

В подсистеме "Материальные ценности" имеется файл MVIC.DBF, содержащий пять полей: MVIC (MVIC, OMIS, MV02, MV03, OM02).

Как видно из примера, два поля файла MVIC.DBF: OMIS и OM02 связывают его с файлом OMIS.DBF.

Все поля файлов, соответствующие атрибутам предметной области, с одинаковой семантикой, должны иметь одинаковое физическое имя.

4.5.3. Распределение файлов по директориям

Все *.DBF* и *.NTX* файлы системы хранятся в разделе под именем *ORD*, размещаемом на диске *D:* (или любом другом, указанном при инсталляции). Рекомендуется инсталлировать систему на диск *D:*, что позволяет нестандартно решать вопросы безопасности и конфиденциальности хранимой информации. Раздел *ORD* должен иметься на каждой машине в сетевом варианте системы.

Структура раздела *ORD* определяется количеством и составом информационных подсистем, эксплуатируемых в системе. Общий вид раздела приведен на рисунке 37, где

ORD - раздел системы на данной ЭВМ;

SUB<*x*> - раздел, первые три буквы имени которого (*SUB*) указывают, что речь идет о подсистеме, а последняя буква: <*x*>, - должна совпадать с именем подсистемы (например: *A, B, F, G, O, M, S, T, U, Y*);

SIDBF - раздел для хранения системных управляющих файлов подсистемы <*x*>, имеющих расширение *.DBF* и имена, начинающиеся на *SI* (т.е. файлов, имеющих в каждой подсистеме на каждой ЭВМ сети);

SINTX - раздел для хранения индексных файлов, подключаемых к системным управляющим файлам, хранящимся в разделе *SIDBF*;

S<*x*>*DBF* - раздел для хранения разделяемых информационных файлов подсистемы <*x*>, (начинающихся на *S*<*x*>);

S<*x*>*NTX* - раздел для записи индексных файлов, подключаемых к разделяемым файлам подсистемы <*x*>, хранящимся в разделе *S*<*x*>*DBF*;

DBF - раздел для записи собственных файлов подсистемы <*x*>, имеющих расширение *.DBF* и имена, начинающиеся на <*x*>;

NTX - раздел для записи индексных файлов, подключаемых к собственным файлам подсистемы <*x*>, хранящимся в разделе *DBF*;

PRG - раздел для хранения исходных, объектных и исполняемых файлов подсистемы <*x*> и командных файлов *MS DOS*;

PRG_WIN - раздел для хранения исходных, объектных и исполняемых файлов подсистемы <*x*> и командных файлов *MS Windows*.

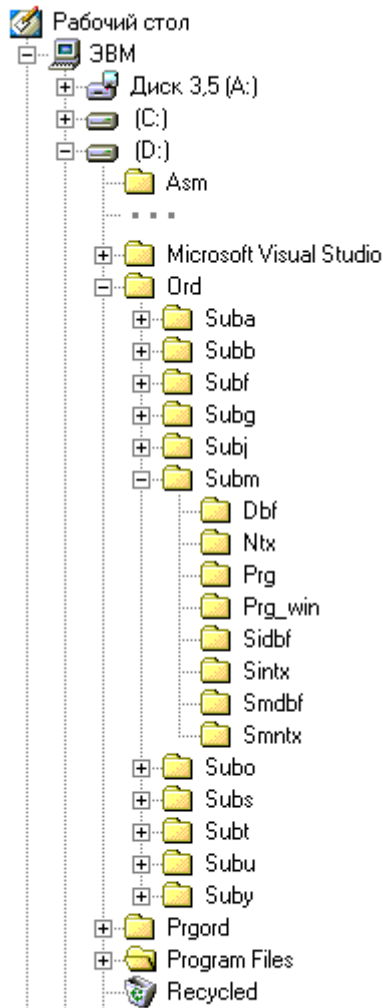


Рис.37. Структура раздела *ORD*

Адрес (номер ЭВМ сети) расположения самой подсистемы *SUB<x>* указывается в файле *SISOA.DBF* (список доступных подсистем).

Принятая структура раздела *ORD*, обуславливает следующие особенности хранения файлов в системе *ORD*:

1) *Системные управляющие файлы* (файлы, имена которых начинаются на *SI...*) имеются в каждой информационной подсистеме, расположенной на каждой ЭВМ сети.

Информация в системных управляющих файлах одноименных подсистем, запускаемых на разных ЭВМ сети, не должна быть противоречивой:

- местоположение (номер ЭВМ) собственных и разделяемых файлов подсистемы, указываемое в файле *SISOA.DBF*, должно быть одним и тем же, в противном случае будет нарушена целостность данных подсистемы;

- структура документов, описываемая управляющими файлами, должна быть идентичной;

- допускается (в целях экономии места на диске и в памяти) на отдельных машинах формировать сокращенную копию подсистемы,

включающую в себя только часть документов подсистемы;

2) *Собственные файлы* любой подсистемы и их индексные файлы должны иметь в системе только одну рабочую копию, расположенную на той машине, номер которой указан в *SISOA.DBF* всех рабочих копий системных управляющих файлов данной подсистемы. Исключение составляют собственные файлы, второй символ в именах которых является символом подчеркивания ("_"): такие файлы располагаются на каждой машине, имеющей рабочую копию данной подсистемы;

3) *Разделяемые файлы* любой подсистемы и их индексные файлы, также как и собственные файлы, должны иметь в системе только одну рабочую копию,

расположенную на той машине, номер которой указан в *SISOA.DBF* всех рабочих копий системных управляющих файлов данной подсистемы;

Определение. ЭВМ, на которой располагается рабочая копия собственных и разделяемых файлов подсистемы $\langle x \rangle$, называется хост-ЭВМ этой подсистемы.

- 4) *Системные информационные файлы* (файлы, начинающиеся с "U") располагаются на каждой машине, на которой имеется рабочая копия хотя бы одной подсистемы, использующей эти файлы. Каждый системный информационный файл имеет эталон, который хранится на одной из машин системы.

Любые модификации системного информационного файла на любой машине будут автоматически перенесены системой во все копии файлов и в эталон. Это осуществляется с целью уменьшения трафика сети.

Описание распределения системных информационных файлов по машинам системы хранится в системных файлах с именами *SU* \langle номер машины \rangle и не должно быть противоречивым.

5. ТЕСТИРОВАНИЕ

Этап тестирования может повлечь затраты, которые составят половину общих расходов на создание системы. В процессе тестирования используются данные характерные для системы в рабочем состоянии, т.е. данные для тестирования нельзя выбирать случайным образом.

В процессе тестирования используются различные критерии. Наиболее важными из которых являются следующие: каждая ветвь программы должна быть опробована и программа должна выдать правильный результат; - каждый путь в программе должен быть испытан и программа должна выдать правильный результат; - для каждой спецификации программы необходимо располагать набором тестовых данных позволяющих установить правильность работы программы. При организации данных для тестирования полезно руководствоваться следующими положениями:

- в начале используются простые тесты затем более сложные, т.к. для проверки сложных программ недостаточно одного или двух тестов;

- следует учитывать логику операторов, например, для оператора DO WHILE $X < Y$... ENDDO выбирается значение $X < Y$, $X = Y$ и $X > Y$;

- система предоставляет для анализа ошибочных ситуаций специальный трассировочный файл с именем ORDTRACE.log, который находится в папке, откуда происходит запуск исполняемого модуля;

- результат работы теста должен быть известен до начала его выполнения. Так как появление ошибок в больших программах неизбежно, то программист должен их найти. Поиск ошибок упрощается, если следовать простым правилам - именам переменных желательно давать смысловое содержание - не следует в программе использовать без нужды константы, вместо них используются идентификаторы с начальными значениями (например, для этого существует специальный файл-раздел констант Ordch.ch) .

Приложение 1

А. Системное управление атрибутами документа

В разделе 4 изложены основы описания экранного представления документа в системе *ORD*.

Из содержимого этого раздела можно видеть, что большая часть полей файла *SIUEK*, используемого для этой цели, определяла имена атрибутов документа, имена и характеристики полей файлов, хранящих их значения, расположение элементов документа на экране, порядок их представления в окне поиска экземпляра документа и др.

Одним из наиболее значимых для системы полей в строке описания экранного образа документа, является поле *SI70* (на экране носит обозначение Упр.), хранящее признак управления. Как указывалось, эти признаки определяют характер элемента экранного образа описываемого, строкой файла *SIUEK*. Причем значения признака *SI70* не только выделяют обычные атрибуты документа, но и определяют структурообразующие элементы и, что играет большую роль в работе системы, задают специфические особенности обработки различных элементов документа.

Признак управления, может принимать одно из следующих значений: **?**, **A**, **D**, **E**, **G**, **H**, **h**, **J**, **K**, **k**, **L**, **I**, **M**, **N**, **P**, **Q**, **q**, **R**, **r**, **S**, **s**, **T**, **t**, **U**, **W**, **X**, **Y**, **Z**, **z**.

Пример строки (записи) *SIUEK*, приведен в таблице П.1.

Таблица П.1. Содержимое строки (записи файла) *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
<i>01lr</i>	<i>UG900001</i>	<i>lgr+</i>	<i>R</i>	5	2			10	78	0		0	0

Назначение признаков поясняется ниже, в порядке функциональных особенностей, отличном от алфавитного порядка.

А.1 Оформление документа

Z - атрибут, не имеющий поля в файле (например: заголовок документа, части);

R, r – очистка экрана, рисование рамок с координатами из полей *SI66*, *SI67*, *SI68*, *SI69*. Какое именно действие будет выполнено, определяет следующее правило по значению первого символа в поле *SI65*:

пробел – очистка прямоугольной области экрана с координатами *SI66*, *SI67*, *SI68*, *SI69*;

'1' – рамка без чистки с координатами *SI66*, *SI67*, *SI68*, *SI69*;

'2' – двойная рамка с координатами *SI66*, *SI67*, *SI68*, *SI69*;

'3' – рамка + очистка с координатами *SI66*, *SI67*, *SI68*, *SI69*;

'4' – двойная рамка + очистка с координатами *SI66*, *SI67*, *SI68*, *SI69*;

Второй, третий и четвертый символы поля *SI65* определяют цвет рамки (по известным правилам определения цветов в 16-цветной гамме: три бита RGB и бит интенсивности).

Например, в случае *SI70*='R' и *SI65*="lgr" будет выведена одинарная рамка желтого цвета, при этом информация в границах рамки не будет стерта с экрана.

В примере, приведенном выше в таблице А.1, строка *SIUEK* определяет прямоугольную рамку желтого цвета, верхний левый угол которой расположен во втором столбце пятой строки, а нижний правый угол – в 78-м столбце 10-ой строки.

Первый символ "R" или "r" в части, определяет не только чистку экрана (и/или рамку), но и указывает системе ту часть экрана, которую она должна воспринимать как

единое целое. Т.е., если рамка вышестоящей части охватывает нижестоящие части, то нижестоящие части будут наследовать свойства такой части.

Если часть, помимо плоского изображения, может иметь и табличное (оконное), а координаты окна части отличны от координат (рамки) плоского изображения, то в данной записи *SIUEK* (первого "R"/"r" части) в позициях 18÷25 поля *SI65* указываются координаты окна. При этом в позициях 14 и 15 можно указать системе порядок открытия (позиция 14) и закрытия (позиция 15) подчиненных частей документа в табличном представлении. Наличие символа "A" в позиции 14 и 15 указывает системе, что соответствующую часть следует открывать в табличном виде и закрывать таблицу при открывании и закрытии табличной формы вышестоящей части.

Пример описания такой ситуации приведен в строке *SIUEK*, изображенной в таблице А.2.

Таблица А.2. Строка *SIUEK.DBF*, управляющая табличной формой части

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
01xC	UG900002	1gr+ AA 06031178	R	5	2			10	78	0		0	0

В случае, показанном в таблице А.2 при открывании части 2 в табличной форме все ее подчиненные части должны быть открыты также в табличной форме (символ 'A' в 14-ой позиции поля *SI65*). Символ 'A' в 15-ой позиции приказывает системе закрывать подчиненные части при сворачивании таблицы части 2 в плоское отображение. Комбинация символов '06031178' в позиции 18-25 указывает координаты вывода таблицы:

06 - верхняя строка вывода таблицы; 03 - левый столбец границы окна; 11 - нижняя строка вывода таблицы; 78 - правый столбец вывода окна.

Этот пример также показывает, что координаты рамки и таблицы могут не совпадать. Стиль отображения таблиц рассмотрен в параграфе в *Help* системы.

Внимание! Для частей со статусом {VvE} наличие рамки обязательно.

Для части типа 'H' наличие рамки обязательно, если она не входит в более 'крупную семантическую' часть док-та.

Если часть типа 'H' не содержит символов "R"/"r" в своем описании, и ее непосредственная вершина тоже имеет тип 'H', то система воспринимает такую конструкцию, как один объект. Данный факт также должен быть зафиксирован в поле *SI51* файла *SIAGF* для данной части документа.

При выводе рамки части типа "V" на экране по умолчанию будет выводиться информация о количестве (составе) слоев многослойной части (см. рис. 8).

Понятие "слой части" непосредственно связано с системными массивами с именами *KP[]*, *KPP[]*, *NPI[]*.

Если перед строкой рамки в *SIUEK*, присутствует строка с символом управления "z", то на экран будет выводиться информация, указанная в поле *SI65*. Пример таких записей показан в таблице А.3.

Таблица А.3. Строки *SIUEK.DBF* для вывода информации на рамку

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
01Ik	UG900001	Номер Итого	z	5	2			10	78	0		0	0
01Ir	UG900001	1gr+	R	5	2			10	78	0		0	0

А.2. Представление простых полей документа

G – атрибут, имеющий поле в файле, которое разрешено редактировать;

S – то же самое, что и признак G, но поле, указанное в *SI26* не редактируется;

s – то же самое, что и признак S, но значение поля всегда высвечивается на экране, где бы ни находилась часть, которой принадлежит данное поле;

X – значение атрибута и поля не выводится на экран, но значение поля заносится в файл, если документ модифицировался;

? – значение атрибута и поля не выводится на экран, но значение поля читается из файла и хранится в массиве части.

При записи документа в файл, значение такого поля не записывается;

А.3. Представление полей классификаторов К- и Q-типа

К – атрибут, значение которого хранится в простом классификаторе.

В поле *SI26* указывается имя файла простого классификатора, а в 25-м разряде поля *SI65* – длина якоря классификатора. (Длина поля файла, хранящего якорь классификатора и длина поля якоря в классификаторе должны совпадать.)

Значения полей *SI56*, *SI57*, *SI76* характеризуют поле "прокрутки" простого классификатора.

Пример записи о простом классификаторе в файле *SIUEK* представлен в таблице А.4.

Таблица А.4. Описание простого классификатора в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость	
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>	
9093	UF110001	Город	2	K	10	51	SFBN	sf 03	10	58	3	C	15	0

Наличие строки со значением управляющего поля "K" подразумевает наличие в системе файла с именем *SF03*. Файл *SF03* должен иметь якорное поле *SF03* и служебное поле с именем *NNN2*. При этом тип служебного поля *NNN2* символьный, длина поля *NNN2* – 15 символов.

Длина поля якоря *SF03* равна двум байтам (обязательно указывается в последнем разряде поля *SI65*, содержащем имя классификатора).

Правила работы с классификатором могут регламентироваться кодом разрешенных с ним действий, который указывается в строке вызова классификатора, как показано в таблице А.5, или системной функцией *DOPU* (см. [6], параграф 2.3.1.).

Таблица А.5. Описание разрешенных для простого классификатора действий в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость	
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>	
9093	UF110001	Город	N*2	K	10	51	SFBN	sf 03	10	58	3	C	15	0

Символ, стоящий перед "*", может принимать значения:

"A" – допускаются любые модификации данных;

"E" – допускается только редактирование данных;

"N" – модификация данных запрещена.

Если элемент управления отсутствует, то по умолчанию работает правило "A". При этом система учитывает 'природу' классификатора по его имени. Кроме указанных ограничений для работы с K классификаторами, возможно использование правил *DOPU* и *DIAP* (см. Help системы).

Если возникает необходимость использовать в одной и той же части один и тот же классификатор, то данный факт отображается в *SIUEK* так, как показано в таблице А.6:

Таблица А.6. Описание повторного использования простого классификатора в *SIUEK*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
-------	----------	------------------------	-----	------------	------------	------	------	-----------	-----------	-----------	-----	-----------	--------------

SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9093	UF110001	Город N*2	K	10	51	SFBN	sf 03	10	58	3	C	15	0
9094	UF110001	Город PN*2	K	10	51	SFBN	sf 0X	10	58	3	C	15	0
9095	UF110001	ORD SFOX SF03	X	0	0			0	0	0		0	0

В строке содержащей вызов классификатора (9094) указывается в 22 позиции служебный символ "P", а в следующей строке с символом управления "X" указывается фактическое имя классификатора SF03.

Имя, эквивалентное имени SF03, система сгенерирует автоматически, если подобный классификатор формируется в визуальном режиме. В рассматриваемом примере таким именем является SFOX.

По своей природе K классификатор это список возможных значений данного свойства в данном объекте.

K – то же самое, что и признак K, но поле, указанное в SI26, редактировать запрещено;

Q – атрибут, значение которого хранится в сложном классификаторе.

В поле SI26 указывается имя файла сложного классификатора, а значения полей SI56, SI57, SI76 характеризуют поле файла, имя которого записано в поле SI20, хранящего значение якоря из файла сложного классификатора.

Внимание! Следующая за Q служебная запись SIUEK должна содержать в поле SI65 информацию: <имя образа Q><пробел><имя файла Q><пробел><имя поля "прокрутки" из Q>< служебные символы> и иметь символ управления X

Например, "UY327030 SYMY SY03"

В таблице А.7 приведен пример вызова Q-классификатора и описания его образа в файле SIUEK..

Таблица А.7. Описание вызова Q-классификатора в файле SIUEK.DBF

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поис ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327003	Ед.изм.	Q	11	58	SYRM	SYMY	10	78	3	C	2	0
31G9	UY327003	UY327030 SYMY SY03 ZY03	X	0	0			0	0	0		0	0
31GD	UY327002	Следующая часть	X	0	0	YRSV		0	0	0		0	0
31GH	UY327002	Расход в производство	G	14	22	YRSV	YR16	14	47	4	N	11	2
...
31GQ	UY327030	Образ Q	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GU	UY327030	Ед.изм. IND	S	0	0	SYMY	SY03	0	0	9	C	3	0

Строка SIUEK с номером 31G5 содержит вызов Q-классификатора.

Строка SIUEK с номером 31G9 (служебная) содержит правила вызова Q-классификатора, т.е., имя экранного образа классификатора, имя файла классификатора и имя поля прокрутки. В поле управления (SI70) должен находиться символ X или x.

Строка SIUEK с номером 31GQ содержит описание образа Q классификатора.

Порядок следования атрибутов в образе классификатора, при его отображении на экране указывается в поле "Порядок", при формировании образа Q-классификатора (при помощи утилиты DB.EXE).

Если в поле управления (SI70) служебной записи для Q, указать символ "x", то при использовании Q-классификатора можно задавать различные типы отношений, в которых участвует часть типа Q (см. [6]).

Сам образ Q-классификатора должен входить в описание (типа) документа в SIUEK и в SIAGF. В образе Q поля SI72 (поиск) для всех информационных полей должны содержать 9.

Внимание! Поле "прокрутки", указанное в точке вызова Q-классификатора, дублируется (под другим именем) в вызывающем файле. Имя такого поля система генерирует автоматически, и оно начинается с символа 'Z'.

Если возникает необходимость, чтобы содержимое файла классификатора показывалось в алфавитном порядке, то в образе Q-классификатора указывается служебное слово 'IND'. Служебное слово 'IND' может не принадлежать полю 'Прокрутки'. Поле со служебным словом 'IND' должно иметь индекс.

Для рассматриваемого примера в поле *SI26* записи файла *SIORD*, описывающей поле *SYMY* файла *SYMY.DBF*, его имя должно начинаться с маленькой буквы: *sY03*.

По своей природе Q-классификатор – это список возможных значений данного свойства в описываемом объекте, соответствующий объекту Q. Поэтому, в отличие от K-классификатора, Q-классификатор должен быть какой-то частью другого документа или самостоятельным документом.

Некоторые особенности использования Q-классификаторов приведены в параграфе 2.3.1 -"Особенности работы с Q-классификатором" в [6].

q – то же самое, что и признак Q, но поле, указанное в *SI26*, редактировать запрещено.

А.4. Представление модельных (Н) классификаторов

Н – подключение внутри данного экранного образа модельного классификатора.

В поле *SI65* указывается служебная информация. Например, "Z UO994004 111", где Z – служебный символ, UO994004 – имя экранного образа модельного классификатора, 111 – код работы с классификатором, три разряда которого (после имени экрана модельного классификатора и пробела), имеют следующее значение:

- 1-я позиция: 1 - элементы модельного классификатора можно редактировать,
0 - элементы модельного классификатора редактировать запрещено;
- 2-я позиция: 1 - записи классификатора разрешено прокручивать,
0 - записи классификатора запрещено прокручивать;
- 3-я позиция: 1 - в модельный классификатор разрешено добавлять записи
0 - в модельный классификатор запрещено добавлять записи
2 - в модельном классификаторе разрешено редактировать только добавленные записи.

По умолчанию все значения этого кода принимаются равными нулю.

Пример описания в *SIUEK* использования Н-связи представлен в таблице А.9.

Таблица А.9. Описание вызова Н-классификатора в файле *SIUEK.DBF*

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY327002	Ед.изм.	G	9	58	SYRM	SYMC	9	66	0	C	2	0
31G9	UY327002	Z UY327003 010	H	0	0	SYRM	SYMY	0	0	0		0	0
31GA	UY327002	Состав изделия	S	0	0	SYRM	SY04	0	0	0		0	0
...
31GQ	UY327003	Расход в производство	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GX	UY327003	1gr+	R	10	2	SYMY	SYMY	12	78	0		0	0
31GU	UY327003	Ед.изм.	S	11	33	SYMY	SY03	11	50	0	C	3	0

В этом примере строка с адресом 31G9 содержит вызов модельного классификатора с именем экранного образа UY327003, описание которого начинается со строки 31GQ. Символ 'Z' является служебным и обозначает обязательное наличие в части Н рамки.

Код работы с классификатором "010", означает разрешение только выбора значения экземпляра объекта из 'Н', без права его модификации. (Обычно правила работы с 'Н' классификатором определяются из режима визуального создания модели документа).

В поле *SI65* первой строки образа классификатора в *SIUEK* (копия которого для текущего документа хранится в массиве *SARY*), содержится ссылка (*31G9) на точку вызова 'H' классификатора.

Если служебный символ 'z' – то в части 'H' рамка отсутствует. Это означает, что часть 'H' в качестве составной (включаемой) входит в другую часть 'H'. При этом, координаты рамки вышестоящей H должны “охватывать” информационные поля нижестоящих H. Пример описания такой ситуации приведен в таблице А.10.

Таблица А.10. Описание вложенных H-классификаторов в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поис ск	Тип	Дли на	Гочн ость	Н- поиск	
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>		<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>	<i>SI73</i>
3101	UA00S002	Приход	M	0	0	AA0J	aa0J	0	0	0	C	3	0	0	
3102	UA00S002	*.... ВСЯ	S	0	0	AA0J	aa04	0	0	0	C	3	0	0	
3109	UA00S002	Z UA00S003 0100 AA02*F	H	0	0	AA0J	AA0E	0	0	0	C	3	0	0	
311A	UA00S003	*3109 Накладная прихода	M	0	0	AA0E	aa0E	12	78	0		0	0	0	
311B	UA00S003	1gr+	R	5	3			9	75	0		0	0	0	
311C	UA00S003	Кол-во рулонов	S	6	4	AA0E	AA2N	6	20	6	N	7	2	3	
311E	UA00S003	Цена за метр	G	6	4	AA0E	AA2N	6	44	99	N	10	2	4	
311H	UA00S003	z UA00S004 000	H	0	0	AA0E	AA07	0	0	0	C	3	0	0	
3130	UA00S004	*311H Дата поставки	M	0	0	AA07	AA07	0	0	0	C	3	0	0	
3131	UA00S004	N накладной	G	8	4	AA07	AA0P	8	12	5	D	8	0	1	
3133	UA00S004	Поставщик	Q	8	34	AA07	SFOR	8	39	99	C	3	0	2	
3135	UA00S004	UA00S032 SFOR SF08 ZF08	X	0	0			0	0	0		0	0	0	

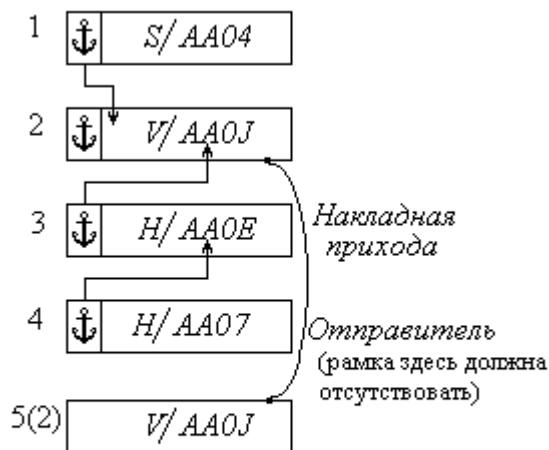


Рис. А.1. Граф структуры документа со связанными H-классификаторами

В этом случае в поле *SI73* (H-Поиск) надо указать порядок показа атрибутов всех связанных H., так как часть H с экранном образом *UA00S003* включает и экранный образ *UA00S004* другого H. Фрагмент графа такого документа приведен на рисунке А.1.

В *SIAGF* такого документа в поле *SI51* (Название части) для части 3 указывается:

Накладная прихода #01 \$,

где:

- Накладная прихода – имя части
 - # - служебный символ (23-я позиция)
 - 01 - количество (указанное в символьном виде) подчиненных классификаторов, которые должны восприниматься, как одно целое.
 - \$ – служебный символ (33-я позиция)
- Особенности работы с частью H рассмотрены в [6], основные моменты приведены ниже.

Слой H-части всегда однозначно соответствует слою вышестоящей части (части, номер которой указан в поле *SC63* файла *SIAGF.DBF*).

Слой H-части всегда является экземпляром какого-то типа документа. Этот тип документа должен быть описан в системе.

Использование части типа "H" в документе, позволяет установить и поддерживать всевозможные связи (отношения) между документами системы (см. HELP "Правила" вызываемый при модификации образа части типа "H").

Система позволяет поддерживать отношения 1:1 (один к одному), 1:m (один ко многим), n:m (многие ко многим) между исходным документом и документом типа "Н", учитывая семантику таких связей. (Например, правила передача свойств объекта типа "Н" объекту, с которым он связывается, или, наоборот, наследование им свойств объекта, с которым он связан.)

Это определяют:

- тип установленной структурной связи между "Н" и вышестоящей частью;
- код модельной связи (код работы с классификатором "Н") и;
- код самой структурной связи (символы, указанные в поле SC50 файла SIAGF для части "Н").

При занесении в файл Н-части якоря вышестоящей части между этими частями образуется, так называемое, "кольцо".

Если между модельным классификатором и вышестоящей частью образовано кольцо по якорям (т.е., слой (запись) модельного классификатора не может использоваться в другом документе, если данный слой уже был задействован в каком-либо процессе), то работают правила, поддерживающие связь 1:1. В этом случае в файле модельной части, должно быть поле типа N длиной 1, имя которого сформировано по правилу: "Н" + right(TDOK,3).

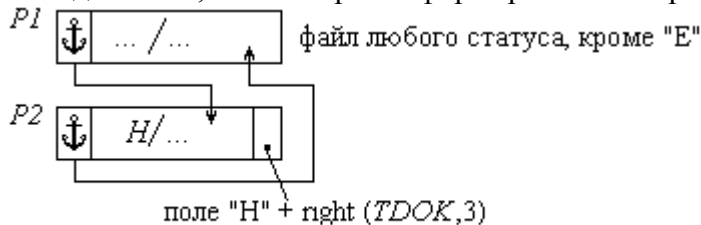


Рис. А2. Н-классификатор с "кольцом"

Например, если между частями P1 и P2 задано "кольцо", то эта связь будет выглядеть так, как показано на рисунке А2. Признак наличия кольца для "Н" части фиксируется в системном управляющем файле SIAGF в поле S150.

Если кольца нет, то для экземпляров документа типа "Н" допускается связь 1:1, 1:m, n:m с передачей права использовать один и тот же экземпляр "Н" с различными слоями документа (в котором "Н" задействован).

Если семантика работы с частью "Н", допускает добавление нового слоя (и, следовательно, нового экземпляра документа "Н" в системе), то для таких документов в файле SIAGF в последнем 33-м разряде поля S151 записи, соответствующей Н-части, должен быть проставлен символ '\$'.

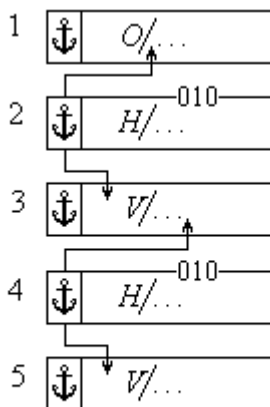


Рис. А3. Фрагмент структуры документа

Для записей, добавленных в Н-классификатор, система сформирует маршрут документа Н (см. [6], параграф 2.1.3. Маршрут формы (документа)).

Если часть типа "Н", имеет подчиненные части (+30 в SC63 SIAGF.DBF), и все подчиненные части должны наследовать поведение части "Н", то для подчиненных частей необходимо указать признак наследования.

Пусть, например, имеется фрагмент структуры документа, показанный на рис. А.3:

Часть 2 является вершиной для частей 3,4,5. Если для предметной области необходимо, чтобы после выбора значения из Н система автоматически заполнила части

3,4,5, и поведение этих частей наследовало бы поведение части 2, то в первых строках SIUEK таких частей (символ управления "М") указывается семантический код управления данной части. Длина этого кода 17 байт (см. пример документа 0315 подсистемы 'Y' в полной версии системы). Перед ним находится признак "W", указывающей системе, что данная часть наследует свойства вышестоящей.

h – могут быть использованы все свойства "Н", но с запретом на использование одного и того же слоя Н в данном экземпляре документа.

Поэтому в образе такого "Н" должно быть поле внесенного якоря первого файла из первой части документа. Копирование подобного якоря должен находиться под контролем символа управления "N" - см. "Структурная связь для подчиненного объекта".

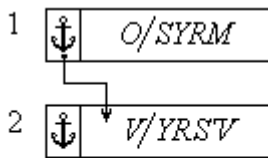
А.5. Связь частей документа

М - признак того, что в поле *SI20* данной записи таблицы *SIUEK* хранится имя одного из файлов настоящего документа, а в поле *SI26* – имя поля собственного якоря этого файла.

Если при этом первым символом поля *SI65* является "*", то следующие за "*" четыре символа поля *SI65* задают адрес для копирования данных. Т.е. копия информации из поля, указанного в поле *SI26* данной записи *SIUEK*, будет занесена в поле, имя которого указано в поле *SI26* записи *SIUEK*, задаваемой адресом копирования. Это действие выполняется в режимах *UREG* = 2 (создание) или *UREG* = 4 (редактирование).

Символ "М" в *SI70* указывает на собственный якорь файла, который в режимах *UREG* = 2 или *UREG* = 4 будет сформирован при появлении новой записи (нового документа или слоя) в части. Длина поля с символом управления "М" определена в байтах. По умолчанию длина якорного поля составляет 3 байта, что допускает наличие в системе немногим менее 2 в 24-й степени различных экземпляров данного объекта. Если эта величина не соответствует конкретным требованиям, следует указать необходимую длину якоря.

Пусть документ ("0377") имеет структуру, показанную на рисунке А4.



Понятиями предметной области, соответствующими графу, представленному на этом рисунке, могут быть, например, *Фирма* и *Отдел*.

Т.е., структура документа предполагает, что объект *Фирма* (*SYRM*) может иметь несколько *Отделов* (*YRSV*). Эту структура можно отобразить в *SIUEK* документа так, как показано в таблице А.11.

Рис. А4. Пример структуры документа

Таблица А.11. Описание документа "0377" в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY377001	Фирма	M	0	0	SYRM	SYRM	0	0	0	C	2	0
31G9	UY377001	Наименование фирмы	G	5	7	SYRM	YY_1	5	17	1	C	30	0
...
31GC	UY377002	Отдел	M	0	0	YRSV	YRSV	0	0	0	C	2	0
31GD	UY377002	*31G5 Связка с фирмой	N	0	0	YRSV	SYRM	0	0	0	C	2	0
31GH	UY377002	Наименование отдела	G	9	22	YRSV	YR16	9	47	9	C	20	0

В таблице А.11 строки 31G5 и 31GC имеют символ управления "М". Это означает, что строка 31G5 определяет собственный якорь объекта *Фирма* (*SYRM*), а строка 31GC – собственный якорь объекта *Отдел* (*YRSV*).

Строка 31GD указывает на связь объекта *Отдел* (*YRSV*) с объектом *Фирма* (*SYRM*).

Модель, представленная на рисунке Рис. А.4., по умолчанию поддерживает только структурную связь, если же между объектами *Фирма* и *Отдел* необходимо отслеживать и очередность связи, то описание структуры примет вид таблицы А.12.

Таблица А.12. Описание связи по очередности в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY377001	Фирма	M	0	0	SYRM	SYRM	0	0	0	C	2	0
31G9	UY377001	Наименование фирмы	G	5	7	SYRM	YY_1	5	17	1	C	30	0

31GA	UY377001	Связь по очередности	X	0	0	SYRM	YRSV	0	0	0	C	2	0
...
31GC	UY377002	*31GA Отдел	M	0	0	YRSV	YRSV	0	0	0	C	2	0
31GD	UY377002	*31G5 Связка с фирмой	N	0	0	YRSV	SYRM	0	0	0	C	2	0
31GH	UY377002	Наименование отдела	G	9	22	YRSV	YR16	9	47	9	C	20	0

Появление строки 31GA в *SIUEK.DBF* и ссылки *31GA в строке 31GC предписывает системе соблюдать правило: последний экземпляр объекта *Отдел*, физически появляющийся в этой связи, будет логически первым, т.е. логически первым всегда будет новый “отдел”.

Если же принципиально важным для данного фрагмента предметной области является уточнение параметра Времени установления подобной связи (см., Help системы), то описание такой ситуации в *SIUEK* приобретет вид, показанный в таблице А.13.

Таблица А.13. Описание связи по времени в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY377001	Фирма	M	0	0	SYRM	SYRM	0	0	0	C	2	0
31G9	UY377001	Наименование фирмы	G	5	7	SYRM	YY_1	5	17	1	C	30	0
31GA	UY377001	TIME-SYRM+YR17 время	X	0	0	SYRM	YRSV	0	0	0	C	2	0
...
...
31GC	UY377002	*31GA Отдел	M	0	0	YRSV	YRSV	0	0	0	C	2	0
31GD	UY377002	*31G5 Связка с фирмой	N	0	0	YRSV	SYRM	0	0	0	C	2	0
31GH	UY377002	Наименование отдела	G	9	22	YRSV	YR16	9	47	2	C	20	0
31GK	UY377002	Дата образования отдела	G	11	22	YRSV	YR17	11	47	3	D	8	0

Строка 31GA содержит команду *TIME* с параметром *SYRM+YR17*, которая предписывает системе устанавливать логическую связь между объектами *Фирма* и *Отдел* с учетом значений полей *SYRM* и *YR17* объекта *Отдел*. При этом система учитывает тип связывающего поля (в данном примере тип поля *YR17* – 'дата').

В отличие от таблицы А.12, правило, зафиксированное на рисунке А.13, в общем случае, позволяет учитывать любое функциональное поведение, которое необходимо устанавливать на момент актуализации связи между объектами *SYRM* и *YRSV*. Если имя этого правила начинается со служебного слова *TIME*, то система будет отслеживать связь между объектами *Фирма* и *Отдел* по самой поздней дате, имеющейся у объекта *Отдел*, независимо от того, когда физически информация об этом объекте появляется в базе данных.

Физически система создаст составной индекс по полям *SYRM+YR17* файла *YRSV.DBF* и будет представлять информацию в логическом порядке, установленном этим индексом. Имя составного индекса система сгенерирует автоматически и пропишет его в поле “Имя атрибута на экране” после создания эталона документа, как показано в таблице А.14.

Таблица А.14. Описание связи по времени в файле *SIUEK.DBF* после создания эталона

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31GA	UY377001	TIME-SYRM+YR17 ZZ01	X	0	0	SYRM	YRSV	0	0	0	C	2	0

В строке 31GA после создания эталона содержится имя составного индекса “ZZ01” по полям *SYRM+YR17* файла *YRSV.DBF*.

Количество полей в составном индексе не имеет ограничения.

Внимание! Если в документе, используется правило *М (копирование из многослойной части) и специфицируется передача в географически удаленную локальную сеть, то уточнение правила копирования обязательно (см. Help системы "Подготовка информации для передачи в другую сеть").

Если для такого документа указана команда *SIME* (вместо *TIME*), то локальная система реализует связь по времени, а удаленная система будет реализовывать связь по очередности (как в табл. А.12).

Правило “*М” может применяться и в наследовании. Например, пусть модель

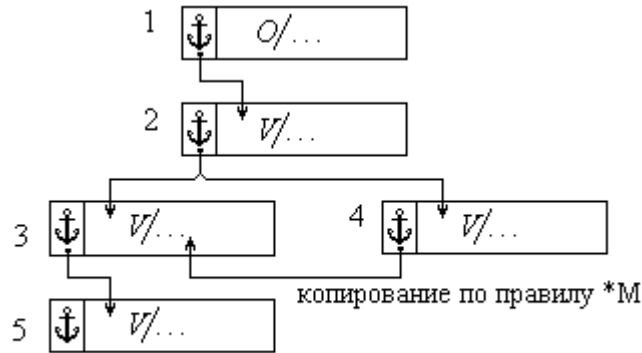


Рис А5. Структура документа со связью *М

документа имеет вид, показанный на рисунке А5. В этом документе выполнено копирование в часть 3 собственного якоря части 4. Такой случай, т.е. двойная связь части 3 с частью 2: напрямую и через часть 4, для системы означает, что все слои части 4 соответствуют каждому слою части 3.

Наличие части 5 позволяет выполнить специфическую функцию пополнения/фильтрации слоев части 4, если обе эти части, используют один и тот же файл в качестве основного. В этом случае, записи, перечисленные в слоях части 5, починенных конкретному слою части 3, физически попадают в тот же файл, что и записи части 4. Но при этом такие записи исключаются из числа записей части 4, являющихся общими для всех слоев части 3. С другой стороны, они как бы пополняют список записей, относящихся к конкретному слою части 3, так как природа свойств, описываемых частью 4 и частью 5, одинакова.

N – Структурная связь для подчиненного объекта. Признак того, что система должна внести в файл F_1 , имя которого указано в поле *SI20*, собственный якорь файла F_2 , для которого адрес M-строки *SIUEK* задан символами со 2-го по 5-й (адресом копирования) поля *SI65* данной записи *SIUEK*. Поле *SI65* (Имя атрибута) N-строки всегда начинается с символа "*" и содержит адрес копирования.

Таблица А.15. Описание N-связи в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY377001	Фирма	M	0	0	SYRM	SYRM	0	0	0	C	2	0
31G6	UY377001	Кредитная история	M	0	0	SYDM	SYDM	0	0	0	C	2	0
31G7	UY377001	*31G5 Фирма	N	0	0	SYDM	SYRM	0	0	0	C	2	0
31G9	UY377001	Наименование фирмы	G	5	7	SYRM	YY_1	5	17	1	C	30	0
...
...
31GC	UY377002	*31GA Отдел	M	0	0	YRSV	YRSV	0	0	0	C	2	0
31GD	UY377002	*31G5 Связка с фирмой	N	0	0	YRSV	SYRM	0	0	0	C	2	0
31GE	UY377002	*31G6 Связка с кред.ист.	N	0	0	YRSV	SYDM	0	0	0	C	2	0
31GH	UY377002	Наименование отдела	G	9	22	YRSV	YR16	9	47	9	C	20	0

Строки 31G7,31GD,31GE определяют структурную связь для подчиненных объектов. Имена полей, по которым осуществляется связь, должны совпадать. Например, строка 31GE указывает на необходимость копирования собственного якоря файла *SYDM.DBF* из его поля *SYDM* (строка 31G6) в поле *SYDM* файла *YRSV.DBF*.

Внимание! Недопустимо использовать символы управления {M, N} не для якорных полей. Значения якорных полей недоступны пользователю, но доступны программисту, что снимает многие проблемы по поддержанию адекватности модели и предметной области.

А.6. Копирование полей документа

F - Признак "F" используется для копирования значения одного поля документа в другое и указания правил и ограничений для этой операции. Обычно правила копирования записываются или уточняются после того, как эталон документа был создан под контроль системы впервые.

Копировать информацию можно из любых частей (кроме части типа "E") в любые части (кроме части типа "E"):

- из вышестоящей части в графе документа в нижестоящую;
- из нижестоящей части в графе документа в вышестоящую;
- внутри одной части.

В строке, содержащей указание на копирование F, записывается также модификатор – служебное слово, определяющее порядок копирования. В качестве таких слов обычно используются: "INV" либо "OUT", указывающие, что информация копируется в часть ("INV"), либо из части ("OUT").

В поле атрибута (SI65) записывается команда копирования в следующем формате:

<*> <адрес_копирования> <пробел> <параметр> <пробел> <модификатор> <пробел | тип_классификатора> <пробел | "?"> [<ограничитель>],

например,

*!!!! POLE INV NEWS, где

"*" – служебный символ (первая позиция поля),

!!!! – адрес копирования (номер строки SIUEK) – четыре символа, начиная со второй позиции, указывающий источник или приемник копируемой информации

POLE – параметр: имя поля (источника или приемника информации) или одно из зарезервированных слов: TDOK, MDOK, TIME.

Параметр должен начинаться с 7-ой позиции поля атрибута SI65;

INV – модификатор: одно из служебных слов, INV, OUT, COP, определяющих правило копирования.

Модификатор должен начинаться с 12-ой позиции поля атрибута SI65;

NEWS – необязательный ограничитель: одно из служебных слов, NEWS, NEWR, NEWI, описывающий ограничения копирования.

Ограничитель (если он присутствует) должен начинаться с 18-ой позиции поля атрибута SI65.

Если копирование выполняется из поля классификатора типа K или Q, то в 16-й позиции проставляется символ используемого классификатора, в противном случае, в этой позиции ставится пробел.

При необходимости оповещения пользователя о действии ограничителя, в 17-позиции проставляется знак вопроса, в противном случае, в ней оставляется пробел.

Синтаксис команды копирования контролируется утилитой DB.EXE. При этом различаются следующие варианты копирования:

1 - Отложенное копирование простого поля;

2 - Отложенное копирование из Q;

3 - Прямое копирование из Q;

4 - Прямое копирование простого поля из (в) H;

5 - Прямое копирование значения поля K или Q-классификатора из (в) H;

6 - Прямое копирование из (в) H с ограничением;

- ограничение NEWS; - ограничение NEWR; - ограничение NEWI;

7 - Согласованное копирование (COP);

8- Специальные случаи копирования по F.

Примечание. Все отложенные варианты копирования могут иметь признак обязательного или не обязательного наследования.

Все виды копирования могут иметь признак временного согласования.

Любое копирование может осуществляться пользовательской функцией, однако, семантика копирования в этом случае остается вне контроля системы.

Отложенное копирование простого поля

Этот вид копирования используется для дублирования (повторения) информации.

Пример такой операции приведен в строке 31GL таблицы А.16.

Таблица А.16. Описание отложенного копирования простого поля в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327001	Проводка	M	0	0	SYRM	SYRM	0	0	0	C	2	0
31G9	UY327001	Дата проводки	G	5	7	SYRM	YY_1	5	17	1	D	8	0
...
31GD	UY327002	Служебн.	X	0	0	YRSV		0	0	0		0	0
31GH	UY327002	Расход на производство	G	14	22	YRSV	YR16	14	47	9	N	11	2
31GL	UY327002	*31G9 YY_1 INV	F	0	0	YRSV	YY_1	0	0	0	D	8	0

Информация из элемента массива вышестоящей части 1, соответствующего полю YY_1 файла SYRM (строка 31G9), в режимах создания и редактирования будет копироваться в элемент массива нижестоящей части 2, соответствующий полю YY_1 файла YRSV (строка 31GL). Правила передачи информации из части в часть определяются системой автоматически на основании статусов частей и их расположения в графе документа (с учетом специфических правил для каждого копирования, если они явно указываются в строке с символом управления "F").

Отложенное копирование из Q

Если признак "F" используется для переноса значения поля "прокрутки" из классификаторов типа "K" или "Q", то проставить данный признак можно только после создания эталона документа, когда система определила служебные поля самих "K" и "Q". Описание такой операции в файле *SIUEK.DBF* приведено в таблице А.17.

Таблица А.17. Описание отложенного копирования из Q классификатора в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G1	UY327002	Поле части 2	?	0	0	YRSV	SYM1	0	0	0	C	2	0
...
31G5	UY327003	Ед.изм.	Q	11	58	SYRM	SYMY	11	66	9	C	2	0
31G9	UY327003	UY327030 SYMY SY03 ZY03	X	0	0			0	0	0		0	0
...
31GD	UY327004	Служебн.	X	0	0	YRSV		0	0	0		0	0
31GH	UY327004	Расход на производство	G	14	22	YRSV	YR16	14	47	9	N	11	2
31GL	UY327004	*31G9 ZY03 INV	F	0	0	YRSV	YY_1	0	0	0	D	8	0
...
31GQ	UY327030	СН ед.изм.	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GU	UY327030	Ед.изм.	S	0	0	SYMY	SY03	0	0	9	C	3	0
31GZ	UY327030	Производитель	S	0	0	SYMY	SY08	0	0	9	C	3	0

В соответствии с таблицей А.17 в части 3 вызывается образ "Q" классификатора (строка 31G5 с полем управления "Q"), при этом следующей строкой указывается, что поле прокрутки является поле SY03.

Описание образа "Q"-классификатора начинается со строки 31GQ и содержит 3 поля.

После создания эталона документа система определит служебное имя поля в файле *SYRM.DBF* (такое поле обязательно будет начинаться с символа "Z") для хранения значения поля прокрутки из Q-классификатора. Для рассматриваемого примера это поле с именем ZY03 (строка 31G9). Только после этого можно будет вставить строку с символом управления "F".

Для данного примера (табл. А.17) это строка 31GL, указывающая на необходимость копирования значения поля SY03 из файла *SYMY.DBF* ("Q"), вызываемого в части 3, в соответствующий полю YR17 файла *YRSV.DBF* элемент массива находящейся ниже части 4.

Следует отметить, что в данном случае копирование информации будет осуществлено только в момент, когда курсор пройдет поле управления с символом "F" части 4, т.е. является отложенным по времени. Если же копирование из Q-классификатора должно осуществляться в момент выбора из него значения (т.е. копирование является прямым), то в самом образе классификатора следует указать правило такого копирования так, как это показано в таблице А.18.

Прямое копирование из Q

Если признак "F" используется для копирования значения поля из классификаторов типа "Q" в другую часть документа или в часть, откуда происходит вызов классификатора, то этот признак указывается в самом образе классификатора. После адреса копирования в поле SI65 этой строки SIUEK обязательно указывается имя поля файла, в которое будет копироваться информация.

Перед определением данного правила следует создать эталон документа. При этом система определит служебное поле с первым символом "Z" в его имени, в котором будет храниться поле прокрутки Q.

Пример описания такого копирования в SIUEK приведен в таблице А.18 (строка 31G9).

Таблица А.18. Описание прямого копирования из Q классификатора в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327003	Ед.изм.	Q	11	58	SYRM	SYMY	11	66	9	C	2	0
31G9	UY327003	UY327030 SYMY SY03 ZY03	X	0	0			0	0	0		0	0
31GA	UY327003	Состав изделия	S	12	5	SYRM	SY04	12	25	0	C	10	0
...
31GD	UY327004	Служебн.	X	0	0	YRSV		0	0	0		0	0
31GH	UY327004	Расход на производство	G	14	22	YRSV	YR16	14	47	9	N	11	2
31GL	UY327004	*31G9 ZY03 INV	F	0	0	YRSV	YR17	0	0	0	C	3	0
...
31GQ	UY327030	СН ед.изм.	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GU	UY327030	Ед.изм.	S	0	0	SYMY	SY03	0	0	9	C	3	0
31GW	UY327030	*31GA SY04 OUT	F	0	0	SYMY	SY05	0	0	0	C	10	0

В этом примере, согласно содержащемуся в строке 31GW указанию на прямое копирование, при выборе какого-либо экземпляра из Q-классификатора *SYMY.DBF*, информация из поля SY05 этого файла будет скопирована в элемент массива части 3, соответствующий полю SYRM->SY04.

Порядок действий, выполняемых при этом системой, будет следующим:

1. В соответствующий полю SYRM->SYMY элемент массива части 3 заносится собственный якорь классификатора *SYMY.DBF*;
2. В элемент массива части 3, соответствующий полю SYRM->ZY03, заносится информация из поля прокрутки SYMY->SY03;
3. В элемент массива части 3, соответствующий полю SYRM->SY04 заносятся данные из поля SYMY->SY05;

4. И, наконец, если система (до получения указания "Сохранить документ") попадет на строку 31GL, то будет выполнено отложенное копирование поля прокрутки Q-классификатора в элемент массива части 4, соответствующий полю YR17 файла YRSV.

Если пользователь выберет действие "Сохранить документ", не дойдя до строки 31GL, то система проверит невыполненные операции отложенного копирования. Если невыполнение их не противоречит логике документа, то система проверит дополнительные признаки управления. На основании результатов такой проверки система либо скопирует информацию из части 3 в часть 4 (если от части 4 не отказались) в момент подготовки документа для записи в файлы (до исполнения пользовательских функций типа "Y" и "W") либо не станет производить такое копирование.

Если анализ неисполненных операций обнаружит противоречие, то система оповестит об этом пользователя соответствующим сообщением.

Прямое копирование простого поля из (в) Н

По определению собственный якорь модельного классификатора всегда заносится в часть, которой он подчинен. Кроме того, любая другая информация может копироваться из Н-классификатора (OUT) и в него (INV), если в его образе указать служебный символ управления "F".

При использовании признака "F" для копирования значения простого поля из классификаторов типа "Н" в другую часть документа, этот признак указывается в самом образе классификатора. После адреса копирования в поле SI65 этой строки SIUEK обязательно указывается имя поля файла, в которое будет копироваться информация и модификатор OUT. Пример такой ситуации представлен в таблице А.19.

Таблица А.19. Описание копирования простого поля Н-классификатора в файле SIUEK.DBF

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327002	Ед.изм.	G	11	58	SYRM	SYMC	11	67	9	C	2	0
31G9	UY327002	Z UY327003 010	H	0	0	SYRM	SYMY	0	0	0	C	2	0
31GA	UY327002	Состав изделия	S	12	5	SYRM	SY04	12	25	0	C	10	0
...
31GQ	UY327003	*31G9 СН комплектующ.	M	0	0	SYMY	SYMY	0	0	0		0	0
31GU	UY327003	Тип	S	14	22	SYMY	SY03	14	47	9	C	3	0
31GW	UY327003	*31GA SY04 OUT	F	0	0	SYMY	SY05	0	0	0	C	10	0

Представленный в этой таблице фрагмент предписывает произвести прямое копирование содержимого поля SYMY->SY05 модельного классификатора SYMY (экранный образ UY327003) в элемент массива части 2, соответствующий полю SYRM->SY04, в момент выбора значения из классификатора. Непосредственно это указание содержится в строке 31GW, все символы которой являются служебными и контролируются системой.

Модификатор OUT указывает, что копирование производится из модельного классификатора в элемент массива части, соответствующий строке 31GA. При этом строка 31GW не содержит ограничителей, определяющих правила изменения значения SYRM->SY04 при изменении значения SYMY->SY05. Поэтому система будет хранить в поле SYRM->SY04 документа типа 0327 первоначально выбранное значение поля SYMY->SY05 до момента следующего выбора значения из этого классификатора.

Примечание. Строка 31G9 содержит точку вызова модельного классификатора с кодом 010, указывающим, что из классификатора можно только выбрать существующую в нем запись, но нельзя добавить новую запись или редактировать существующую.

Строка 31GQ является началом описания экранного образа модельного классификатора и в поле SI65 всегда содержит адрес точки вызова модельного классификатора.

При необходимости копировать какие-то значения из других частей документа в сам Н-классификатор, в строке с символом управления "F" должен использоваться модификатор *INV* и один из ограничителей *NEWS*, *NEWR*, *NEWI* (см. ниже).

Прямое копирование значения поля К или Q-классификатора из (в) Н

При использовании признака "F" для копирования значения поля "K" или "Q" классификатора, принадлежащего части типа "H", в другую часть документа, этот признак указывается в строке образа Н-части. После адреса копирования в поле *SI65* этой строки *SIUEK* обязательно указывается имя поля файла, в которое будет копироваться информация, модификатор *OUT* и символ типа классификатора, из которого производится копирование. Описание такого случая представлено в таблице А.20.

Таблица А.20. Описание копирования поля классификатора из Н-части в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY327002	Ед.изм.	G	11	58	SYRM	SYMC	11	67	9	C	2	0
31G9	UY327002	Z UY327003 010	H	0	0	SYRM	SYMY	0	0	0	C	2	0
31GA	UY327002	Состав изделия	S	12	5	SYRM	SY04	12	25	0	C	10	0
...
31GQ	UY327003	*31G9 СН комплектующ.	M	0	0	SYMY	SYMY	0	0	0		0	0
31GU	UY327003	Тип	S	14	22	SYMY	SY03	14	47	9	C	3	0
31GW	UY327003	*31GA SY04 OUT K	F	0	0	SYMY	SY05	0	0	0	C	10	0
31HA	UY327003	Наименование	2 K	0	0	SYMY	SY05	0	0	0	C	10	0

Представленный в этой таблице фрагмент задает прямое копирование содержимого поля *SYMY->SY05* (поле прокрутки вложенного К-классификатора) в момент выбора значения из Н-классификатора *SYMY* (экранный образ *UY327003*) в элемент массива части 2, соответствующий полю *SYRM->SY04*. Это указание содержится в строке *31GW*, все символы которой являются служебными и контролируются системой.

Модификатор *OUT* указывает, что копирование производится из модельного классификатора в элемент массива части, соответствующий строке *31GA*. При этом строка *31GW* содержит указание на то, что копируемое поле *SY05* является полем К-классификатора. Кроме того, как и в предыдущем примере, эта строка не содержит ограничителей, определяющих правила изменения значения *SYRM->SY04* при изменении значения *SYMY->SY05*. Поэтому система будет хранить в поле *SYRM->SY04* документа типа 0327 первоначально выбранное значение поля *SYMY->SY05* до момента следующего выбора значения из модельного классификатора.

Примечание. Строка *31G9* содержит точку вызова модельного классификатора с кодом 010, указывающим, что из классификатора можно только выбрать существующую в нем запись, но нельзя добавить новую запись или редактировать существующую.

Строка *31GQ* является началом описания экранного образа модельного классификатора и в поле *SI65* всегда содержит адрес точки вызова модельного классификатора.

При необходимости копировать какие-то значения из других частей документа в сам Н-классификатор, в строке с символом управления "F" должен использоваться модификатор *INV* и один из ограничителей *NEWS*, *NEWR*, *NEWI* (см. ниже).

Если из модельного классификатора необходимо произвести копирование *не значения, а собственного якоря* К или Q-классификатора, то в поле управления строки *SIUEK*, содержащей команду копирования, записывается символ "J".

Прямое копирование из (в) Н с ограничением

В рассмотренных выше случаях копирования из Н с модификатором *OUT*, во всех экземплярах рассматриваемого типа документа, в которые производилось копирование

значения некоторого поля, система отслеживает изменения этого поля. Если копировавшееся значение будет модифицировано в Н-классификаторе, то во всех документах этого типа, в которых выполнялось копирование данного значения, оно также будет модифицировано.

Однако, такое изменение будет производиться только в момент установления новой связи между целевым объектом (экземпляром документа, в котором производится копирование) и модельным классификатором, т.е. в момент фиксации выбора (подтверждения) значения из него.

Такое правило будет называться правилом “принудительного” наследования, подчеркивая, что, если значение свойства (атрибута) A_i наследуется (копируется) объектом (документом) D_2 из объекта D_1 , то в случае его изменения в объекте D_1 , оно будет изменено и в объекте D_2 . (Причем, в данном случае изменение затрагивает весь класс объектов D_2 .)

Это правило будет всегда обрабатываться в режимах создания или редактирования документа D_2 . Однако его можно ограничить, если после модификатора *OUT* в команде копирования указать временное (или частное) правило наследования. Аналогичным образом можно ограничить и копирование в обратном направлении, т.е. в Н-классификатор, при значении модификатора, равном *INV*.

Как отмечалось ранее, ограничители могут принимать значения *NEWS*, *NEWR* и *NEWI* и указываются в команде копирования, начиная с 18-й позиции поля *SI65*.

Ограничитель *NEWS* указывает на то, что изменения скопированного из Н-классификатора или в Н-классификатор значения, если они имеют место в самом классификаторе или документе, соответственно, производятся **только в режиме создания** документа и только при фиксации выбора значения из Н-классификатора.

Ограничитель *NEWR* указывает на то, что изменения скопированного из Н-классификатора или в Н-классификатор значения, если они имеют место в самом классификаторе или документе, соответственно, производятся при редактировании **только в редактируемом экземпляре документа** и только в моменты фиксации выбора значения из Н-классификатора.

Ограничитель *NEWI* указывает на то, что изменения скопированного из Н-классификатора или в Н-классификатор значения, если они имеют место в самом классификаторе или документе, соответственно, производятся во всех случаях при редактировании документа в моменты фиксации выбора значения из Н-классификатора. При этом **старые значения сохраняются системой** в специальном файле.

Пример использования ограничителя *NEWS* с модификатором *OUT* приведен в таблице А.21.

Таблица А.21. Описание ограничения копирования *NEWS* в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
<i>31G5</i>	<i>UY327002</i>	Ед.изм.	<i>G</i>	11	58	<i>SYRM</i>	<i>SYMC</i>	11	67	9	<i>C</i>	2	0
<i>31G9</i>	<i>UY327002</i>	<i>Z UY327003 010</i>	<i>H</i>	0	0	<i>SYRM</i>	<i>SYMY</i>	0	0	0	<i>C</i>	2	0
<i>31GA</i>	<i>UY327002</i>	Состав изделия	<i>S</i>	12	5	<i>SYRM</i>	<i>SY04</i>	12	25	0	<i>C</i>	10	0
...
<i>31GQ</i>	<i>UY327003</i>	* <i>31G9</i> СН комплектующ.	<i>M</i>	0	0	<i>SYMY</i>	<i>SYMY</i>	0	0	0		0	0
<i>31GU</i>	<i>UY327003</i>	Тип	<i>S</i>	14	22	<i>SYMY</i>	<i>SY03</i>	14	47	9	<i>C</i>	3	0
<i>31GW</i>	<i>UY327003</i>	* <i>31GA SY04 OUT NEWS</i>	<i>F</i>	0	0	<i>SYMY</i>	<i>SY05</i>	0	0	0	<i>C</i>	10	0

Как следует из определения ограничителя *NEWS*, в примере, приведенном в таблице А.21, значение свойства модельного классификатора *SYMY->SY05* будет копироваться в часть 2 в *SYRM->SY04*. В дальнейшем, даже если значение этого свойства будет изменено в объекте, отраженном в части 3, оно сохранит свое первоначальное значение в объекте части 2.

Если может возникнуть необходимость все-таки изменить первоначально скопированное значение свойства, то перед ограничителем *NEWS* строке, содержащей команду копирования, ставится символ "?", как показано в таблице А.22.

Таблица А.22. Описание оповещения об ограничении копирования в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
<i>31GA</i>	<i>UY327002</i>	Состав изделия	<i>S</i>	12	5	<i>SYRM</i>	<i>SY04</i>	12	25	0	<i>C</i>	10	0
...
<i>31GQ</i>	<i>UY327003</i>	* <i>31G9</i> СН комплектующ.	<i>M</i>	0	0	<i>SYMY</i>	<i>SYMY</i>	0	0	0		0	0
<i>31GU</i>	<i>UY327003</i>	Тип	<i>S</i>	14	22	<i>SYMY</i>	<i>SY03</i>	14	47	9	<i>C</i>	3	0
<i>31GW</i>	<i>UY327003</i>	* <i>31GA SY04 OUT ?NEWS</i>	<i>F</i>	0	0	<i>SYMY</i>	<i>SY05</i>	0	0	0	<i>C</i>	10	0

Символ "?" перед ограничителем *NEWS* указывает системе на необходимость оповестить пользователя в случае изменения значения поля *SYMY->SY05*, которое ранее было скопировано. Если при этом пользователь обладает соответствующими правами доступа, то он может заменить старое значение в элементе массива части, соответствующего *SYRM->SY04* (строка *31GA*), новым.

Использование перечисленных правил копирования должно быть семантически совместимым с кодом разрешенных действий модельного классификатора.

Примеры копирования свойств в Н-классификатор рассмотрены в [6].

Согласованное копирование

Под согласованным копированием понимается выполнение одинаковых операций копирования в слоях многослойной части, вышестоящая часть которой обычно также является многослойной. Эта разновидность обозначается модификатором *COP* в команде копирования строки с признаком управления "F", причем оказываются возможными следующие случаи копирования:

- во всех слоях подчиненной части (указатель "m");
- во всех слоях части, подчиненных данному слою вышестоящей части (указатель "n");
- в одном слое (указатель "1"), что эквивалентно простому копированию.

В том случае, если вышестоящая часть является однослойной, указатели "m" и "n" дадут один и тот же результат.

Формат команды копирования (содержимое поля *SI65* строки с признаком управления "F") при использовании модификатора *COP* отличается от формата с модификаторами *INV* и *OUT* и имеет следующий вид:

```
<*> <адрес_строки_копируемого_поля> <пробел> <имя_копируемого_поля> <пробел>
'<i>COP</i>' <пробел> <адрес_строки_поля_назначения> <пробел> <имя_поля_назначения> <"m" |
"n" | "1">.
```

При выполнении команды копирования с модификатором *COP* может также вызываться пользовательская функция. В этом случае в последней позиции приведенного формата проставляется символ "P". Команда производит копирование в одном текущем слое части, а затем вызывает на исполнение пользовательскую функцию, записанную в следующей строке *SIUEK*. Особенности оформления вызова такой функции обсуждаются ниже (см. P-функция).

Согласованное копирование может выполняться совместно с другими видами копирования. Пример использования двух операций копирования в одной и той же части документа приведен в таблице А.23.

Таблица А.23. Описание двух операций копирования в одной части в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор	Коор	Файл	Поле	Y	X	Пои	Тип	Дли	Гочн
-------	----------	------------------------	-----	------	------	------	------	---	---	-----	-----	-----	------

				Y	X			поля	поля	СК		на	ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327002	Ед.изм.	G	11	58	SYRM	SYMC	11	67	9	C	2	0
31G9	UY327002	Z UY327003 010	H	0	0	SYRM	SYMY	0	0	0	C	2	0
31GA	UY327002	Исходный состав изделия	S	12	5	SYRM	SY04	12	25	0	C	10	0
...
31GQ	UY327003	*31G9 СН комплектующ.	M	0	0	SYMY	SYMY	0	0	0		0	0
31GU	UY327003	Тип	S	14	22	SYMY	SY03	14	47	9	C	3	0
31GV	UY327003	*31GU SY03 COP 31GW SY051	F	0	0			0	0	0		0	0
31GW	UY327003	*31GA SY04 OUT NEWS	F	0	0	SYMY	SY05	0	0	0	C	10	0

В данном случае код разрешенных действий (Z UY327003 111) для объекта, представленного модельным классификатором, позволяет выбирать любые экземпляры H-объекта и модифицировать их при этом, в том числе и создавать его новые экземпляры при создании (редактировании) док-та "0327".

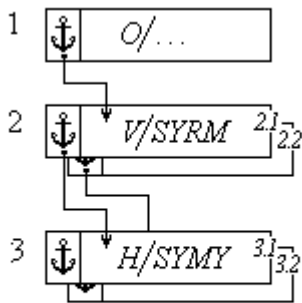


Рис. А.6. Граф экземпляра документа 0327

Логика связи частей документов описывается в файле *SIAGF* и, частично, в *SIUEK*, оказывая влияние на семантику согласованного копирования. Пусть структура документа типа "0327" выглядит так, как показано на рисунке А.6 (причем на рисунке сразу показано наличие двух слоев в части 2 и подчиненной ей части 3).

Строка 31GV задает операцию согласованного копирования из SYMY->SY03 в SYMY->SY05, указывая, что данную операцию следует выполнять в текущем слое "H", если значение SYMY->SY03 модифицируется. Классификатор "H"-типа может иметь связь 1:M, т.е.,

каждый слой "V" может наследовать в момент создания различные значения свойства из "H". Таким образом, данное правило позволяет учитывать динамику изменения объекта "H" при установлении этой связи.

Строка 31GW задает передачу в SYRM->SY04 из SYMY->SY05 уже скопированного в последнее поле по команде предыдущей строки значения SYRM->SY03. Но, эта операция будет выполняться только при создании экземпляра документа "0327", так как в данной строке использован ограничитель *NEWS*.

Изменить наследованное значение свойства в дальнейшем нельзя, так как это запрещает ограничитель *NEWS*. Точнее, это можно сделать уже только внешними средствами, непосредственно на файле.

Изменение самого объекта "H" при связывании и запоминание новых значений его свойств, также может быть регламентировано правилами. Это особенно важно, когда один и тот же экземпляр "H" связывается с несколькими слоями "V" и подвергается изменению непосредственно в сеансе работы с документом. При разработке моделей таких ситуаций, необходимо учитывать, к какой категории относится наследуемое свойство, так как в предметной области оно может представлять свойство, как объекта, так и связи, либо переходить из одной категории в другую.

Специальные случаи копирования по F

Командой копирования предусматривается три специальных случая:

- копирование типа документа (параметр команды "TDOK");
- копирование маршрута (параметр команды "MDOK");
- простое копирование.

Копирование типа документа используется в том случае, когда возникает необходимость запомнить в какой-либо части документа системный номер типа документа (*SIIM*), которому она принадлежит.

По умолчанию система всегда в основном файле документа формирует служебное поле с именем *SSSM*, в котором будет храниться системный номер типа документа, создающего запись данного файла.

Наличие данного поля в основном файле обусловлено тем, что система допускает использование одного и того же файла в качестве основного для различных типов документов. Поле используется для контроля различных противоречий, возникающих при работе с документами, использующими данный файл в качестве основного.

Однако, возможна ситуация, когда некоторый внутренний файл F_i документа получает статус внешнего классификаторного (см. [6]), что кодируется символом "v".

По определению, внутренний файл является представителем свойств только данного типа документа (объекта) и не может использоваться в качестве основного, внешнего или внутреннего файла в документах другого типа. Внутренний файл также не может использоваться в других документах в качестве "Q" или "H"-классификаторов.

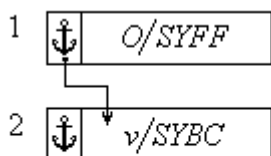
Если же разработчик приписывает некоторому внутреннему файлу F_i документа D_i свойства "Q" или "H"-классификаторов, т.е. дает ему статус внутреннего классификаторного, то при этом должны быть выполнены следующие условия:

1. В файлах *SIAGF* и *SIAOF* данного типа документа такого файла F_i указывается статус "v" (статус внутреннего классификаторного).
2. Документы, использующие файл F_i в качестве классификатора, должны содержать правила, не противоречащие природе файла F_i , как внутреннего свойства объекта (документа D_i).
3. Файл F_i должен иметь служебное поле *SSSM*, которое будет содержать информацию о типе документа, создающего запись данного файла.
4. В описании экранного образа части, содержащей F_i документа D_i , должно указываться правило копирования типа документа, фиксирующее системный номер типа документа в файле F_i .

В таблице А.24 приводится пример описания правила **копирование типа документа**, граф, структура которого представлен на рисунке А.7. Реальный пример использования статуса внутреннего классификаторного файла можно видеть в документах 0102 - "Организация" и 0203 - "Платежное поручение" в подсистеме F, поставляемой с полной версией системы *ORD*.

Таблица А.24. Описание копирования типа документа в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
3110	<i>UY307001</i>	Собственный якорь файла	<i>M</i>	0	0	<i>SYFF</i>	<i>SYFF</i>	0	0	0	<i>C</i>	4	0
3115	<i>UY307001</i>	Свойство 1	<i>G</i>	5	58	<i>SYFF</i>	<i>SYAC</i>	5	66	1	<i>C</i>	2	0
3120	<i>UY307001</i>	Свойство 2	<i>S</i>	7	30	<i>SYFF</i>	<i>SYF4</i>	8	41	3	<i>C</i>	10	0
3121	<i>UY307001</i>	Показать последнее первым	<i>X</i>	0	0	<i>SYFF</i>	<i>SYBC</i>	0	0	0	<i>C</i>	2	0
3123	<i>UY307002</i>	*3121	<i>M</i>	0	0	<i>SYBC</i>	<i>SYBC</i>	0	0	0	<i>C</i>	2	0
3127	<i>UY307002</i>	*3110 Внесен. из <i>SYFF</i>	<i>N</i>	0	0	<i>SYBC</i>	<i>SYFF</i>	0	0	0	<i>C</i>	4	0
3130	<i>UY307002</i>	lgr+	<i>R</i>	9	4			14	65	0		0	0
3132	<i>UY307002</i>	Свойство 1v	<i>G</i>	10	6	<i>SYBC</i>	<i>SY03</i>	10	37	9	<i>C</i>	3	0
313V	<i>UY307002</i>	Свойство 2v	<i>G</i>	12	6	<i>SYBC</i>	<i>SY04</i>	12	40	0	<i>C</i>	3	0
313W	<i>UY307002</i>	*313W <i>TDOK INV</i>	<i>F</i>	0	0	<i>SYBC</i>	<i>SSSM</i>	0	0	0	<i>C</i>	4	0



В таблице А.24 строка 313W содержит указание на копирование системного номера типа документа ("0307") в поле *SYBC*->*SSSM*, т.е., фактически эта строка предписывает системе заносить во все записи данного

Рис. А7. Структура документа с файлом статуса v

типа документа значение "0307". Служебное поле *SSSM* должно присутствовать в файле *SYBC*.

Если в части несколько файлов, то правило копирования типа документа записывается для каждого файла части.

Следует отметить, что наличие статуса "v" у файла *SYBC* предполагает его использование в качестве "Q", "H" или X (см [9], Приложение 2, Б. Особенности работы с "v" частью) в документах другого типа.

Кроме того, строка 3123 содержит скрытое правило, приказывающее системе в режимах "Создания" и "Редактирования" показывать многозначные свойства, записанные в файле *SYBC.DBF* в обратном порядке. Т.е. последняя добавленная в файл *SYBC.DBF* запись (относящаяся к данному экземпляру документа) будет показываться первой. Другими словами последнее значение свойства становится первым.

Для обеспечения данного порядка необходимо, чтобы в файле, которому подчинен файл типа "V" или "v", присутствовало имя якорного поля части {Vv} (см. строку 3121), а первая строка части типа {Vv} (см. строку 3123), содержала бы адрес строки с якорем согласования.

Копирование маршрута используется в тех случаях, когда при создании экземпляра документа одного типа должен быть сформирован и зафиксирован маршрут документа другого типа. (Эта операция, по сути, является не копированием, а формированием и записью маршрута.)

Примеры копирования маршрута рекомендуется посмотреть в [9], Приложение 2. Б. Особенности работы с "v" частью. Вид строки *SIUEK*, задающей копирование маршрута, представлен в таблице А.25.

Таблица А.25. Описание копирования маршрута в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
0045	<i>UY01B003</i>	*0045 <i>MDOK INV S01C 2</i>	<i>F</i>	0	0	<i>YGYR</i>	<i>S01C</i>	0	0	0	<i>C</i>	4	0

Как следует из таблицы, формат команды копирования имеет вид, несколько отличный от типового:

"*"<адрес_переноса> <"MDOK"> <"INV"> <имя_маршрутного_поля>
<количество_частей_маршрута>,
где

"*" – служебный символ;

0045 – адрес текущей строки *SIUEK*;

"MDOK" – параметр, указывающий, что данная строка требует записать единицы для каждой части в поле маршрута экземпляра документа, создаваемого одновременно с (порождаемого) данным документом;

"INV" – модификатор;

имя_маршрутного_поля – имя поля маршрута экземпляра документа, создаваемого одновременно с данным документом. В рассматриваемом примере – это поле *S01C* маршрута документа типа 001C, в соответствии с правилами формирования имени этого поля (имя сформировано по правилу: "S" + right(*TDOK*,3));

количество_частей_маршрута – количество частей (начиная с первой) экземпляра документа, создаваемого одновременно с данным документом, для которых в поле маршрута будут записаны единицы.

Примечание. Обычно, документ, экземпляр которого создается одновременно с данным документом, относится к другому типу, однако это не является обязательным требованием. Из вышесказанного также следует, что копирование маршрута выполняется *только в режиме создания документа*.

Простое копирование используется для дублирования вводимой в документ информации.

Пример описания операции простого копирования приведен в таблице А.26.

Таблица А.26. Описание операции простого копирования в файле *SIUEK.DBF*

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9001	UY01B002	Дата операции	X	0	0	YUFF	YU11	0	0	0	D	8	0
...
9101	UY01B003	Дата операции	G	7	30	YU00	YU11	7	45	0	D	8	0
9103	UY01B003	*9001*YU11 OUT NEWS9101	F	0	0	YU00	YU11	0	0	0	D	8	0

Как видно, в этом случае команда копирования также имеет несколько отличный формат: перед параметром – именем поля записан символ “*”, а после ограничителя *NEWS* записан адрес строки, указывающей источник копируемой информации.

Строка 9103 предписывает скопировать содержимое соответствующего ей элемента массива части в элемент массива части, соответствующий строке 9001.

Типы полей источника и приемника копируемой информации должны совпадать.

Простое копирование информации по F может осуществляться и в часть типа "H".

Модификатор *NEWS* закрывает возможность изменения скопированной при создании документа информации в режиме его редактирования. Однако, если известно, что может возникнуть необходимость все-таки изменить первоначально скопированное значение свойства, то команда копирования должна содержать знак вопроса перед модификатором *NEWS*, как показано в таблице А.27:

Таблица П.27. Разрешение редактирования при использовании модификатора *NEWS*

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9001	UY01B002	Дата операции	X	0	0	YUFF	YU11	0	0	0	D	8	0
...
9101	UY01B003	Дата операции	G	7	30	YU00	YU11	7	45	0	D	8	0
9103	UY01B003	*9001*YU11 OUT ?NEWS9101	F	0	0	YU00	YU11	0	0	0	D	8	0

Символ "?" перед ограничителем *NEWS* указывает системе на необходимость оповестить пользователя в случае изменения значения поля YU00->YU11, которое ранее было скопировано. Если при этом пользователь обладает соответствующими правами доступа, то он может заменить старое значение элемента массива части, соответствующего строке 9001, новым.

J – этот признак используется для копирования значения якорного поля "K" или "Q" классификатора в другое поле документа. В остальном его действие аналогично действию признака "F", а формат команды копирования, записываемой в поле SI65 строки *SIUEK* с признаком "J", соответствует формату для признака "F" (см. [команда копирования](#)).

Примеры, приведенные в таблицах А.28 ÷ А.30, иллюстрируют три из возможных вариантов записи в файле *SIUEK* операции копирования с использованием признака "J".

Таблица А.28. Описание двух операций копирования в одной части в файле *SIUEK.DBF*

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31G5	UY327003	Ед.изм.	Q	11	58	SYRM	SYMY	11	67	9	C	2	0

31G9	UY327003	UY327030 SYMY SY03 ZY03	X	0	0			0	0	0		0	0
31GD	UY327002	Служебн.	X	0	0			0	0	0		0	0
31GH	UY327002	Расход на производство	G	14	22	YRSV	YR16	14	47	9	N	11	2
31GK	UY327002	*31G9 SYMY INV	J	0	0	YRSV	SYMY	0	0	0		0	0
31GL	UY327002	*31G9 ZY03 INV	F	0	0	YRSV	YR17	0	0	0		0	0
31GQ	UY327030	СН ед.изм.	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GU	UY327030	Ед.изм.	S	0	0	SYMY	SY03	0	0	9	C	3	0

В данном примере команда копирования по "J" (строка 31GK) задает копирование якоря Q-классификатора из элемента массива части 3, соответствующего полю SYMY файла SYRM.DBF, в элемент массива нижестоящей части-продолжения, соответствующий полю SYMY файла YRSV.DBF. А команда копирования с признаком "F", записанная в следующей строке, задает операцию копирования значения Q-классификатора в элемент массива части-продолжения, соответствующий полю YR17 файла YRSV.DBF.

В том случае, когда при редактировании документа необходимо разрешить изменения копирующегося значения, но при этом зафиксировать это значение на момент входа в документ, применяется дублирование полей. Пример, приведенный в таблице А.29, иллюстрирует такую ситуацию для документа, структура которого показана на рисунке А.8.

Таблица П.29. Описание копирования с редактированием и фиксацией начального значения

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
31A0	UY327001	СЯ документа	M	0	0	AA01	AA01	0	0	0	C	2	0
31A3	UY327001	Состав накладной	M	0	0	AA02	AA02	0	0	0	C	2	0
31A4	UY327002	*31A0 ВСЯ документа.	N	0	0	AA02	AA01	0	0	0	C	2	0
31A5	UY327002	ORD дубль	?	0	0	AA02	sYmy	0	0	0	C	2	0
31A7	UY327002	ORD дубль	?	0	0	AA02	zy03	0	0	0	C	7	0
31A9	UY327002	Ед.изм.	Q	11	58	AA02	SYMY	11	67	1	C	2	0
31B0	UY327002	UY327030 SYMY SY03 ZY03	X	0	0			0	0	0		0	0
31B5	UY327003	Часть 3	M	0	0	AA03	AA03	0	0	0	C	2	0
31B6	UY327003	*31A3 ВСЯ части 2	N	0	0	AA03	AA02	0	0	0	C	2	0
31GH	UY327003	Расход на производство	G	14	22	AA03	YR16	14	47	9	N	11	2
31GK	UY327003	*31A9 SYMY INV_ NEWRsYmy	J	0	0	AA03	SYMY	0	0	0		0	0
31GL	UY327003	*31A9 ZY03 INV_ NEWRzy03	F	0	0	AA03	YR17	0	0	0		0	0
31GQ	UY327030	СН ед.изм.	M	0	0	SYMY	SYMY	0	0	0	C	2	0
31GU	UY327030	Ед.изм.	S	0	0	SYMY	SY03	0	0	9	C	3	0

В данном примере копирование производится из части 2 в часть 3.

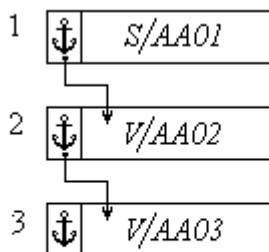


Рис А.8. Структура документа для примера таблицы А.29

Команда копирования, записанная в строке 31GK файла SIUEK.DBF, указывает, что из части 2 в элемент массива части 3, соответствующий полю SYMY файла AA03.DBF, будет копироваться (модификатор INV) собственный якорь классификатора с именем SYMY, выбор значения поля прокрутки, которого производится в строке Q (31A9). Ограничитель NEWR в этой команде, разрешает модифицировать это значение и предписывает, что в случае его изменения, собственный якорь вновь выбранной записи был бы скопирован во все слои части 3, подчиненные данному слою части 2.

скопирован во все слои части 3, подчиненные данному слою части 2.

Символ "_", записанный в команде копирования непосредственно после модификатора *INV*, указывает системе, что в команде копирования с символом управления "J" (31GK) присутствует продублированное поле с именем (начертанием) *sYmy*. Такое поле должно присутствовать в части, из которой идет копирование. Оно используется для хранения первоначального значения поля прокрутки классификатора, существовавшего в момент входа в экземпляр документа. Описание этого дублирующего поля помещено в строке с номером 31A5:

31A5	UY327002	ORD дубль	?	0	0	AA02	sYmy	0	0	0	C	2	0
------	----------	-----------	---	---	---	------	------	---	---	---	---	---	---

Как указывалось выше, символ управления "?" предписывает системе подкачать это поле в массив части в момент загрузки документа в память, а при записи документа не переписывать его на диск. Именно по значению этого поля в данном случае система определяет необходимость копирования нового выбранного значения из части 2 в часть 3.

Слова "ORD Дубль" система использует для того, чтобы отличить свои поля с символом управления "?", от пользовательских полей.

Часть, в которую производится копирование, не обязательно должна быть подчинена вышестоящей части непосредственно.

При использовании команды копирования с признаком "J" для копирования значений полей "K" или "Q"-классификаторов, принадлежащих "H"-части, в другую часть документа, эта команда записывается в строке образа H-части. После адреса копирования в поле *SI65* этой строки *SIUEK* обязательно указывается имя поля файла, в которое будет копироваться информация, модификатор *OUT* и символ типа классификатора, из которого производится копирование (так же, как и для признака "F"). Описание такого случая представлено в таблице А.30.

Таблица П.30. Описание копирования поля классификатора из H-части в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31G5	UY327002	Ед.изм.	G	11	58	SYRM	SYMC	11	67	9	C	2	0
31G9	UY327002	Z UY327003 010	H	0	0	SYRM	SYMY	0	0	0	C	2	0
31GA	UY327002	Состав изделия	S	12	5	SYRM	SY04	12	25	0	C	10	0
31GB	UY327002	СН состава изделия	X	0	0	SYRM	SY04	0	0	0	C	10	0
31GQ	UY327003	*31G9 СН комплектующ.	M	0	0	SYMY	SYMY	0	0	0		0	0
31GU	UY327003	Тип	S	14	22	SYMY	SY03	14	47	9	C	3	0
31GW	UY327003	*31GA SY04 OUT K	F	0	0	SYMY	SY05	0	0	0	C	10	0
31HA	UY327003	*31GB SY04 OUT K	J	0	0	SYMY	SY05	0	0	0	C	10	0
31HB	UY327003	Наименование	2 K	0	0	SYMY	SY05	0	0	0	C	10	0

При сравнении таблиц А.28 и А.30 можно заметить определенные синтаксические различия в записи пар команд копирования в отношении адресов строк и имен полей. Однако, эти различия не имеют принципиального значения и просто должны учитываться при описании соответствующих ситуаций.

А.7 Пользовательские и системные функции

А - функция пользователя, вызываемая автоматически.

Внимание! Имя функции пользователя должно состоять из восьми символов.

Пользовательские функции могут располагаться в любых частях документа, но будут вызываться системой на исполнение, если части, в которых они использованы, входят в маршрут экземпляра документа.

Например, в таблице А.31 показано описание вызова функции пользователя *USERFNC1* типа А в строке 9085 файла *SIUEK.DBF*.

Таблица А.31. Описание вызова функции пользователя в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907v	UF100001	Справочник БАНКОВ	Z	2	28			0	0	0		0	0
9084	UF100001	Функция пользователя	A	0	0			0	0	0		0	0
9085	UF100001	<i>USERFNC1</i>	X	0	0			0	0	0		0	0
908F	UF100001	Название	G	4	3	<i>SFBN</i>	<i>sf11</i>	4	14	1	C	62	0

Вызов функции осуществляется системой автоматически в момент прохождения данной строки системного массива *SARY* (см. пункт 4.5, рис. 28), при перемещении фокуса ввода мышкой или клавиатурой между двумя объектами, если один из них соответствует строке, имеющей индекс в массиве, меньший, чем строка вызова функции, а другой – строке, имеющей больший индекс. Причем вызов функции производится независимо от того, в каком направлении происходит перемещение фокуса: от предшествующего объекта к последующему или наоборот.

При вызове пользовательской функции система передает ей шесть параметров: *P1 ÷ P6*, оформляя вызов в виде *USERFNC1 (P1, P2, P3, P4, P5, P6)*. Вызываемая функция может и не использовать какие-либо или даже все их этих параметров в своем теле.

Фактические значения передаваемых параметров, определяются следующим образом:

- P1* – режим вызова документа: {"S", "G"}, (обычно не используется, т.к. может быть определен по значению *PUBLIC* переменной *UREG*), причем *P1* = "S", если документ вызван для просмотра и *P1* = "G", если документ вызван для создания или редактирования;
- P2* – индекс элемента массива *SARY*, в котором находится вызов функции *USERFNC1*, т.е. строки с символом управления "A"
- P3* – признак типа процедуры "A": *P3* = "A"
- P4* – код направления перемещения фокуса ввода (*PUBLIC* переменная *KOD*). *P4* = 1, если перемещение фокуса, обуславливающее вызов функции, является переходом от объекта строки с меньшим индексом к объекту строки с большим индексом; *P4* = 2, если направление перехода противоположное.
- P5* – индекс текущего слоя в массиве части, содержащей пользовательскую функцию (части с индексом *P6*), из которого произведен данный вызов функции
- P6* – динамический индекс (номер) части, в которой вызывается функция

Ниже приведен пример функции *USERFNC1*, которая используется для запрета редактирования характеристик объекта "Банк".

```

FUNCTION USERFNC1() //Запрет редактирования банка
MEMVAR UREG, SCrash
// UREG – глобальная переменная, передающая прикладной программе
//      информацию о режиме функционирования системы.
// SCrash – глобальная переменная, регулирующая процесс работы системы:
// SCrash = 1 – указание закончить процесс обработки документа

IF UREG = 4 // система находится в режиме редактирования
  MSay2 (5, 'Режим РЕДАКТИРОВАНИЯ ДОКУМЕНТА Б А Н К закрыт',;
  'Используйте режим ПРОСМОТР ДОКУМЕНТА ')
  // MSay2 – функция, выводящая сообщение на экран; исходный
  // текст хранится в программе MFAW.PRG
  SCrash = 1 // приказ аварийно закончить процесс обработки документа
ENDIF

```

RETURN.T.

Как видно из примера, если пользователь попытается войти на редактирование какого-либо 'Банка', пользовательская функция *USERFNC1* оповестит систему *ORD* о недопустимости операции означиванием глобальной переменной *SCrash*.

Примечание. Пользовательской функции типа "A" также доступна глобальная переменная *Crash*. Если функция типа "A" присвоит переменной *Crash* значение 1, то система закончит обработку документа. Если *Crash* будет присвоено значение 2, которое обычно используется при формировании функцией сообщения о какой-либо ошибке, то система вернется в процесс обработки документа, выставив фокус ввода в указанную или предшествующую позицию.

D – функция пользователя, контролирующая значение, введенное в GET-объект. Имя функции должно содержать 8 символов, также как и для функций типа "A".

Пусть, например, значения, вводимые в поле "Год", должны находиться в диапазоне 1900-2020, тогда это поле может контролироваться пользовательской функцией *USERFNC2*, описание вызова которой в файле *SIUEK.DBF* показано в таблице А.32.

Таблица А.32. Описание вызова функции пользователя типа D в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Гочн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907v	UF100001	Справочник БАНКОВ	Z	2	28			0	0	0		0	0
9084	UF100001	Год	G	3	35	SFBN	sf10	3	4	0	N	4	0
9085	UF100001	USERFNC2 sf10	D	0	0			0	0	0		0	0
908F	UF100001	Название	G	4	3	SFBN	sf11	4	14	1	C	62	0

Вызов функции осуществляется системой автоматически, также как и для функции типа "A", в момент прохождения данной строки системного массива *SARY*, при задании перемещения фокуса ввода мышкой или клавиатурой между объектами, один из которых соответствует строке, имеющей меньший индекс в массиве, чем строка вызова функции, а другой – строке, имеющей больший индекс.

Имя функции типа "D" указывается в строке, следующей после описания поля, значение которого должна контролировать функция. Причем за именем функции в этой же строке после пробела вновь указывается имя поля, которое контролирует функция (это сделано с целью выявления возможных неточностей при оформлении вызова функции).

При вызове функции типа "D" система передает ей шесть параметров: *P1 ÷ P6*, оформляя вызов в виде *USERFNC2 (P1, P2, P3, P4, P5, P6)*. Вызываемая функция может и не использовать какие-либо или даже все их этих параметров в своем теле.

Фактические значения передаваемых параметров, определяются следующим образом:

- P1* – индекс элемента массива *SARY*, соответствующего полю, которое контролирует вызываемая функция типа "D";
- P2* – код последней нажатой перед вызовом функции клавиши;
- P3* – индекс слоя в массиве части, содержащей пользовательскую функцию (части с индексом *P6*), из которого произведен текущий вызов функции
- P4* – значение, введенное в GET-объект пользователем;
- P5* – значение, хранившееся в GET-объекте до редактирования;
- P6* – динамический индекс (номер) части, в которой находится поле, контролируемое функцией и сама функция

Пример исходного кода функции *USERFNC2*, контролирующей значение поля на пребывание его в диапазоне 1900-2020, приведен ниже. (Функция использует только параметр *P4*)

```

FUNCTION USERFNC2 (P1, P2, P3, P4) //проверка введенного значения P4 поля sf10
IF P4 < 1900 .OR. P4 > 2020
    MSAY2 ("w/r", " Допустимый диапазон значений ", "    1900 - 2020 ")
        // функция, выводящая сообщение
    RETURN .F.
ENDIF
RETURN .T. // введенное значение удовлетворяет требованиям контроля

```

Функция типа "D" должна возвращать логическую величину, равную .T., если контролируемое значение удовлетворяет необходимым требованиям, и .F. – в противном случае.

Если функция не вернет логическое значение, *система оповестит пользователя* об этом и не позволит сохранить результаты текущего сеанса работы с документом.

Если функция вернет значение .T., то фокус (курсор) будет установлен на следующий GET-объект (если контролируемый объект не последний).

Если функция вернет значение .F., фокус (курсор) будет установлен на контролируемый функцией GET-объект. При этом если возникает необходимость установить фокус на GET-объект другого поля, то функция диапазона должна в своем теле содержать операторы (только для Windows версии), присваивающие PUBLIC переменным *iSaryPred* и *sOrd* численное значение, равное индексу строки массива *SARY*, которая соответствует этому GET-объекту.

Как и пользовательские функции других типов, функция типа "D" может определить характер действий системы по завершению своего исполнения, используя глобальную переменную *Crash*, которой можно присвоить следующие численные значения:

- 1 – ошибка, не позволяющая продолжить обработку документа;
- 2 – ошибка, позволяющая продолжить обработку документа;
- 3,4 – используются системой;
- 5 – срабатывание ограничения диапазона: при выполнении функции введенное пользователем значение заменено в массиве части значением, соответствующим заданному диапазону (возвращаемое логическое значение .T.);
- 7 – отсутствие права модификации объекта, диапазон значений которого контролирует функция;
- 9 – прекращение обработки документа без сохранения результатов текущего сеанса.

Для контроля системой значения переменной *Crash*, присвоенного ей функцией типа "D", функция, по завершению своего выполнения, должна вернуть логическое значение .F. во всех перечисленных случаях, кроме того, когда переменной *Crash* присвоено значение 5.

В приводимом ниже примере функции *AM000E_D* показаны некоторые варианты использования переменной *Crash*. Причем красным цветом в тексте программы выделены собственные функции системы *ORD*.

```

FUNCTION AM000E_D (y1, y2, y3, y4)
    //Контроль ввода значения цены: В поле "Цена" (аабк) – не должна быть введено
    //значение ниже минимальной установленной для данного товара цены

```

```

MEMVAR KOD, UREG, NP, SARY, PSI75, PSC75, FSIAGF, NPI, SORD, ISARYPred
LOCAL SP2, SP3, ZAA5A, STW, ARVoz, MinAA5F
LOCAL IAA0F
LOCAL IAA6J3 // индекс описания поля цены в массиве SARY
LOCAL IAA6K3 // индекс описания поля суммы в массиве SARY
LOCAL IAA6M3

```

```

SP2 = SARY[PSI75[FSIAGF + 2], 10]
IAA0F = ASCAN (SARY, {X| X[5] = "Aa0F" }, PSI75[FSIAGF + 2], PSC75[FSIAGF + 2])

```

```

IAA6K3 = ACSAN (SARY, {|X| X[5] = "aa6k"}, PSI75[FSIAGF + 3], PSC75[FSIAGF + 3])
IF IAA6K3 = 0
    Crash = 1 // ошибка в структуре документа. Прекратить его обработку
    MSAY ("Ошибка: в документе отсутствует поле aa6k")
    RETURN .F.
ENDIF

IF ! OPENF ("AA0F", "AA0F", .F.) //проверка открытия таблицы (файла) AA0F.DBF
    Crash = 2 // ошибка при открытии таблицы. Продолжить обработку документа
    ISARYPred = IAA6K3
    SORD = ISARYPred // установить фокус, на поле суммы
    RETURN .F. // не удалось открыть файл
ENDIF

IF ! OPENF ("AA5B", "AA5A", .F.) //проверка открытия файла AA5B.DBF
    Crash = 2 // ошибка при открытии файла. Продолжить обработку документа
    ISARYPred = IAA6K3
    SORD = ISARYPred // установить фокус, на поле суммы
    RETURN .F. // не удалось открыть файл
ENDIF

IF ! AA0F->(DBSEEK (IAA0F))
    MSAY ("Ошибка поиска значения IAA0F в файле AA0F")
    Crash = 2 // ошибка поиска. Продолжить обработку документа
    ISARYPred = IAA6K3 ; SORD = ISARYPred // установить фокус, на поле суммы
    RETURN .F.
ENDIF

ZAA5A = AA0F->AA5A
IF AA5B->(DBSEEK (ZAA5A))
    MinAA5F = AA5B->AA5F
    DO WHILE !AA5B->(EOF()) .AND. AA5B->AA5A = ZAA5A
        MinAA5F = MIN (AA5B->AA5F, MinAA5F)
        AA5B->(DBSKIP())
    ENDDO
ELSE
    MSAY ("Минимальная цена для товара " + ZAA5A + " не установлена")
    ISARYPred = IAA6K3 ; SORD = ISARYPred // установить фокус, на поле суммы
    RETURN .T.
ENDIF

IF MinAA5F > y4 .AND. y4 <> 0
    MSAY ("Минимальная цена для этого товара: " + STR (MinAA5F))
    Crash = 5 // взводится в 5, чтобы указать системе, что значение заменено
    // внутри пользовательской функции (возврат .T.)
    SP3[NPI[3], IAA6J3 - PSI75[FSIAGF + 3] + 1] = MinAA5F
    SII_OSAYD (3, PSI75[FSIAGF + 3], PSC75[FSIAGF + 3]) //вывод данных на экран
ENDIF
SOPENCLOSE ("AA0F") //закрытие таблицы AA0F
SOPENCLOSE ("AA5B") //закрытие таблицы AA5B
ISARYPred = IAA6K3 ; SORD = ISARYPred // установить фокус, на поле суммы
RETURN .T.

```


Если в качестве имени функции указать *ORDDIAPZ*, то будет вызвана системная функция *ORDDIAPZ* (исходный текст поставляется), позволяющая контролировать и предусловия входа в поле.

P - функция пользователя, выводящая результат на экран.

В отличие от функции типа "А", функция типа "P" вызывается при нажатии клавиши *<Enter>* (или двойным щелчком мыши), в тот момент, когда фокус ввода установлен на GET-объекте, связанным с каким-либо полем документа. Возвращаемое функцией значение автоматически заносится системой в элемент массива части, соответствующий этому GET-объекту, и высвечивается на экране. Имя функции указывается в *SIUEK* в поле *SI65* (имя атрибута на экране), в строке, следующей после строки с описанием поля, на котором вызывается данная функция.

При вызове функции система передает ей шесть параметров *P1 ÷ P6*, часть или даже все из которых могут и не использоваться.

Фактические значения передаваемых параметров, определяются следующим образом:

- P1* – режим вызова документа: {"S", "G"}, (обычно не используется, т.к. может быть определен по значению PUBLIC переменной *UREG*), причем *P1* = "S", если документ вызван для просмотра и *P1* = "G", если документ вызван для создания или редактирования;
- P2* – значение, оказавшееся в GET-объекте к моменту вызова функции;
- P3* – индекс элемента массива *SARY*, в котором находится вызов функции, т.е. строки с символом управления "P"
- P4* – признак типа процедуры "P": *P4* = "P"
- P5* – индекс текущего слоя в массиве части, содержащей пользовательскую функцию (части с индексом *P6*), из которого произведен данный вызов функции
- P6* – динамический индекс (номер) части, в которой вызывается функция

Функция, после выполнения необходимых вычислений, должна вернуть значение, которое будет автоматически занесено в элемент массива части с индексом, равным значению параметра *P3*, и высвечено на экране в месте, отведенном соответствующему GET-объекту.

Ниже приведен пример функции *USERFNCP*, подсчитывающей сумму по слоям поля многослойной части.

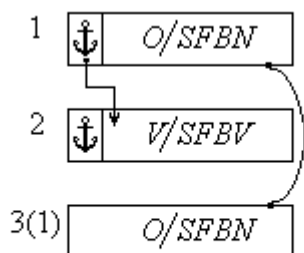


Рис. А 9. Граф структуры документа с полем суммы в части-продолжении

Пусть в документе типа "0907", структура которого приведена на рисунке А.9, в поле "Общее количество" *SF01* файла *SFBN.DBF*, отнесенного к части 3 документа, необходимо занести суммарное количество статей прихода, записанных в слоях элемента массива части 2, соответствующего полю *SF11* файла *SFVN.DBF*. Описание вызова в файле *SIUEK.DBF* функции *USERFNCP*, подсчитывающей эту сумму, приведено в

таблице А.33.

Таблица А.33. Описание вызова функции подсчета суммы в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>

9071	UF100001	СЯ карточки	M	0	0	SFBN	SFBN	0	0	0	C	3	0
9072	UF100001	КАРТОЧКА УЧЕТА	Z	2	28			0	0	0		0	0
9073	UF100001	Общее кол-во (исходное)	X	0	0	SFBN	Sf01	0	0	0	N	14	0
9077	UF100001
9080	UF100002	СЯ прихода	M	0	0	SFBV	SFBV	0	0	0	C	4	0
9081	UF100002	*9071 ВСЯ карточки	N	0	0	SFBV	SFBN	0	0	0	C	3	0
9082	UF100002	1gr+	R	3	2			7	67	0		0	0
9083	UF100002	Дата прихода	G	4	3	SFBV	Sf09	4	20	0	D	8	0
9085	UF100002	Приход, шт. (исходное)	X	0	0	SFBV	Sf11	0	0	0	N	12	0
9087	UF100002	Приход, шт.	G	4	38	SFBV	Sf11	4	52	1	N	12	0
9089	UF100001	Служебное	X	0	0	SFBN		0	0	0		0	0
9091	UF100001	Общее кол-во, шт.	P	6	5	SFBN	Sf01	6	15	0	N	14	0
9093	UF100001	USERFNCP	X	0	0			0	0	0		0	0

В многослойной части 2 (рис. А9), пользователь осуществляет ввод количественных данных о приходе за разные дни в GET-объект (ы), соответствующий (ие) полю *SFBV->sf11*. Задача пользовательской функции *USERFNCP* подсчитать сумму введенных значений (строка 9087), учитывая режим (создание или редактирование), в котором обрабатывается документ. Исходный текст функции следующий:

FUNCTION *USERFNCP* (*P1*, *P2*, *P3*, *P4*, *P5*, *P6*)

//Подсчет общей суммы значений по слоям поля *SVBF->sf11* и занесение ее
//поле *SFBN->Sf01*

MEMVAR *UREG*, *SARY*, *KPP*, *PSI75*, *FSIAGF* //PUBLIC переменные системы *ORD*

// *SARY* – массив экранного образа документа
// *KPP* – массив, указывающий, сколько слоев находится в массивах частей
// *UREG* – режим работы с документом
// *PSI75* – массив индексов начала частей документа в *SARY*
// *FSIAGF* – базовый адрес расположения документа в памяти

LOCAL *F*, *FN2*, *FNI*, *FNS*, *FNX*, *FMOD*

FN2=PSI75[FSIAGF + 2] // индекс строки начала второй части в *SARY*
FNI = ASCAN (SARY, {|X| X[5] = "sf11"}, FN2) // индекс строки *SARY*, содержащей
//поле *sf11*, по которому надо просчитать сумму

IF *UREG* = 2 // режим создания документа
P2 = 0 // очистка начального значения суммы
FOR *F* = 1 **TO** *KPP*[2] // цикл по всем слоям части 2
P2 + = **MLPICK** (*FNI*, 2, *F*) // формирование суммы
// функция **MLPICK** (*X*, *P*, *L*) возвращает значение элемента
// с индексом *X* из слоя *L* части *P*; исходный текст функции
// находится в модуле *MFAW.PRG*

NEXT

ENDIF

IF *UREG* = 4 // режим редактирования документа
FNS = ASCAN (SARY, {|X| X[5] = "sf01"}) // индекс строки *SARY*, описывающей поле
исходного значения суммы *sf01*

P2 = **MLPICK** (*FNS*, 1, 1) // значение суммы до редактирования
FNX = ASCAN (SARY, {|X| X[5] = "Sf11"}, FN2) // индекс строки *SARY*, содержащей поле
//*Sf11*, хранящее "Приход" до редактирования
FOR *F* = 1 **TO** *KPP*[2] // цикл по всем слоям части 2

```

FMOD = MCCPICK (2, F)           // признак модификации слоя части 2
// функция MCCPICK (P, L) возвращает значение признака модификации слоя L части P;
// исходный текст функции находится в модуле MFAW.PRG
DO CASE
CASE FMOD = "2"                 // слой взят из файла и модифицирован
P2 - = MLPICK (FNX, 2, F) + MLPICK (FNI, 2, F)
                                //вычитается старое и добавляется новое значение

CASE FMOD = "3"                 // слой взят из файла и удален
P2 - = MLPICK (FNX, 2, F)       //вычитается старое значение

CASE FMOD = "4"                 // слой новый и модифицирован (непустой)
P2 + = MLPICK (FNI, 2, F)       //добавляется новое значение
ENDCASE
NEXT
ENDIF
RETURN P2                       // возврат нового значения итоговой суммы

```

Примечание. Пользовательской функции типа "P" доступна глобальная переменная *Crash*. Если в функции типа "P" этой переменной присвоить значение 1, то система аварийно закончит обработку документа. Если *Crash* присвоено значение 2, то это предписывает вернуться в процесс обработки документа на исправление ошибки.

В системе имеются стандартные системные функции, позволяющие учитывать динамическое поведение объекта, в котором они используются. Так, например, операции "Сумма" соответствует стандартная системная функция *SUMM*.

Эта функция вызывается на исполнение с двумя параметрами *P1* и *P2*.

Параметр *P1* задает расчетную формулу, по которой вычисляется возвращаемое функцией значение, имеющую вид произвольного арифметического выражения, термами которого являются имена полей файлов документа. Особым условием записи этого выражения является требование отсутствия в нем пробелов (space).

Параметр *P2* определяет особенности вызова функции и может иметь одно из следующих значений: "N", "A", "W".

При *P2* = "N" функция вызывается как функция типа "P", т.е., по нажатию клавиши <Enter>.

При *P2* = "A" функция вызывается как функция типа "A", т.е., автоматически (см. выше A - функция). Однако, автоматический вызов произойдет только после прохождения последнего поля, являющегося параметром вычисляемого арифметического выражения. Результат выполнения функции обязательно высвечивается на экране после ее завершения в GET-объекте поля, контролируемого функцией.

При *P2* = "W" функция вызывается как функция типа "W", т.е., при выходе из документа с сохранением результата (см. ниже W - функция).

Кроме того, если *P2* = "N" или *P2* = "A", то при выходе из документа с сохранением функция вызывается повторно, также как и в случае *P2* = "W".

В таблице А.34 дан пример использования функции *SUMM* в документе, структура которого показана на рисунке А.10.

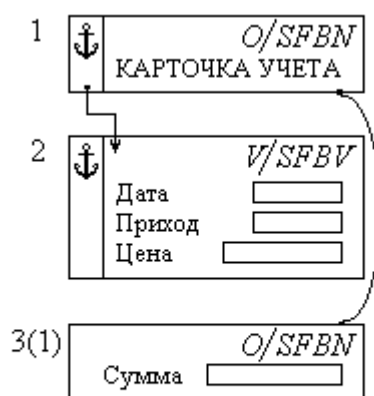


Рис.А.10. Граф структуры документа, содержащего функцию *SUMM*

Таблица А.34.Описание вызова функции *SUMM* в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Гочн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9071	UF100001	СЯ карточки	M	0	0	SFBN	SFBN	0	0	0	C	3	0
9072	UF100001	КАРТОЧКА УЧЕТА	Z	2	28			0	0	0		0	0
9076	UF100002	СЯ прихода	M	0	0	SFBV	SFBV	0	0	0	C	4	0
9079	UF100002	*9071 ВСЯ карточки	N	0	0	SFBV	SFBN	0	0	0	C	3	0
907A	UF100002	lgr+	R	3	2			12	50	0		0	0
907D	UF100002	Дата	G	4	10	SFBV	SF09	4	30	0	D	8	0
907F	UF100002	Контроль ORD	X	0	0	SFBV	SF11	0	0	0	N	8	0
9082	UF100002	Контроль ORD	X	0	0	SFBV	SF11	0	0	0	N	8	0
9085	UF100002	Приход	G	6	10	SFBV	sf11	6	30	1	N	8	0
9087	UF100002	Контроль ORD	X	0	0	SFBV	SFKK	0	0	0	N	12	0
908A	UF100002	Контроль ORD	X	0	0	SFBV	SFKK	0	0	0	N	12	0
908F	UF100002	Цена	G	8	10	SFBV	sfkk	8	26	0	N	12	0
9091	UF100001	Служебная	X	0	0	SFBN		0	0	0		0	0
9093	UF100001	Дубль Рез.ORD	X	0	0	SFBN	SF01	0	0	0	N	14	0
9095	UF100001	Сумма	P	11	10	SFBN	Sf01	11	24	0	N	14	0
9097	UF100001	<i>SUMM(sf11*sfkk,N)</i>	X	0	0			0	0	0		0	0
9099	UF100001	<i>(a*b)</i>	X	0	0			0	0	0		0	0

В этом примере в многослойной части 2 пользователь может ввести или изменить некоторые значения в GET-объекты, соответствующие полям *SFBV->sf11* и *SFBV->sfkk*.

Функция *SUMM*, в данном случае, должна подсчитать во всех слоях части 2 произведения значений, введенных в GET-объекты полей “Приход” (строка 9085) и “Цена” (строка 908F) каждого слоя, и занести сумму этих произведений в GET-объект поля ‘Сумма’ части 3 (строка 9095).

Арифметическое выражение, передаваемое в функцию *SUMM* в качестве фактического значения первого параметра, дублируется в виде прототипа $(a*b)$ для целей контроля в строке, следующей за функцией (строка 9099).

Синтаксис арифметического выражения соответствует общепринятым правилам и допускает использование любых арифметических операций. Порядок выполнения операций может регламентироваться круглыми скобками. *Использовать символы пробелов при записи выражения не допускается.*

Примечания. 1. Первый символ арифметического выражения - прототипа, записываемого в скобках в строке, следующей за строкой с функцией *SUMM*, должен обозначать его первый операнд. Так как это арифметическое выражение повторяет по форме выражение, переданное функции в качестве параметра, то данный символ должен соответствовать и его первому операнду. Например, в рассматриваемом случае символ “a” в строке 9099 соответствует полю *sf11* первого параметра функции *SUMM* в строке 9097.

Каждый следующий операнд обозначается очередной буквой латинского алфавита.

В качестве операнда может использоваться и константа, но она не может быть первой в выражении. Например, $(a*0.8+b*0.3)$.

2. Строка с выражением-прототипом должна идти сразу после строки с функцией *SUMM*.

3. Всем строкам, которым соответствуют поля, использованные в качестве операндов арифметического выражения, переданного в функцию, должны непосредственно предшествовать по две контрольные строки с текстом 'Контроль *ORD*' в поле *SI65*.

Перед строкой, соответствующей полю результата, должна размещаться контрольная строка с текстом 'Дубль Рез. *ORD*' в ее поле атрибута.

Имена полей в контрольных строках должны быть записаны символами верхнего регистра.

4. Если содержимое функции не умещается в поле *SI65*, то для переноса его в следующую строку необходимо использовать символа “;” (semicolon – точку с запятой).

5. Функция вызывается на исполнение только в режимах создания (*UREG* = 2) и редактирования (*UREG* = 4).

6. Функция может встраиваться в документ в режиме визуальной модификации образа документа утилитой *DB.EXE*. Для этого надо указать имя *SUMM* в поле 'Контроль' при модификации статуса поля и система автоматически запишет служебную информацию в *SIUEK*

7. Как правило, все поля, участвующие в операции *SUMM*, должны находиться в образе документа выше, чем поле, в котором регистрируется результат функции.

Функция *SUMM* может использоваться и для подсчета значения арифметического выражения только в одном слое. В таблице А.35 дан пример такого использования функции *SUMM* в документе, структура которого показана на рисунке А.11 и отличается от предыдущего примера наличием дополнительного расчетного поля “Сумма” во второй части.

На рис. А11 в части 2 и в части 3 используется стандартная функция *SUMM*. При этом во

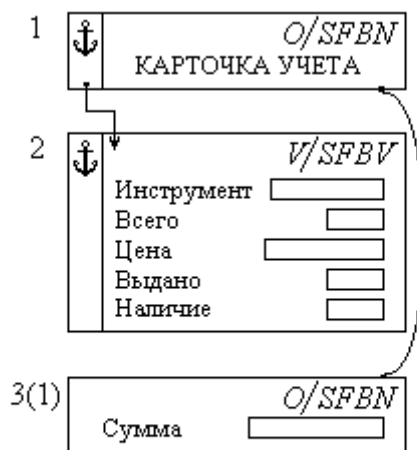


Рис. А 11. Граф структуры документа, содержащего две функции *SUMM*

второй части сумма считается в каждом слое, а в третьей части сумма подсчитывается по всем слоям второй части.

Таблица А.35. Описание различных вызовов функции *SUMM* в файле *SIUEK.DBF*

Сист.	№ образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Точн ость
SI75	SI52	SI65	SI70	SI66	SI67	SI20	SI26	SI68	SI69	SI72	SI56	SI57	SI76
9071	UF100001	СЯ карточки	M	0	0	SFBN	SFBN	0	0	0	C	3	0
9072	UF100001	КАРТОЧКА УЧЕТА	Z	2	28			0	0	0		0	0
9076	UF100002	СЯ прихода	M	0	0	SFBV	SFBV	0	0	0	C	4	0
9079	UF100002	*9071 ВСЯ карточки	N	0	0	SFBV	SFBN	0	0	0	C	3	0
9076	UF100002	1gr+	R	3	2			14	50	0		0	0
907D	UF100002	Инструмент	2K	4	10	SFBV	SF09	4	27	1	C	11	0
907F	UF100002	Контроль ORD	X	0	0	SFBV	SF11	0	0	0	N	6	0
9082	UF100002	Контроль ORD	X	0	0	SFBV	SF11	0	0	0	N	6	0
9085	UF100002	Всего	G	6	10	SFBV	sf11	6	32	2	N	6	0
9087	UF100002	Контроль ORD	X	0	0	SFBV	SFKK	0	0	0	N	12	0
908A	UF100002	Контроль ORD	X	0	0	SFBV	SFKK	0	0	0	N	12	0
908F	UF100002	Цена	G	8	10	SFBV	sfkk	8	26	0	N	12	0
9092	UF100002	Контроль ORD	X	0	0	SFBV	SF12	0	0	0	N	6	0
9095	UF100002	Контроль ORD	X	0	0	SFBV	SF12	0	0	0	N	6	0
9098	UF100002	Выдано	G	10	10	SFBV	Sf12	10	32	0	N	6	0
909A	UF100002	Дубль Рез.ORD	X	0	0	SFBV	SF15	0	0	0	N	6	0
909F	UF100002	Наличие	P	12	10	SFBV	SF15	12	32	0	N	6	0
90A2	UF100002	SUMM(sf11-Sf12.A)	X	0	0			0	0	0		0	0
90A5	UF100002	(a-b)	X	0	0			0	0	0		0	0
90A8	UF100001	Служебная	X	0	0	SFBN		0	0	0		0	0
90AB	UF100001	Дубль Рез.ORD	X	0	0	SFBN	SF01	0	0	0	N	14	0
90AF	UF100001	Сумма	P	16	10	SFBN	Sf01	16	24	0	N	14	0
90B5	UF100001	SUMM(sf11*sfkk.N)	X	0	0			0	0	0		0	0
90B9	UF100001	(a*b)	X	0	0			0	0	0		0	0

В отличие от предыдущего примера функция *SUMM* вызывается и во второй и в третьей части. Во второй части (строка 90A2) она подсчитывает сумму в каждом слое, заносит результат в GET-объект, соответствующий полю *SFBV*->*SF15*, и высвечивает его в строке 12, начиная со столбца 42 (в графическом режиме монитора пиксельные координаты, естественно, другие). Причем функция вызывается автоматически после ввода последнего из значений, входящих в вычисляемое арифметическое выражение.

В части 3 (строка 91B5) функция будет вызываться на исполнение только после нажатия пользователем клавиши <Enter>. Вычисленное произведение будет выводиться в 16-ой строке, начиная со столбца 24.

Функция *SUMM* обладает свойствами функции типа "W", поэтому перед записью документа на диск в режиме выхода с сохранением, она будет вызвана повторно для исключения возможного изменения пользователем значений аргументов без перерасчета результирующего значения вычисляемого выражения.

W – функция пользователя, исполняемая после вызова пользователем режима записи документа на диск: "Выход с сохранением".

К моменту исполнения данной функции информация документа находится в буфере системы *ORD* (в массивах частей), но еще не записана в файлы документа. Поэтому пользовательская функция имеет возможность определить, не нарушаются ли какие-либо семантические ограничения предметной области для данного экземпляра документа. Если какие-либо нарушения имеются, то функция может оповестить систему о недопустимости занесения некорректной информации в базу данных.

Пример вызова функции типа "W" с именем *FCONBNK1* приведен в таблице А.36. Параметры в функцию при ее вызове не передаются.

Функция вызывается на исполнение только в режимах создания (*UREG* = 2) и редактирования (*UREG* = 4) и может располагаться в любой части документа. Однако вызов

функции будет производиться только в том случае, если часть, в которой она размещена, входит в маршрут экземпляра документа.

Таблица А.36. Описание вызова функции типа *W* в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Поиск	Тип	Дли на	Точн ость
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907v	UF100001	Справочник БАНКОВ	Z	2	28			0	0	0		0	0
9085	UF100001	FCONBNK1	W	0	0			0	0	0		0	0
908F	UF100001	Название	G	4	3	SFBN	SF11	4	14	1	C	62	0

Ниже приводится пример исходного текста функции типа "W" *FCONBNK1()*

```
FUNCTION FCONBNK1 () //Проверка правильности заполнения документа
```

```
MEMVAR NP, KPP, Crash //PUBLIC переменные системы ORD
```

```
// NP – массив номеров частей, имеющих в экземпляре документа
```

```
// KPP – массив, указывающий, сколько слоев находится в массивах частей
```

```
// Crash – код характера ошибки обработки документа
```

```
IF ASCAN (NP, 8) = 0 // проверка наличия восьмой части документа
```

```
MSAY2 ("w/r", " Без части 8 документ ", " не имеет смысла ")
```

```
// функция, выводящая сообщение на экран
```

```
Crash = 1 // прекратить обработку документа
```

```
RETURN .F.
```

```
ENDIF
```

```
IF KPP[8] = 0 // проверка наличия записей в восьмой части документа
```

```
MSAY2 ("w/r", " Без данных части 8 документ ", " не может заноситься в файлы")
```

```
// функция, выводящая сообщение на экран
```

```
Crash = 2 // вернуться на исправление ошибки в документе
```

```
RETURN .F.
```

```
ENDIF
```

```
RETURN .T.
```

Как видно из примера, пользовательская функция типа "W" указывает системе (с помощью глобальной переменной *Crash*), какие следует предпринять действия в случае обнаружения ошибок пользователя.

Y – функция пользователя, исполняемая после записи экземпляра документа в файлы.

К моменту исполнения данной функции информация документа уже переписана в его файлы. Пользовательская функция имеет возможность определить, не нарушаются ли какие-либо семантические ограничения предметной области для данного экземпляра документа. Если какие-либо нарушения имеются, то функция типа "Y" может оповестить систему о некорректности, выполненной операции.

Если пользовательская функция установит значение глобальной переменной *Crash*, равным 1, то система оповестит пользователя, что документ занесен в файлы, в нем есть противоречия, выйдет из документа, и будет ожидать указания пользователя на выполнение процедуры восстановления предыдущего состояния.

Если пользовательская функция установит значение глобальной переменной *Crash* = 2, то система оповестит пользователя, что документ занесен в файлы, в нем есть противоречия, но вернется в обработку документа и предоставит пользователю возможность исправить ошибку.

Если, функция установила значение *Crash*, равное 1 или 2, то до устранения ошибки данный документ будет доступен только машине, проводшей с ним последний процесс.

Внимание! При удалении документа функции типа "Y" также вызываются для исполнения.

Пример вызова функции типа "Y" с именем *FCONBNK2* приведен в таблице А.37. Параметры в функцию при ее вызове не передаются.

Таблица А.37. Описание вызова функции типа Y в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точ ность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907v	UF100001	Справочник БАНКОВ	Z	2	28			0	0	0		0	0
9085	UF100001	<i>FCONBNK2</i>	Y	0	0			0	0	0		0	0
908F	UF100001	Название	G	4	3	<i>SFBN</i>	<i>SF11</i>	4	14	1	C	62	0

U - стандартная системная функция с именем *FUNIQ*.

Эта функция предназначена для проверки уникальности значения, вводимого в GET-объект, соответствующий некоторому полю.

Функция вызывается на исполнение при выходе из GET-объекта после нажатия клавиши <Enter> либо при выходе из него любым способом после изменения его значения. Строка с именем *FUNIQ* и параметром "U" должна следовать в *SIUEK.DBF* непосредственно за строкой с описанием проверяемого поля. В случае попытки ввода в это GET-объект этого поля значения, которое уже есть в файле, выдается соответствующее сообщение и предлагается повторить ввод.

Проверяемое поле может быть любого типа, а его символом управления (поле *SI70* - 'Упр') может быть элемент из множества {Q, K, G}. Файл должен быть обязательно индексирован по этому полю.

Пример вызова функции *FUNIQ*, имеющей тип "U", приведен в таблице А.38.

Таблица А.38. Описание вызова функции *FUNIQ* в файле *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коор Y	Коор X	Файл	Поле	Y поля	X поля	Пои ск	Тип	Дли на	Точ ность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907v	UF100001	Справочник БАНКОВ	Z	2	28			0	0	0		0	0
9085	UF100001	Название	G	4	3	<i>SFBN</i>	<i>SF11</i>	4	14	1	C	62	0
908F	UF100001	<i>FUNIQ</i>	U	0	0			0	0	0		0	0

Наличие функции *FUNIQ* после поля 'Название' гарантирует, что пользователь не сможет ввести в систему информацию о банках, имеющих одинаковое название.

Эта же функция может использоваться и для проверки реляционного ключа части.

Правила описания вызова *FUNIQ* отличаются от стандарта вызова других функций и рассмотрены в [9] и Help системы.

А.8. Дополнительные служебные признаки

L - признак фиктивной части (см. описание в *SIAGF Приложение 2*).

I – (эль латинское строчное) служебный символ управления.

При наличии данного символа в многослойной части, в момент перехода на обработку следующей части документа в режиме создания задается вопрос о необходимости добавить слой в текущую часть.

I – строка, содержащая координаты вывода графических изображений (дополнительно).

T, t – служебные символы управления, используемые системой для обработки реляционных ключей (см. [9], Приложение 1, Ограничения целостности, поддерживаемые функцией *FUNIQ*).

z – служебный символ управления, используемый системой, при обработке H-связей, представленных одним блоком .

Приложение 2

Б. Структура управляющих системных таблиц

Б.1. Таблица *SISOA* (“Адреса подсистем”)

Содержимое таблицы используется для указания правил распределения информации по локальной вычислительной сети и правил совместного (синхронного) использования одной и той же информационной подсистемы на различных ЭВМ сети.

Все рабочие копии каждой подсистемы имеют собственную таблицу *SISOA*, в которой указывается:

а) местоположение (номера машин) собственных и разделяемых файлов всех подсистем, с которыми работает подсистема;

б) номер исполнительной машины, на которой запускается данная рабочая копия подсистемы (модуль *ORD.EXE*) и располагаются ее управляющие системные файлы. (Имя исполняемого модуля *ORD.EXE* определяется спецификациями поставки и, например, может быть *C.EXE*, *T.EXE* и др.)

Структура *SISOA* представлена в таблице Б.1, а пример ее содержимого, описывающего распределение подсистем по ЭВМ локальной сети, показан на рисунке Б.1.

Таблица Б.1. Структура *SISOA*

Название поля	Содержимое поля	Поле файла	Тип	Длина
Подсистема	Имя подсистемы	<i>SI20</i>	С	8
N ЭВМ	Номер ЭВМ, на которой хранятся собственные и разделяемые файлы подсистемы, указанной в поле <i>SI20</i>	<i>SI21</i>	С	2
Электронный диск	Имя электронного диска машины, где хранится подсистема, указанная в поле <i>SI20</i>	<i>SI28</i>	С	1
Синхронность	Признак параллельного использования собственных и разделяемых файлов подсистемы, имя которой указано в поле <i>SI20</i>	<i>SI30</i>	С	2
Имя машины	Символьное имя машины	<i>SIUU</i>	С	15

Последняя запись файла (с именем *SSSS*) всегда содержит номер машины, на которой расположен данный экземпляр файла *SISOA.DBF* (исполнительной машины), что используется системой для реализации контрольных функций (рис. Б.1). Удалять ее и запись с подсистемой *SUBU* из *SISOA.DBF* не допускается. Подсистема *SUBU* используется в *ORD* для контроля сетевого окружения системы.

Признак параллельного использования файлов подсистемы $\langle S_i \rangle$ (синхронной работы) проставляется в том случае, если к собственным файлам данной подсистемы, обращаются несколько исполнительных машин, на которых имеются рабочие копии подсистемы $\langle S_i \rangle$.

(При неаккуратном назначении имен файлов может возникнуть ситуация, когда к собственным файлам подсистемы $\langle S_i \rangle$ будет обращаться подсистема $\langle S_j \rangle$. Если эта рабочая копия подсистемы $\langle S_j \rangle$ находится на другой машине сети, то также возникает необходимость в простановке признака параллельной работы для подсистемы $\langle S_i \rangle$.)

Признак параллельной (синхронной) работы проставляется только на хост-машине подсистемы $\langle S_i \rangle$, т.е., на той машине, где расположена рабочая копия собственных и разделяемых файлов подсистемы $\langle S_i \rangle$, и только в *SISOA.DBF* этой подсистемы. Указание

данного признака в любом другом *SISOA.DBF* не окажет влияния на алгоритм открытия и использования файлов.

Подсистема	N ЭВМ	Элек. диск	Синхронность
SUBA	05	H	
SUBB	05	H	
SUBF	05	H	P
SUBG	05	H	
SUBM	05	H	
SUBO	01	E	
SUBS	05	H	
SUBT	08	F	
SUBU	05	H	
SUBY	05	H	ST
SUBJ	05	H	
SUBQ	05	H	
SUBD	05	H	
SSSS	05	H	

Рис. Б.1. Пример распределения подсистем по машинам сети

При наличии признака "P" в поле *SI30* для подсистемы $\langle S_i \rangle$ в управляющем системном файле *SISOA.DBF* подсистемы $\langle S_i \rangle$, расположенном на машине-хозяйке подсистемы $\langle S_i \rangle$, как разделяемые, так и собственные файлы этой подсистемы, будут обрабатываться в совместном режиме. Это разрешит работу с собственными файлами другим машинам сети. (Исключение составляют собственные файлы, второй символ в имени которых является символом подчеркивания)

Пусть в примере, показанном на рисунке. Б.1, файл *SISOA.DBF* расположен в директории *D:\ORD\SUBF\SIDBF*. В этом случае он является системным управляющим файлом подсистемы "F". Причем последняя запись файла указывает на то, что он расположен на машине с номером 05, а из второй записи следует, что эта машина является хост-машиной подсистемы "F" (так как собственные и разделяемые файлы подсистемы "F" расположены на этой же машине).

Система использует в своей работе виртуальный диск, который и указывается в файле *SISOA.DBF*. Если в качестве виртуального диска используется электронный диск, (что увеличивает производительность на 15-20%) , то в *CONFIG.SYS* каждой машины, необходимо внести строку, обеспечивающую загрузку драйвера *RAMDRIVE.SYS*, создающего электронный диск в памяти, данной ЭВМ.

Например:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 1200 512/E или
DEVICE=C:\WINDOWS\RAMDRIVE.SYS 1200 512/E.
```

Емкость электронного (виртуального) диска определяется объемом самого большого файла *SIUEK.DBF*, из всех подсистемы, используемых на данной машине.

Если *RAMDRIVE.SYS* не используется, то в качестве литеры диска следует указать имя диска, на котором находится *ORD* на текущей машине.

Внимание! На начальном этапе освоения системы рекомендуется в поле *SI30* (синхронность) записи файла *SISOA.DBF* для разрабатываемой подсистемы занести символы "ST", указывающие системе, что это режим обучения. В данном режиме система не использует синхронизацию информационных процессов в локальной сети и не формирует журнал информационной целостности для глобальных сетей. Этот режим также исключает Следующие возможности:

- тиражирование данных;
- распараллеливание информационных процессов, включая блокирование исполнения функций контроля целостности;
- исполнение функций пользователя на других машинах сети.

Если система *ORD* функционирует в однопользовательской среде, то указанные выше возможности блокируются по умолчанию.

Количество подсистем, входящих в систему *ORD*, принципиально не ограничено. Однако на начальном этапе рекомендуется в качестве символа подсистемы использовать следующие буквы латинского алфавита {A, B, C, D, E, F, G, H, J, K, L, M, N, O, P, R, T, Q, Y, W, V}.

При добавлении пользователем новой записи в *SISOA.DBF* (новой подсистемы) система автоматически подготовит всю необходимую информацию для работы новой подсистемы.

Б.2. Таблица *SIORD* (“Справочник файлов подсистемы”)

Таблица *SIORD*, имеющаяся в каждой подсистеме, содержит описание всех используемых в ней файлов и имеет структуру, представленную в таблице Б.2.

Таблица Б.2 Структура файла *SIORD.DBF*

Поле файла	Назначение	Тип	Длина
SI20	Имя файла	C	8
SI39	Номер поля (используется системой)	N	4
SI25	Название поля (содержательное)	C	25
SI26	Имя поля (в структуре файла)	C	4
SI56	Тип поля	C	1
SI57	Длина поля	N	2
SI76	Точность поля	N	1
SI58	Характеристика поля (семантический код)	N	2
SI77	Признак спец. Индекса	C	1

Файл	Назначение	Поле	Тип	Длина	Точность	Сем.код	Спец.код
SPV1	СН записи проводки	sPV1	C	4	0	20	
SPV1	СН проводки	sFA1	C	4	0	21	
SPV1	Дата	BG99	D	8	0	1	
SPV1	Счет дебета	SB35	C	2	0	21	
SPV1	Знач. клас-ра	ZB36	C	7	0	6	
SPV1	Текст дебета	SB01	C	35	0	1	
SPV1	Счет кредита	SB76	C	2	0	21	
SPV1	Знач. клас-ра	ZB76	C	7	0	6	
SPV1	Текст кредита	SB02	C	35	0	1	
SPV1	sFA1 + SB03	z201	C	1	0	5	
SPV1	Сумма проводки	SB03	N	15	2	1	
SPV1	Документ-оригинал	SB04	C	4	0	1	
SPV2	СН записи проводки	sPV2	C	4	0	20	
SPV2	СН проводки	sFA2	C	4	0	21	

Рис. Б.2. Пример описания файлов (таблиц) в *SIORD*

Примечание. Нажатие клавиши F1-Справка позволяет получить необходимую информацию по любой функциональной клавише.

При первом запуске Db.exe система устанавливает маркер курсора на таблицу (файл) с именем СИМ, чтобы предоставить пользователю наиболее короткий способ перехода к часто используемым сервисным функциям системы (см. например, рис. 5).

Если маркер курсора находится не на описании таблицы СИМ открываются возможности, которые можно увидеть при нажатии клавиши F1.

Пример описания в *SIORD* приведен на рисунке Б.2. На этом рисунке показано описание файла *SFVI.DBF*, содержащего 12 полей, и первые два поля из описания файла *SFA1.DBF*. Количество полей в используемых файлах не имеет ограничений со стороны системы *ORD*.

Содержимое *SIORD* может заполняться ручным способом или автоматически: в момент визуального построения документа. Для имеющегося на диске файла система также может автоматически составить его описание в *SIORD*, если оно там отсутствует. Во всех случаях первое поле файла должно иметь статус собственного якоря.

Б.2.1. Правила описания логической структуры файла в *SIORD*

На экране утилиты *DB.EXE* физическим полям файла *SIORD.DBF* соответствуют следующие имена:

<i>SI20</i> – Файл	(Имя файла - начинается с латинского символа)
<i>SI25</i> – Назначение поля	(Смысловая нагрузка поля в файле)
<i>SI26</i> – Поле	(Имя поля в файле - начинается с латинского символа)
<i>SI56</i> – Тип	(Тип поля в файле - любой допустимый в языке C)
<i>SI57</i> – Длина	(Длина поля в файле - любая допустимая в языке C)
<i>SI76</i> – Точность	(Точность для цифрового поля, допустимая в языке C)
<i>SI58</i> – Сем. код	(Семантический код, допустимый для <i>ORD</i>)
<i>SI77</i> – Спец. индекс	(Поле имеет индекс специальной конструкции. Это не обычный составной индекс, а, именно, специальный)

Описание файла в таблице *SIORD* должно соответствовать следующим требованиям:

1. Все записи, относящиеся к описанию одного файла, должны располагаться в *SIORD* последовательно друг за другом и содержать в поле *SI20* имя описываемого файла (пример описания файлов в *SIORD* приведен выше). Чередувание с записями других файлов и пустыми записями не допускается.

2. В поле *SI26* ("Поле") первой записи для файла должно быть внесено имя собственного якоря этого файла.

3. Если файл имеет индекс по некоторому полю, то в записи, соответствующей данному полю, его имя в *SI26* должно начинаться со строчной (маленькой) буквы.

4. Если поле файла является полем классификатора, то в записи, соответствующей данному полю, вторая буква имени в *SI26* должна быть строчной.

5. Для всех полей (атрибутов) описываемого файла в полях *SI56*, *SI57* и *SI76* проставляются их обычные характеристики: тип (N, C, D, L, M), длина и точность, соответственно.

6. В поле *SI58* для каждого атрибута проставляется его системная характеристика, значения которой следующие:

1 - атрибут СОБСТВЕННЫЙ, если атрибут появляется в системе только в связи с данным документом, файлом, объектом. Собственный атрибут характеризует свойство данного объекта и принадлежит только этому объекту;

2 - атрибут **ЯКОРНЫЙ**, если значения данного атрибута уникально идентифицируют сущность (объект) хранимую в системе;

3 - атрибут **СИСТЕМНЫЙ**, если его значение формируется системой;

4 - атрибут **ВНЕСЕННЫЙ**, т.е. этот атрибут вносится в данный документ (файл, объект) из другого документа (файла, объекта, части документа), где он может быть собственным, якорным, системным без учета количественных характеристик связей.

Если значения, вносимые в данный атрибут, зависят от контекста каких либо других атрибутов объекта и требуют контроля Системы, то атрибут получает статус **ВНЕСЕННЫЙ-СОГЛАСОВАННЫЙ** и его семантический код должен быть больше 40;

5 - атрибут **ВИРТУАЛЬНЫЙ**, если его значение или имя вычисляется на основании алгоритма, но не хранится ни в одном файле системы. Частным случаем **ВИРТУАЛЬНОГО** значения являются составные индексы.

Например, на рисунке Б.2 для файла *SFV1.DBF* определен составной индекс по полям *SFA1+SB03*. Имя составного индекса *SFVIZZ01.NTX*. Семантический код для поля *zZ01* равный 5, указывает системе, что имя такого поля может отсутствовать в физической структуре файла, а значения постоянного индекса по полю *zZ01* будут автоматически отслеживаться системой.

6 - атрибут **ВЫЧИСЛЯЕМЫЙ**, т.е. атрибут значения которого формируются с помощью алгоритма.

7 - атрибут **КЛЮЧЕВОЙ** - это атрибут или группа атрибутов, семантически уникально идентифицирующая отношение модели, а, следовательно, кортеж отношения;

8 - атрибут, имеющий предысторию;

9 - атрибут, имеющий множество значений.

10 - атрибут части типа "Е" (в части этого типа могут быть несколько атрибутов с семантическим кодом 21 и только один с кодом 10)

Некоторые атрибуты могут обладать сочетанием системных характеристик, для которых используются значения с 11 по 99. В частности, Системные характеристики со значениями с 11 по 20 указывают на особенности атрибутов, являющихся собственными свойствами данного объекта (файла). Характеристики со значениями с 21 по 30 описывают особенности атрибутов, характеризующих данный объект, но являющихся собственными свойствами других объектов, т.е., “внесенных” атрибутов. Характеристики со значениями с 31 по 50, также относятся к внесенным атрибутам, но имеющим более сложный характер взаимосвязи с данным объектом.

Перечисленные характеристики обсуждаются более подробно в соответствующих разделах настоящего описания. Но наиболее важную роль среди них играют характеристики со значениями 20 (собственный якорный системный атрибут) и 21 (внесенный якорный системный атрибут), с помощью которых и устанавливаются структурные основные связи объектов (или файлов) документа. Эти связи обсуждались в п. Приложении 1 и были представлены на рис. 3.1.

Внимание! В настоящей версии системы *ORD* обязательно проставление характеристик 20, 21, 44, 5, 6 и 10 для соответствующих типов атрибутов, во всех остальных случаях следует указать значение характеристики, равное 1.

Поле *SI77* может содержать признак "Y", если подключен драйвер многоаспектного поиска.

Б.3. Таблица *SIDO* (“Атрибуты документов”)

Таблица *SIDO*, структура которой представлена в таблице Б.3, содержит списки атрибутов, входящих в состав документов каждого типа и используется генератором

незапланированных запросов. Вызов генератора (QBE) производится из пункта меню “Сервис” (рис. 2).

Заполнение файла, обычно, производится автоматически из меню редактирования таблицы *SIIM*.

Таблица Б.3. Структура файла *SIDO.DBF*

Поле файла	Название поля	Содержимое поля	Тип	Длина
<i>SIDO</i>	СН	Системный номер записи об атрибуте	С	4
<i>SIIM</i>	Документ	Тип документа, которому принадлежит описываемый атрибут	С	4
<i>SI27</i>		Зарезервировано	С	1
<i>SI29</i>		Зарезервировано	N	2
<i>SI25</i>	Атрибут	Наименование атрибута в документе (аналогично <i>SIORD.DBF</i>)	С	40
<i>SI51</i>	Обозначение	Сокращенное наименование атрибута	С	10
<i>SI26</i>	Поле	Имя поля, в котором хранится значение атрибута (поля атрибута)	С	4
<i>SI20</i>	Файл	Имя файла, в котором хранится атрибут	С	8
<i>SI70</i>	Управление	Тип атрибута, как элемента экранного образа документа	С	1
<i>SN51</i>	Смещение	Смещение атрибута в файле <i>SIUEK.DBF</i> по отношению к началу образа (в случае, если в <i>SI70</i> указан <i>Q</i> или <i>K</i> тип атрибута)	N	4
<i>SI56</i>	Тип	Тип атрибута и поля атрибута	С	1
<i>SI57</i>	Длина	Длина поля атрибута	N	2
<i>SI76</i>	Точность	Точность поля атрибута	N	1
<i>SI81</i>	Повтор	Порядковый номер вхождения поля в документ, для полей с одинаковым именем, входящих в документ несколько раз	N	1
<i>SI_2</i>	Поле классиф.	Имя поля, в котором записано значение прокрутки атрибутов типа <i>Q</i> или <i>K</i>	С	4
<i>SI_3</i>	Тип классиф.	Тип поля значения прокрутки <i>Q</i> - или <i>K</i> -атрибутов	С	1
<i>SI_4</i>	Длина классиф.	Длина поля значения прокрутки <i>Q</i> - или <i>K</i> -атрибутов	N	2
<i>SI_5</i>	Точность классиф.	Точность поля значения прокрутки <i>Q</i> - или <i>K</i> -атрибутов	N	1

Поле *SI25* должно содержать уникальную информацию, позволяющую понять назначение этого атрибута во всей системе, так как данная информация используется генератором незапланированных запросов для выбора атрибутов поиска.

Содержимое файла будет заполнено системой, если, находясь в окне прокрутки таблицы *SIIM* (“Имена документов подсистемы”) утилиты *DB.EXE*, кликнуть мышкой на кнопке “Заполнить *SIDO*” или нажать клавишу F11.

Б.4. Таблица *SIIM* (“Список типов документов подсистемы”)

Таблица *SIIM* сопоставляет каждому документу подсистемы его имя. Таблица имеется в каждом экземпляре подсистемы, но тождество этих таблиц в разных экземплярах системы

не обязательно. Обычно, эта таблица на хост-ЭВМ подсистемы должна включать все документы, а ЭВМ с остальными (клиентскими) экземплярами подсистемы могут содержать только часть документов. Структура таблицы *SIIM* представлена в таблице Б.4.

Таблица Б.4. Структура файла *SIIM.DBF*

Поле файла	Название поля	Содержимое поля	Тип	Длина
<i>SIIM</i>	Номер	Системный номер типа документа TDOK	C	4
<i>SI50</i>	Имя документа	Имя типа документа	C	45
<i>SI54</i>	Служебное	Шифр типа документа или иная информация	C	6
<i>SSRT</i>	Использование в реестре	Имена полей данного типа документа, используемых для бухгалтерского учета	C	60
<i>SI55</i>	Цвет	Код цветовой гаммы (для DOS версии)	C	1
<i>SI1M</i>	Служебное	Используется системой при архивации документов этого типа	C	1
<i>SI2M</i>	Служебное	Параметры для архива	C	1
<i>SI3M</i>	Служебное	Параметры для архива	C	8
<i>SI4M</i>	Служебное	Параметры для архива	C	4
<i>SI5M</i>	Служебное	Параметры для архива	C	4
<i>SINE</i>	Служебное	Используется системой при передаче документов этого типа в другую сеть	C	1
<i>SIXO</i>	Служебное	Используется системой	C	1
<i>SIVO</i>	Служебное	Используется системой	C	1
<i>SIZ1</i>	Служебное	Используется системой	C	1

Добавление новой записи в данный файл является отправной точкой для создания нового понятия (документа) в системе при автоматизированном его построении с помощью утилиты *DB.EXE*.

Поле *SIIM* (совпадает с именем файла) предназначено для хранения системного номера типа документа (номер уникален во всей системе и определяется администратором системы). После определения нового типа документа, он получает статус **понятия** системы. (Значение этого поля в момент работы системы с документом копируется в глобальную переменную *TDOK*.)

Первый символ в любом номере документа равен "0" (нуль). Наличие любого другого символа в данной позиции превращает этот документ в невидимый для системы ("комментирует" документ).

Поле *SI50* (Имя документа) предназначено для хранения смыслового определения понятия (типа документа).

Поле *SI55* (Цветовая гамма документа DOS версии) предназначено для хранения признака цветовой гаммы документа. Каждый тип документа (по желанию пользователя) может использовать свою цветовую гамму. По умолчанию система использует стандартную гамму (признак "1"). Режим установки цветовой гаммы находится в головном меню подсистемы или встроен в *DB.EXE* (зависит от версии системы *ORD*).

В Windows версии установки цвета производятся из меню приложения и сохраняются в файле *ORD.INI* или в реестре Windows.

Поле *SI54* (Шифр типа документа) предназначено для пользователя и к значениям, находящимся в этом поле, система безразлична.

Исключение для документа с системным номером "0P00" - Здесь прописан код операционной системы, в которой работает *ORD*.

Принята следующая кодировка:

Win9x - 95,98 и для Dos;

Win2k - 2000;

WinXp;
WinNT.

Поле SSRT используется, если в процессе проектирования информационной модели предметной области, проектировщик использовал типовой фрейм "Организация" и сопутствующие ему понятия, связанные с передачей материальных ценностей или денежных средств между организациями.

В этом случае поле SSRT используется для учета взаиморасчетов между организациями и заполняется разработчиком подсистемы для тех документов, которые отображают передачу материальных ценностей или денежных средств между организациями. Основная информация таких документов: тип документа, номер, дата, организация, сумма, фиксируется в специальных файлах сводок документов, имеющих в каждой подсистеме.

Поле SSRT (длиной 60 символов) рассматривается как состоящее из 5 частей по 12 символов. Каждая часть содержит имя поля (первые 4 символа) и имя файла, содержащего это поле (следующие 8 символов, из которых до четырех правых символов может быть пробелами).

Первая часть определяет регистрационный номер документа, вторая – дату его оформления, третья – учетную (итоговую) сумму, четвертая – организацию-дебитора (получающую деньги/товар по документу) и пятая – организацию-кредитора (отдающую деньги/товар).

Во всех других случаях поле не заполняется.

Исключение для документа с системным номером "0P00" - Здесь прописан путь текстового редактора, с которым работает пользователь ORD.

Служебные поля *SI1M*÷*SI5M* предназначены для использования системной архивации документов и учета изменения структуры документа в процессе его эволюции. Состав и назначение этих полей рассмотрено в Help разделе, описывающем процесс архивации и работу с архивом.

Поле *SINE* предназначено для указания возможности передачи данного документа в другую (географически удаленную) локальную сеть. Содержимое поля генерируется автоматически, если администратор системы разрешает передачу документа в другую сеть. (Если документ разрешен для передачи в другую сеть, то значение поля *SINE* = "Y". См. подробную информацию в Help системы)

Поля *SIXO*, *SIVO*, *SIZI* – предназначены для указания правил обработки документа в данной точке (локальной сети) присланного из другой локальной (географически удаленной) сети, отвечающей за адекватность содержимого документа.

Имя точки (локальной сети), указывается в служебном файле *SSID*, который должен присутствовать в каждой локальной сети, обслуживаемой системой *ORD*. (См. подробную информацию в Help системы.)

Прямое использование системной таблицы *SIIM* в прикладных программах пользователя запрещено, поэтому для каждой подсистемы допускается иметь 3 файла дублирующих (возможно, частично) его содержимое.

Имена таких файлов для подсистемы "x" могут быть *S<x>IM*, *<x>S_1* и *<x>S_2*, т.е., например, для подсистемы "F" это файлы *SFIM.DBF*, *FS_1.DBF* и *FS_2.DBF*. Причем *SFIM.DBF* представляет собой список типов документов подсистемы "F", используемых в реестре подсистемы "F" для бухгалтерского учета взаиморасчетов между организациями. Структура этого файла совпадает со структурой файла *SIIM.DBF*.

Файл *FS_1.DBF* является дублем файла *SIIM.DBF* с полями *FS_1*, *SI_1*, *SI49*,..., где *FS_1* и *SI_1* эквиваленты поля *SIIM*, а *SI49* – эквивалент поля *SI50*. Аналогично, файл *FS_2.DBF* тоже является дублем файла *SIIM.DBF* с полями *FS_2*, *SI_2*, *SI48*,..., где *FS_2* и *SI_2* эквиваленты поля *SIIM*, а *SI48* – эквивалент поля *SI50*.

Пример экранного изображения файла *SIIM.DBF* приведен на рисунке Б.3.

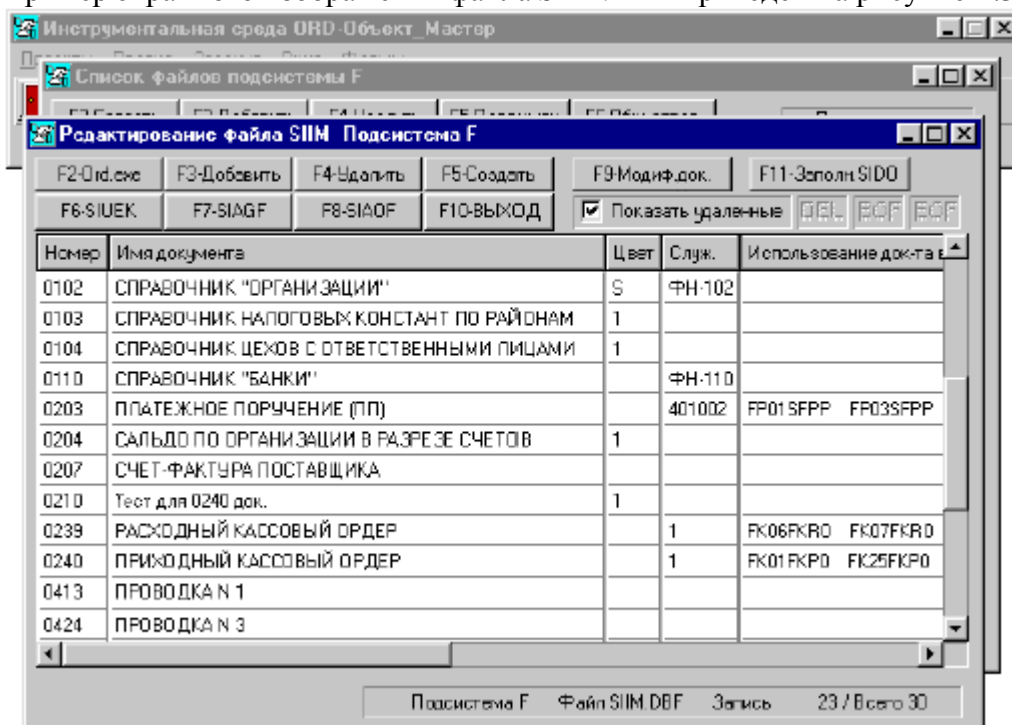


Рис. Б3. Состав таблицы *SIIM*

Назначение кнопок управления, выведенных на окно прокрутки файла *SIIM.DBF*, и соответствующих им функциональных клавиш следующее:

<F2> - запуск исполняемого модуля *ORD.EXE* для документов подсистемы, т.е., возможность обработки существующего документа в среде *ORD*;

<F3> - добавление в конец файла пустой записи, т.е., добавление нового документа и заполнение его характеристик;

<F4> - удаление выбранной записи, при этом происходит физическое удаление сопутствующей информации из всех системных файлов, описывающих данный документ;

<F5> - если документ новый, то переход в режим автоматического визуального создания документа (непосредственно с экрана);

<F6> - если документ старый, то переход в режим просмотра или модификации экранного образа документа (*SIUEK*);

<F7> - если документ старый, то переход в режим просмотра или модификации логической структуры документа (*SIAGF*);

<F8> - если документ старый, то переход в режим просмотра или модификации списка файлов, использованных в образе документа (*SIAOF*);

<F9> - если документ старый, то переход в режим просмотра или визуальной модификации образа документа (модификация всех моделей документа с соблюдением принципа логической независимости данных, т.е., расширение существующих структур данных или их модификация без перезаписи созданных и эксплуатируемых прикладных программ);

<F10> - выход в основной режим (редактирования *SIORD*) или переход на другую подсистему;

<F11> - подготовка документа для работы в режиме незапланированных запросов.

При использовании ручного режима создания документа рекомендуется заполнить системные таблицы в следующем порядке: *SIIM*, *SIORD*, *SIUEK*, *SIAGF*, *SIAOF* затем создать эталон документа и проверить его, вызвав *ORD.EXE*

Примечание. Таблица *SIIM* подсистемы "S" имеет другую структуру и предназначена для хранения базы фреймов системы *ORD*. Его пополнения или модификации возможны, если администратор системы наделен особыми правами, зафиксированными в Лицензии системы.

В.5. Таблица *SIUEK* (“Экранные образы документов”)

Таблица *SIUEK* содержит описание экранных образов типов документов, используемых в подсистеме, и имеет структуру, показанную в таблице Б.5.

Таблица Б.5. Структура файла *SIUEK.DBF*

Поле файла	Название поля	Назначение поля	Тип	Длина
SI75	Сист.	Системный номер записи	C	4
SI52	N образа	Имя части документа	C	8
SI65	Имя атрибута на экране	Имя атрибута документа, изображаемое в его экранном образе	C	25
SI70	Упр.	Признак управления (тип представления поля на экране)	C	1
SI66	Коорд. X	(Строковая)X-координата вывода наименования атрибута на экран	N	2
SI67	Коорд. Y	Y-координата вывода наименования атрибута на экран (в знакоместах экрана 80 × 25)	N	2
SI20	Файл	Имя файла, которому принадлежит поле атрибута	C	8
SI26	Поле	Имя поля, в котором хранится атрибут	C	4
SI68	Коорд. Y поля	(Строковая)X-координата вывода значения атрибута на экран	N	2
SI69	Столбец	Y-координата вывода значения атрибута на экран (в знакоместах экрана 80 × 25)	N	2
SI72	Поиск	Порядковый номер столбца атрибута в окне прокрутки атрибутов при выборе документа	N	2
SI74		Резервное. На экране не показывается	N	1
SI56	Тип поля	Тип поля атрибута	C	1
SI57	Длина	Длина поля атрибута	N	2
SI76	Точность	Точность поля атрибута	N	1
SI78	Функция шаблона	Функция шаблона	C	4
SI79	Цвет	Цвет (для DOS-версии)	C	1
SI73	N	Порядок атрибутов прокрутки H-классификатора	N	2
SI80		Резервное. На экране не показывается	N	1
SI81		Резервное. На экране не показывается	N	1
SC75		Служебное. На экране не показывается	C	4
SI40		Служебное	N	2
SI41		Служебное	C	5

Примечание. Содержимое файла заполняется автоматически, если документ подготавливается визуальными средствами системы.

Столбцы таблицы *SIUEK* имеют следующее назначение:

SI75 – (Сист.). Здесь и далее в скобках указан эквивалент имени поля при изображении его на экране утилитой *DB.EXE*. Системный номер записи таблицы.

Заполняется системой автоматически по возрастанию в момент добавления записи в таблицу.

Если строка (запись файла) *SIUEK* формируется вручную и система выдает сообщение "Исчерпаны свободные адреса", рекомендуется войти в режим визуальной модификации документа и выйти с сохранением (F2). Тогда диапазон свободных адресов будет увеличен автоматически.

Значение поля при добавлении в *SIUEK* новой записи формируется увеличением значения поля в записи, на которой установлен указатель.

SI52 – (N образа). Имя части документа. Длина поля 8 символов:

- первый символ U - любой другой воспринимается как комментарий;
- 2-ой символ - имя текущей подсистемы;
- 3-5 символы - номер типа документа из *SIIM*;
- 6-8 символы - номер части экранного образа.

Например, UY327002:

- документ относится к подсистеме Y;
- тип документа *SIIM* = "0327" (первый символ из номера документа не учитывается);
- номер части экранного образа 002 (т.е. описывается вторая часть).

Поле заполняется системой автоматически при визуальном создании документа.

Пример строки (записи файла) *SIUEK*, описывающей один атрибут экранного образа, приведен в таблице Б.6 (три последних столбца: поля *SI78*, *SI79*, *SI73*, – не показаны).

Таблица Б.6. Содержимое строки (записи файла) *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Коорд Y поля	Коорд. X поля	Поиск	Тип	Дли на	Точ- ность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31GH	UY327002	Расход на производство	G	14	22	YRSV	YR16	14	47	0	N	11	2

SI65 – (Имя атрибута на экране). Имя атрибута с координатами указанными в полях *SI66*, *SI67*.

Внимание! Первым атрибутом в экранном образе документа всегда должен стоять собственный якорь основного файла документа (части) с символом управления (поле *SI70*) "M". Пример такой строки приведен в таблице Б.7 (три последних столбца не показаны).

Таблица Б.7. Содержимое первой строки (записи файла) экранного образа в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
907I	UY327002	Системный код банка	M	0	0	SFBN	SfBN	0	0	0	C	4	0

Следует обратить внимание, что служебная информация не выводится на экран (система определяет тип информации и что с ней можно делать по символу указанному в поле *SI70*). Блокировать вывод имени атрибута на экран можно также, указав для него нулевую координату Y.

Однако для записи *SIUEK*, показанной в табл. Б.6, в 14-й строке экрана, начиная с 22-й колонки, будет выведен текст "Расход на производство", а, начиная с 47-й колонки – значение данного атрибута.

Если длина имени атрибута превышает размер поля *SI65*, и существует необходимость, чтобы система воспринимала этот атрибут как единое целое, то продолжение имени атрибута следует писать в следующей строке (записи) файла *SIUEK*. Пример такого случая показан в таблице Б.8 (три последних столбца не показаны):

Таблица Б.8 Представление длинного имени атрибута в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31GH	UY327002	Расход на производство	G	14	22	YRSV	YR16	14	58	9	N	11	2
31GK	UY327002	в день	Z	14	48			0	0	0		0	0

Примечание. В некоторых случаях содержимое поля *SI65* используется для указания имен исполняемых функций пользователя и самой системы. Например, символ "*", расположенный в первой позиции поля *SI65*, указывает на структурную связь; слово *TIME* - на согласование структурной связи во времени и т.д.

Примеры описания функций пользователя и системы представлены в соответствующих разделах.

SI70 – (Упр.) Признак управления, который может принимать одно из следующих значений: X, ?, Z, z, M, N, F, J, K, k, Q, q, R, r, H, P, A, D, I, G, S, s, W, Y, U, L, l, T, t.

Эти признаки определяют характер элемента экранного образа описываемого данной строкой файла *SIUEK.DBF*. Полное описание правил задания этих признаков приведено в Приложении 1

Выше уже встречалось упоминание в тексте о том, что значения

- M и N указывают на то, что атрибуты являются якорными и используются для связи частей документа в единое целое;
- G и S задают обычный GET-объект, который для S является не редактируемым;
- K, k, Q, q и H говорят о том, что значение поля данного атрибута выбирается из классификатора;
- A, P, W и Y определяют различные виды процедур пользователя, исполняемых в момент навигации по экранному образу или при записи документа;
- R и r задают изображение рамки на экране;
- Z указывает на вывод текста из поля имени атрибута;
- F и J определяют правила копирования значений одних атрибутов в другие;
- X описывает невидимый на экране атрибут или указывает на различные служебные функции данной строки *SIUEK* и т.п.

SI66 – (Координата Y). Строка вывода на экран записанного в поле *SI65* имени атрибута.

Примечание. Наличие 0 в координате Y запрещает вывод имени атрибута на экран.

SI67 – (Координата X) Столбец вывода на экран записанного в поле *SI65* имени атрибута.

SI20 – (Файл) Имя файла, в котором хранится значение атрибута.

В некоторых случаях это поле используется для служебных целей.

SI26 – (Поле) Имя поля, в котором хранится значение атрибута (для краткости называемое далее *полем атрибута*)

SI68 – (Координата Y поля) Строка вывода GET-объекта для поля атрибута.

Примечание. Наличие 0 в координате Y запрещает вывод информации на экран.

SI69 – (Столбец) Столбец вывода GET-объекта для поля атрибута.

Например, в результате обработки системой строки (записи) *SIUEK*, показанной в таблице Б.9 текст "МФО р/счет" будет выведен в седьмой строке, начиная с третьей позиции, а содержимое поля *SF19* файла *SFBN.DBF* в восьмой строке, начиная с пятнадцатой позиции. (При создании нового экземпляра документа, до записи его на диск, в этом GET-объекте, конечно, будет введенная пользователем информация, а не данные из файла).

Таблица Б.9. Содержимое первой строки (записи файла) экранного образа в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
908Z	UF1100012	МФО р/счет	G	7	3	SFBN	SF198	15	2	C	9	0	

При этом наличие в поле *SI72* (поиск) числа 2 указывает системе, что значение поля используется пользователем и системой для различия одного экземпляра документа от другого при выборе/поиске экземпляра документа.

SI72 – (Поиск). Порядковый номер поискового атрибута, указывающий, в каком по счету столбце будет расположен атрибут в окне прокрутки при выборе экземпляра документа.

Атрибуты образа документа, записанные в поле *SI65*, которые предполагается использовать для поиска, следует пронумеровать в поле *SI72*, так, чтобы выполнялись следующие правила:

- 1) нумерация атрибутов должна начинаться с единицы;
- 2) первые номера (не более 97) получают атрибуты, по которым должен осуществляться поиск экземпляра документа;
- 3) поисковые атрибуты следует нумеровать так, чтобы меньший номер присваивался атрибуту, более информативному для поиска документа;
- 4) для всех остальных атрибутов, в том числе, служебных в этом поле следует проставить 0.

Поиск экземпляра документа будет осуществляться по пронумерованным атрибутам.

Таблица Б.9 Указание поисковых атрибутов в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>
31GH	UY327002	Расход на производство	G	14	22	YRSV	YR16	14	32	1	N	11	2
31GK	UY327002	Материал	K	14	48	YRVS	UMCL	14	58	2	C	10	0

Номера, указанные в поле *SI72* в этом примере, определяют, что при выводе объекта "0327" в окно прокрутки для выбора экземпляра документа в первом столбце будут показаны значения атрибута "Расход на производство", а во втором – значения атрибута "Материал".

Внимание! Не следует отождествлять набор атрибутов, используемых для поиска, с множеством атрибутов, выделяемых в качестве реляционного ключа.

Поисковые атрибуты могут распределяться по любым объектам документа. Расположение поисковых атрибутов не имеет связи с полем *SI73* – порядок атрибутов в модельном классификаторе Н.

SI56 – (Тип) Тип поля атрибута.

SI57 – (Длина) Длина поля атрибута

SI76 – (Точность) Точность поля атрибута.

Поля *SI56*, *SI57*, *SI76* заполняются системой автоматически, если структура файла, указанного в поле *SI20*, описана в таблице *SIORD*.

SI78 – (Функция шаблона). Если поле атрибута имеет специфические правила заполнения информацией, то в данном поле проставляется один из признаков, перечисленных в таблице Б.10:

Таблица Б.10. Шаблоны вывода значения атрибута, указываемые в *SIUEK.DBF*

Функция	Тип	Действие
A	C	Разрешает только буквенные символы (латинские)
B	N	Выводит числа, выровненные по левому краю
Z	N	Выводит вместо нулей пробелы
(N	Выводит отрицательные числа, заключенные в круглые скобки с пробелами вместо ведущих нулей
)	N	Выводит отрицательные числа, заключенные в круглые скобки без ведущих пробелов
!	C	Задаёт вывод символов в верхнем регистре

Пример. Пусть в поле атрибута разрешено использовать только прописные символы (латинские и русские), тогда в поле *SI78* надо указать символ “!”.

В этом случае запись в *SIUEK* может выглядеть так, как показано в таблице Б.11, что разрешит ввод в поле атрибута только символов верхнего регистра.

Таблица Б.11. Использование функции шаблона в экранном образе в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность	Функ шабл
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>	<i>SI78</i>
908Z	UF1100012	Шифр	G	7	3	SFBN	SF198	15	2	C	9	0		!

Шаблон может быть и составным, как показано в таблице Б.12. В этом случае разрешен ввод в поле атрибута только символов верхнего регистра, обрамленных точками, которые будут видимы на экране (например, *.ORD.*) и будут записываться в файл.

Таблица Б.12. Использование функции шаблона в экранном образе в *SIUEK.DBF*

Сист.	N образа	Имя атрибута на экране	Упр	Коорд Y	Коорд X	Файл	Поле	Y поля	X поля	Поиск	Тип	Длина	Точность	Функ шабл
<i>SI75</i>	<i>SI52</i>	<i>SI65</i>	<i>SI70</i>	<i>SI66</i>	<i>SI67</i>	<i>SI20</i>	<i>SI26</i>	<i>SI68</i>	<i>SI69</i>	<i>SI72</i>	<i>SI56</i>	<i>SI57</i>	<i>SI76</i>	<i>SI78</i>
908Z	UF1100012	Шифр	G	7	3	SFBN	SF198	15	2	C	9	0		!.

Следует отметить, что подобный или любой другой контроль может быть осуществлен и функцией пользователя (см. описание назначения поля *SI70* – Признак управления).

SI79 - (Цвет) Используется для выделения цветом имени атрибута или иного текста, записанного в поле *SI65*, и может содержать один из трех символов "Y", "P" и "R", значение которых следующее:

Y – цвет имени атрибута или текста должен совпадать с цветом атрибутов *K*- и *Q*-классификаторов

P – цвет имени атрибута или текста должен совпадать с цветом атрибута *P*-процедуры

R – цвет имени атрибута или текста должен совпадать с цветом рамки *H*-классификатора

SI73 - (H) Это поле задает поисковых атрибутов для *H*-классификатора в том случае, если при выборе значений из него значений возникает необходимость изменить порядок их показа по сравнению с порядком атрибутов в экранном образе *H*. Тогда в этом поле проставляются поисковые атрибуты по правилам, совпадающим с правилами заполнения поля *SI72* (Поиск). Обычно, это поле заполняют, если несколько *H*-классификаторов объединены одной рамкой.

Замечание. При построении экранного образа документа следует учитывать, что образ документа может состоять из нескольких частей (см. примеры учебных документов в ЧАСТИ 2). Допускается построение многостраничных экранных образов документов.

Пример экранного отображения таблицы *SIUEK* приведен на рисунке Б.4 (из-за ограничений формата окна прокрутки три последних столбца не показаны).

Сист.	N образа	Имя атрибута на экране	Упр	КоэфУ	КоэфХ	Файл	Поле	Упола	Глоб.	Понск	Тип	Длина	Точн.	
91Fm	UF110001	Системный номер банка	M	0	0	SFBN	sfBN	0	0	0		0	0	
91Fz	UF110001	СПРАВОЧНИК БАНКОВ	Z	2	28			0	0	0		0	0	
91Fu	UF110001	ICONSBN	W	0	0			0	0	0		0	0	
91Q!	UF110001	Название	G	4	3	SFBN	sf11	4	14	1	C	62	0	
91Q'	UF110001	ИРедBan	Y	0	0			0	0	0		0	0	
91Q-	UF110001	Почтовый индекс банка	G	6	3	SFBN	SF02	6	28	0	C	6	0	
91Q3	UF110001	БИК	G	8	3	SFBN	SF19	8	15	2	C	10	0	
91Q9	UF110001	FUNIQ	U	0	0			0	0	0		0	0	
91Q?	UF110001	МФО к/счет	G	8	28	SFBN	SF13	8	40	0	C	9	0	
91QE	UF110001	Местоположение банка 1	K	10	3	SFBN	sf12	10	25	0	C	11	0	
91QK	UF110001	Город	2	K	0	51	SFBN	sf03	10	58	3	C	15	0
91QQ	UF110001	Улица	G	12	3	SFBN	SF04	12	10	0	C	25	0	
91QU	UF110001	Дом	G	12	37	SFBN	SF05	12	41	0	C	8	0	
91QJ	UF110001	N счета	G	15	3	SFBN	SF09	15	28	0	C	20	0	

Рис. Б.4. Пример содержимого таблицы *SIUEK*

Если утилитой *DB.EXE* открыт экранный образ документа в *SIUEK*, то пользователю доступны следующие режимы:

<F1> – получение справки по файлу *SIUEK*;

<F2> – переключение режимов видимости записей, помеченных на удаление:

режим DEL ON - помеченные на удаление записи не видны,

режим DEL OFF - помеченные на удаление записи видны, но не доступны для редактирования; однако высвечиваемое количество записей файла не меняется (в него включены помеченные на удаление);

<F3> – добавление пустой записи в файл.

Если указатель стоит в конце файла, то осуществляется запись в поле *SI75* следующего системного номера, а в поле *SI52* текущего номера экранного образа.

Если указатель стоит не на конце файла, то система вставит запись вслед за текущей, прописав в нее следующий системный номер (если есть свободный!) и номер предыдущего экранного образа.

Если в процессе выполнения операции добавления прозвучит звуковой сигнал, то система выдаст соответствующее предупреждение об исчерпанности адресного пространства части и заставит переместить информацию документа в свободную область *SIUEK* (Для этого можно использовать клавишу F5).

При использовании ручного способа создания образа документа, рекомендуется добавление записи о новом документе производить, находясь на конце файла (нажатие Ctrl/End или Ctrl/PgDn устанавливает указатель в файле на его конец).

<F4> – удаление выбранной записи. При этом происходит физическое удаление (с упаковкой). Если к моменту этой операции документ уже существовал и был описан в *SIAGF*, то данная операция (если это необходимо), отразится и в *SIAGF*;

<F5> – копирование экранного образа из любого другого документа любой подсистемы (или копирование всех моделей документа).

При нажатии <F5> появляется сообщение-инструкция, но здесь следует подчеркнуть некоторые особенности этой процедуры:

- записи *SIUEK*, предназначенные для копирования, изначально расположены в порядке возрастания СН (системных номеров), даже если отдельные записи (в режиме "КОПИРОВАНИЕ ЧАСТИ *SIUEK*") выбирались в произвольном порядке. Но, если при их редактировании были изменены поля *SI75*, то порядок записей, оставаясь неизменным, может перестать соответствовать порядку возрастания СН;

- как в режиме "КОПИРОВАНИЕ ЧАСТИ *SIUEK*", так и в режиме "КОПИРОВАНИЕ ДОКУМЕНТА" разработчику документа предоставляются возможности управления адресом (СН начала) документа или группы строк. СН записи может состоять из символов с кодами от 33 ("!") до 126 ("~"). При изменении начального адреса СН всех записей изменяются на одну и ту же величину. Это изменение затрагивает и ссылки на другие записи в поле *SI65*. При наличии в СН исходного документа символов вне указанного выше диапазона коррекция адресов может выполняться с ошибкой;

- в результате выполнения операции копирования всего документа автоматически заполняются все сопутствующие системные файлы.

<F7> – переход на описание логической структуры данного типа документа в файле *SIAGF*, если она определена в системе. Если переход сопровождается звуковым сигналом, это означает, что система оповещает об ошибочной ситуации.

<F8> – переход на описание списка файлов данного типа документа в файле *SIAOF.DBF*, если он определен в системе (аналогично <F7>)

<F9> – переход на запись о документе данного типа в файле *SIIM.DBF*, если он определен в системе (аналогично <F7>)

Как видно, клавиши <F7>, <F8>, <F9> - контекстно-зависимы (переход всегда осуществляется к записям системных файлов, принадлежащим рассматриваемому типу документа).

<F10> – выход в основной режим (редактирования *SIORD.DBF*)

<F12> – получение справки *ORD*.

<Alt/G> – переход по номеру записи;

<Backspace> – удаление левого символ этого ключа;

<Alt/Y> – сброс ключа поиска;

<Alt/Z> – удаление текущего столбца из окна прокрутки;

<Alt/A> – восстановление всех полей для ввода.

Примечание. В окне прокрутки файла *SIUEK.DBF* утилиты *DB.EXE* работает быстрый поиск по индексируемым полям (просто вводятся искомые значения этого поля), в верхней строке экрана виден ключ поиска.

Если система не реагирует на нажатие вышеуказанных функциональных клавиш, то это, скорее всего, означает, что была нажата клавиша <Enter> и система ждет ввода информации.

Если при нажатии функциональной клавиши система издает звуковой сигнал - это является предупреждением об ошибочной ситуации.

Б.6. Таблица *SIAOF* (“Списки файлов документов”)

Таблица *SIAOF*, структура которой представлена в таблице Б.13, содержит перечень файлов, используемых в документе. Файл хранится в каждой копии подсистемы. Используется для контроля правильности заполнения файлов, описывающих модели документа, при ручном режиме формирования этих моделей.

Таблица Б.13. Структура файла *SIAOF.DBF*

Поле файла	Название поля	Содержимое поля	Тип	Длина
<i>SIIM</i>	Номер	Системный номер типа документа	C	4
<i>sI20</i>	Файлы документа	Имя файла документа	C	8
<i>SI59</i>	Статус файла	Признак типа файла	C	16
<i>SI52</i>	Экран	Имена экранных образов	C	8
<i>SI60</i>	N части	Порядковый номер части	N	2

Пример экранного представления содержимого таблицы *SIAOF* утилитой *DB.EXE* приведен на рисунке Б.5.

Номер	Файлы документа	Статус файла	N образа	N Части
0110	SFBN	S	UF110001	1
0110	SFBP	S	UF110002	2
0110	SFBR	S	UF110003	3
0110	SFBN	Q	UF110003	3
0110	SF12	K	UF110001	1
0110	SF03	K	UF110001	1
0203	SFPF	O	UF203001	1
0203	SFP4	B	UF203001	1
0203	SFBC	H	UF203002	2
0203	SFBC	H	UF203004	4
0203	SFP1	E	UF203005	5
0203	SFP5	V	UF203008	8
0203	SFP2	V	UF203009	9
0203	SFOR	Q	UF203001	1

Подсистема F Файл SIAOF.DBF Запись 75 / Всего 221

Примечание. Части продолжения в файле *SIAOF.DBF* не описываются.

Правила заполнения полей *SI59*, *SI52*, *SI60* изложены ниже при рассмотрении файла *SIAGF.DBF*. Содержимое файла заполняется автоматически, если документ подготавливается визуальными средствами системы.

Если утилитой *DB.EXE* открыто окно прокрутки файла *SIAOF.DBF*, то пользователю доступны следующие действия:

<F1> – получение справки по файлу *SIAOF.DBF*;

<F2> – переключение режимов видимости записей, помеченных на удаление:

режим DEL ON - помеченные на удаление записи не видны,

режим DEL OFF - помеченные на удаление записи видны, но не доступны для редактирования; однако высвечиваемое количество записей файла не меняется (в него включены помеченные на удаление);

<F3> – добавление конец файла пустой записи для занесения в нее информации о новой части документа;

<F4> – удаление выбранной записи. При этом происходит физическое удаление (с упаковкой). Если к моменту этой операции документ уже существовал и был описан в файлах *SIAGF.DBF*, и *SIUEK.DBF*, то данная операция на их содержимом *не отразится*;

<F5> - очистка содержимого документа, если такой документ существует в системе;

<F6> – переход на описание экранного образа документов данного типа в файле *SIUEK.DBF*, если он определен в системе. Если переход сопровождается звуковым сигналом, это означает, что система оповещает об ошибочной ситуации.

<F7> – переход на описание логической структуры данного типа документа в файле *SIAGF.DBF*, если она определена в системе (аналогично <F6>).

<F8> – проверка согласованности статусов файлов. **Важнейший** режим работы, проверяющий правильность построения информационной модели предметной области. Использовать **только** в случае официальной передачи разработанной или модифицированной структуры документа в работу. **При использовании этого режима необходимо, чтобы все реальные системные (SI...) файлы всех подсистем находились на данной ЭВМ.**

<F9> – переход на запись о документе данного типа в файле *SIIM.DBF*, если он определен в системе (аналогично <F6>)

<F10> – выход в основной режим (редактирования *SIORD.DBF*)

<F11> - реиндексация файлов входящих в документ.

<F12> – получение справки *ORD*.

<Alt/G> – переход по номеру записи;

<Backspace> – удаление левого символ этого ключа;

<Alt/Y> – сброс ключа поиска;

<Alt/Z> – удаление текущего столбца из окна прокрутки;

<Alt/A> – восстановление всех полей для ввода.

Примечание. В окне прокрутки файла *SIAOF.DBF* утилиты *DB.EXE* работает быстрый поиск по индексируемым полям (просто вводятся искомые значения этого поля), в верхней строке экрана виден ключ поиска.

Если система не реагирует на нажатие указанных выше функциональных клавиш, то это, скорее всего, означает, что была нажата клавиша <Enter> и система ждет ввода информации.

Если при нажатии функциональной клавиши система издает звуковой сигнал - это является предупреждением об ошибочной ситуации.

Б.7. Таблица *SIAGF* (“Логическая структура документов”)

Таблица *SIAGF* определяет логическую структуру документа. В ней указывается, из каких частей состоит документ и как эти части связаны между собой. При этом различаются *статическая структура типа* документа и *динамическая структура экземпляра* документа данного типа. Если первая содержит в себе все возможные части и их связи (в том числе, условные), то вторая – описывает состав частей конкретного экземпляра документа в текущий момент времени.

Структура самого файла *SIAGF.DBF*, представлена в таблице Б.14.

Каждая запись (строка) файла *SIAGF.DBF* описывает одну часть документа. Более подробно назначение полей файла можно охарактеризовать следующим образом:

SIIM – (N) Системный номер типа документа. Должен присутствовать в файле *SIIM*;

SC60 – (Часть) Номер части документа.

Нумерация частей сквозная по всему документу. В процессе работы с документом его части могут заполняться только в порядке возрастания их номеров;

SI75 – (Начало) Системный номер первой строки (записи) части в файле *SIUEK.DBF*;

SC75 – (Конец) Системный номер последней строки (записи) части в файле *SIUEK.DBF*;

SC61 – (Переход). Номер части (экранного образа) документа, к заполнению (просмотру) которой надо перейти после заполнения (просмотра) данной части. (Т.е., в

режиме создания или редактирования документа от нее не отказались.) Если в поле стоит код 98, то это обозначает, что данная часть является последней в экранном образе документа (конец документа);

Таблица Б.14. Структура файла *SIAGF.DBF*

Поле файла	Название поля	Назначение поля	Тип	Длина
<i>SIIM</i>	N	Системный номер типа документа	C	4
<i>SC60</i>	Часть	Номер части документа	N	2
<i>SI75</i>	Начало	Системный номер первой записи части в файле <i>SIUEK.DBF</i>	C	4
<i>SC75</i>	Конец	Системный номер последней записи части в файле <i>SIUEK.DBF</i>	C	4
<i>SC61</i>	Переход	Следующая по порядку часть при заполнении данной части	N	2
<i>SC62</i>	Отказ	Следующая по порядку часть при отказе от данной части	N	2
<i>SC63</i>	Вершина	Головная часть для данной части	N	2
<i>SI59</i>	Статус	Тип (основного) файла части	C	1
<i>SI51</i>	Название части	Название части	C	33
<i>SA01</i>	Фон	Признак цвета фона части	C	1
<i>SA01÷SA37</i>	На экран не выводятся	Служебные и зарезервированные		
<i>SC70</i>	Экран	Резервное. На экране не показывается	N	1
<i>SC50</i>	Спец.1	Тип статуса части	C	1
<i>SC51</i>	Спец.2	Признак вызова процедур при удалении части	C	1

SC62 – (Отказ) Номер следующей части документа в случае отказа от данной части.

Если часть входит в блок (примеры блоков см. ниже), то проставляется номер последней в блоке части, а в последней части блока проставляется номер части, с которой начинается блок.

Если в поле стоит код 98, то это обозначает, что данная часть является последней в маршруте документа (конец документа). Если в поле стоит код 99, то отказ от данной части запрещен.

Внимание! Для первой части документа код 99 обязателен.

Примечание. В случае необходимости, разработчик приложения, может предусмотреть процедурное управление формированием структуры экземпляра документа. Для этого достаточно в режиме создания документа заменить пользовательской функцией номера частей, к которым осуществляется переход, т.е., значения переменных, в которые при обработке документа скопированы поля *SC61* и *SC62* (public переменные – массивы *pSC61* и *pSC62*). Однако такие манипуляции желательно производить при наличии определенных навыков работы с системой *ORD*.

SC63 – (Вершина) Номер головной части для текущей части.

Поле заполняется по следующим правилам:

1. Для первой части документа в поле записывается 0, что обозначает отсутствие у данной части головной, т.е. вершину дерева.

2. Для всех других частей в этом поле указывается номер части, являющейся головной для данной, причем этот номер увеличивается на:

30, если у текущей части имеются подчиненные части, не являющиеся ее частями-продолжениями, т.е. часть сама является вершиной поддерева;

50, если текущая часть является частью-продолжением своей базовой части.

SI59 – (Статус). Тип части - символ из множества: *O, S, X, x, B, H, h, V, v, E, L*. Причем символы:

- *O, S, X, x, B* определяют однослойную часть;
- *H* и *h* указывают на то, что часть является модельным классификатором;
- *V* и *v* указывают на многослойную часть;
- *E* задает текстовую часть;
- *L* указывает на замыкающую часть которая, используется для приказа добавления

слоев в группу частей (блок) в режиме создания. Если в поле *SI50* для части стоит символ *F*, то головная часть для блока - мнимая (содержит символ *Z* в поле *SI50* и не содержит полей для вывода на экран). Для блока частей соблюдается правило: каждому слою мнимой части соответствует один слой подчиненной части.

Правила взаимосвязи статусов частей приведены на рис. 3.

SI51 – (Название части). Название данной части документа (заголовок части).

Здесь указывается смысловое назначение части и, в случае необходимости, ряд специальных символов, определяющих общие правила работы с частью и с документом в целом. Длина поля 33 байта.

Если в этом поле для первой части документа в 26-ой позиции указать символ "#", то можно определить, как показывать документ в режимах создания, просмотра и редактирования, причем 27-28 позиции определяет режим создания, 29-30 позиции – режим просмотр, а 31-32 позиции – режим редактирования

Если в *SI51* части модельного классификатора в 23-ой позиции указан символ "#", то это обозначает, что подчиненные ему модельные классификаторы должны восприниматься как одно целое. При этом следующие два символа определяют количество таких классификаторов.

Пусть, например, граф структуры документа имеет вид, показанный на рисунке Б.6.

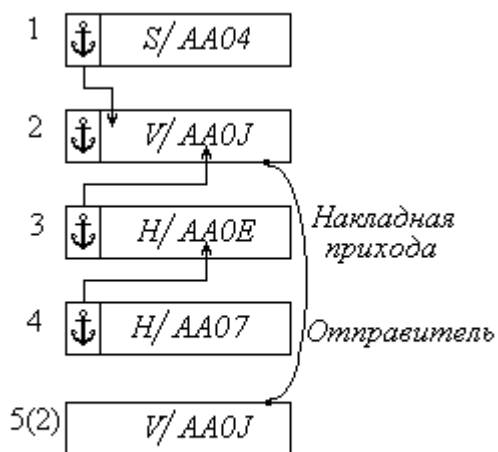


Рис. Б.6. Граф структуры документа

Если из семантических соображений необходимо чтобы части 3 и 4 воспринимались как единое целое, то в файле *SIAGF.DBF* для части 3 документа в поле *SI51* указывается необходимая информация:

Накладная прихода #01 \$,
 где "#"- служебный символ (23-я позиция),
 "01" – символы, преобразование которых в число дает величину (равную 1), указывающую, сколько подчиненных классификаторов должны восприниматься как единое целое, "\$" - служебный символ (33-я позиция).

При этом образы всех подчиненных классификаторов не должны содержать строк с символами управления *R* или *r*, и точки их вызова в поле *SI65* должны начинаться с "z" (см. пример:

документ 000S – “Карточки учета на складе” и пояснения к таблице А10 приложения 1).

SC70 – (Экран) Номер страницы экрана, на которой расположена данная часть. Части документа могут располагаться на нескольких страницах, номера которых указываются в данном поле. Количество страниц в экранном образе не ограничено;

SC50 – (Спец.1) Блок: специальные признаки части.

Если часть не содержит специальных символов управления, то поле не заполняется.

В поле может находиться признак из множества - {*Z, F, D, K, k*} (все буквы латинские):

Z - признак мнимой части. Часть содержит только атрибуты;

F - признак конца блока. При этом номер части в поле *SC63*, указывает на головную часть блока (см. рис. Б.7).

D - признак возможности удаления слоя части (см. рис. Б.7).

Если для части в данном поле проставлен признак D , то эта часть входит в блок и при удалении слоя из части будет выполнено удаление из всего блока.

Если слой удаляется из части-продолжения, для которой признак D не проставлен, то слой удаляется и чистится, т.е. все данные в слое заменяются "пустышками". При этом если в поле $SC51$ стоит признак Y , то перед удалением слоя будут вызваны для исполнения процедуры типа A и P .

Если для части в $SC50$ признак D отсутствует и у этой части есть подчиненные, то новые слои (появившиеся в текущем сеансе работы с экземпляром документа) этих частей удаляются, а взятые из файла помечаются на удаление.

K - признак наличия "кольца". Файл части должен содержать поле с именем " H " + $\text{right}(TDOK,3)$. Учитывает работу файла " H " только с документом типа $TDOK$;

k - признак наличия совместного "кольца" для группы из нескольких типов документов. Файл H -части должен иметь поле " $K999$ ". В этом случае выбранная в одном из экземпляров какого-либо типа документа группы запись части " H " (при наличии "кольца") становится закрепленной только за этим экземпляром, и недоступна другим экземплярам данного типа документа. В качестве признака занятости экземпляра " H " система заносит "1" в указанные поля. В случае удаления экземпляра документа, в котором была задействована запись из " H ", система занесет "0" в указанные поля. (Пример документа, использующего "кольцо", представлен в подсистеме "Y".)

$SC51$ – (Спец.2) Служебный признак, определяющий особенности удаления документа, его части или слоя многослойной части. Признак может принимать значения Y , S , N и D или быть не заполнен.

Y – указывает на необходимость перед удалением слоя из массива многослойной части (по нажатию клавиши <F8>) вызвать процедуры типа A и P , имеющиеся в части.

N – указывает, что часть принадлежит другому документу и не подлежит физическому удалению, при удалении (по нажатию клавиши <Ctrl/F8>) текущего экземпляра документа. (Для части типа " H " наличие данного признака не обязательно, т.к. часть данного типа всегда принадлежат другому документу.)

Документы, содержащие в этом поле значение N для первой части, могут быть удалены системой только по разрешению Администратора.

S – определяет особенности совместного удаления связанных документов.

Если статус основного файла документа " x " (документ всегда создается автоматически), то система не дает возможность удалить такой документ. Удаление документа со статусом основного файла типа " x ", производится в момент удаления документа порождающего документ статуса " x ".

Если тип удаляемой части " H ", то информация остается в файле " H ", но уничтожается признак кольца (если он есть) и все собственные якоря, внесенные в данном экземпляре документа в " H ".

Если удаляется документ D_i , в головной части, которого есть файлы типа " B ", а на основании этих " B " были сформированы документы с головным файлом статуса " x ", то такие документы будут удалены вместе с документом D_i .

Если какая-то часть в этих документах обладает специальным признаком ограничения удаления, то эти ограничения учитываются при удалении. При этом если файл F_i используется в документах с головным файлом статуса " x " в качестве " H ", то запись " H " будет удалена. Информация из головного файла таких документов будет удалена только, если в поле $SC51$ занесено значение S , т. е. присутствует признак согласованного удаления.

D – (в 1-ой части) указывает, что этот документ порождает другие, и при его удалении порожденные им документы будут удалены, если они в поле $SC51$ своих первых частей содержат признак согласованного удаления S .

Внимание! Для программиста всегда доступна глобальная переменная KOD (коды действий в системе), позволяющая определить текущее выполняемое в системе действие и,

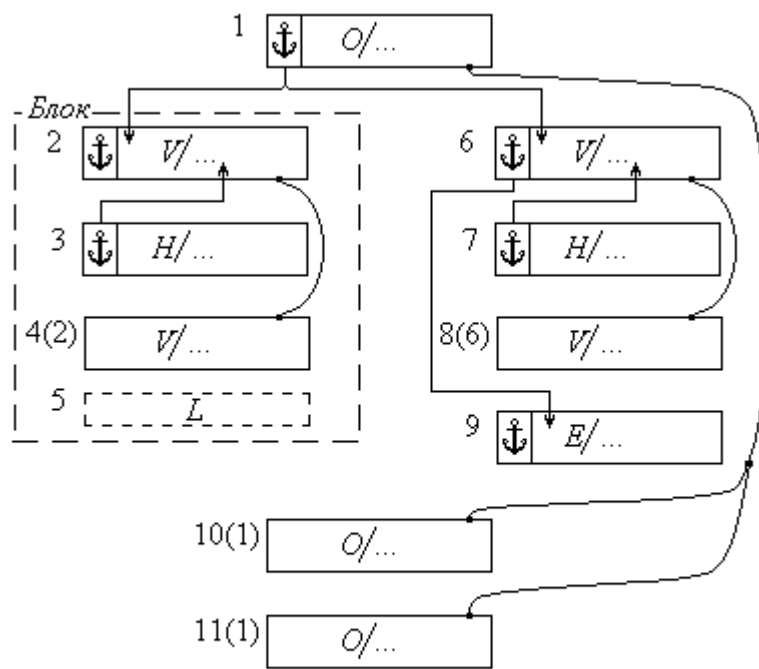


Рис. 5.8. Граф структуры документа 0405

при необходимости динамически изменить правила редактирования конкретного экземпляра документа (см. также Приложение 1, пункт А.7).

SA01 – (Фон). Цветовая гамма для фона части. Если в поле записан символ, отличный от пробела, то для фона части используется специфический набор цветов.

Редактирование файла SIAGF Подсистема Y											
F2-Переключить		F3-Добавить		F4-Удалить		F5-Переименовать		<input checked="" type="checkbox"/> Показать удаленные <input type="button" value="DEL"/> <input type="button" value="BOF"/> <input type="button" value="EOF"/>			
F6-СИУЕК		F8-СИОФ		F9-СИИМ		F10-ВЫХОД					
N	Часть	Начало	Конец	Пережд	Отказ	Вершина	Статус	Экран	Спец.1	Спец.2	Название части
0405	1	98j\$	98^z	2	99	0	O	1			ОСНОВНАЯ
0405	2	98/2	98/D	3	5	31	V	1	Z		МНИМАЯ ЧАСТЬ 1
0405	3	98/J	98/ь	4	5	2	H	1			H-классификатор 1
0405	4	98/n	980[5	5	52	V	1	D		ПРОДОЛЖЕНИЕ ЧАСТИ 1
0405	5	980K	980K	6	2	2	L	1	F		ЗАМЫКАЮЩАЯ ДЛЯ ЧАСТИ 1
0405	6	980R	980j	7	9	31	V	1			Часть 6
0405	7	980p	981N	8	9	6	H	1			H-классификатор 2
0405	8	981T	981I	9	11	56	V	1			Продолжение части 6
0405	9	981r	981x	10	10	6	E	1			Часть 9
0405	10	981~	983v	98	98	51	O	1			ПРОДОЛЖЕНИЕ ЧАСТИ 1
0405	11	97xk	97xH	98	99	51	O	1			ПРОДОЛЖЕНИЕ ЧАСТИ 1
0405	30	97xN	97xf	0	99	0	O	0			O-классификатор
0500	1	97xl	97yP	2	99	0	O	1			ОПЕРАЦИЯ
0500	2	97yv	97yp	3	99	1	V	1			Сведения

Подсистема Y Файл SIAGF.DBF Запись 1 / Всего 70

Рис. Б.7. Описание структуры документа TDOK=0405 в таблице SIAGF

На рисунке Б.7 показан пример содержимого файла *SIAGF.DBF*, описывающего структуру документа типа 0405, графическое представление которой приведено на рисунке Б.8.

На рис.Б.8 показана статическая логическая структура документа (включающая все части, которые могут входить в документ данного типа), состоящая из одиннадцати частей, и указывающая все возможные варианты порядка их следования при формировании динамической структуры экземпляра документа.

Части 2-5, образуют блок 1 (т.е. группу частей, первой из которых является многослойная часть, каким-либо образом семантически объединенных; более точные

определения блока даны в [6]), при этом для части 4 указана специфика ее обработки (символ *D* в поле *SI51* (Спец.1)), указывающая системе на особенности выполнении операции "УДАЛИТЬ" с информацией этой части.

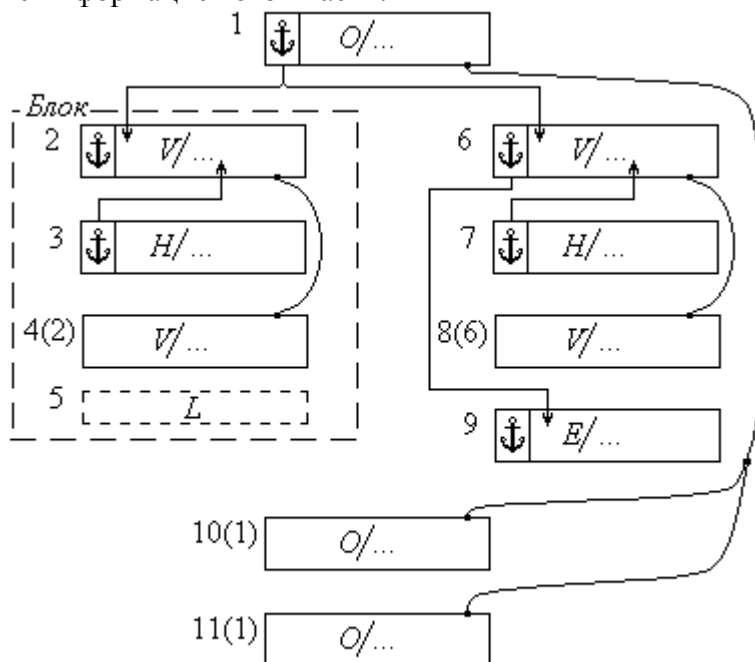


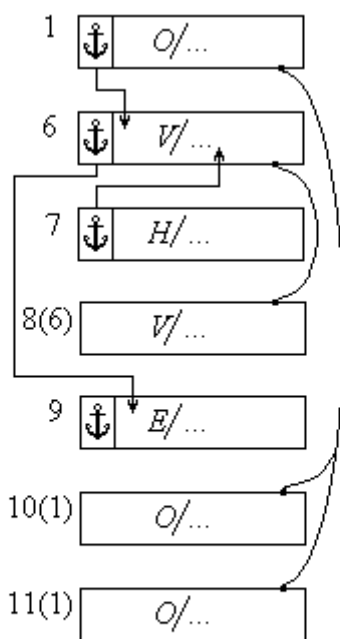
Рис. Б8. Граф структуры документа 0405

Для рассматриваемого примера, часть 4 является частью-продолжением части 2 и входит в блок. Поэтому система, получив указание удалить (*KOD* = 8) слой части, удалит информации всего слоя блока, в который входит часть 4.

При обработке части 6, в случае ее заполнения переход будет осуществлен на часть 7. Если же семантика предметной области такова, что при формировании экземпляра документа данного типа необходимо отказаться от части (данная часть по каким-то соображениям должна быть исключена из документа), то система исключит из экземпляра документа части 6, 7 и 8. Тогда в этом экземпляре непосредственно за частью 4 будет следовать часть 9. (Замыкающая часть 5 является служебной, на экране не показывается и информации самого документа не содержит.)

По мере освоения системы *ORD*, администратор системы, помимо стандартных средств формирования (динамической) структуры экземпляра документа, может овладеть навыками процедурного ее изменения.

Рис. Б9. Граф структуры экземпляра документа 0405 после отказа от частей 2+5



На рисунке Б.9 показан один из вариантов динамической структуры документа 0405, полученный, отказом от частей блока (2 ÷ 5) при создании или редактировании экземпляра документа рассматриваемого типа.

Внимание! Если в документе используются Q-классификаторы, то после описания частей документа, они описываются в *SIAGF.DBF*. При этом описываются только поля *SC60* (30), *SI75*, *SC75*, *SI59* (Q)

Б.8. Таблица *SIMN* (“Меню подсистемы”)

Содержимое таблицы *SIMN* позволяет определять пользовательскую часть меню информационной подсистемы. Этот способ, обычно, используется для демонстрационных версий системы, но может также применяться и в рабочих приложениях.

Таблица Б.15. Структура файла *SIMN.DBF*

Поле файла	Название поля	Содержимое поля	Тип	Длина
<i>SIMN</i>	N	Системный номер пункта меню	C	4
<i>SIML</i>	Метка пункта	Метка пункта меню	C	8
<i>SIMA</i>	Тип пункта	Тип пункта файла	C	1
<i>SIMT</i>	Название пункта	Название пункта меню	C	45
<i>SIMC</i>	Процедура	Вызов процедуры	C	40
<i>SIIM</i>	Документ	Тип документа	C	4
<i>SIMP</i>	Допуск	Возможность редактирования документа	C	6
<i>SIMI</i>	Пользователь	Файлы и их индексы открываемые	C	70
<i>SIM2</i>	Пользователь	перед вызовом документа	C	70
<i>SIMH</i>	Id_Help	Идентификатор help документа	N	7
<i>SIMK</i>	Количество	Максимальное кол-во экземпляров документа	N	7

Меню любой подсистемы включает в себя следующие обязательные пункты:

- Информация;
- ГенМеню;
- Сервис;
- Связь;
- Окна;
- Помощь.

Между пунктами ГенМеню и Сервис в подсистеме обычно располагаются 2-3 или более пунктов, обеспечивающих выбор документов подсистемы и вызов ее процедур.

В зарегистрированных версиях меню может либо непосредственно создаваться программным кодом соответствующих модулей, либо включаться в ресурсы исполнительного модуля, либо заполняться по содержимому файла *SIMN.DBF*. В демо-версии используется только последний вариант.

Описание меню с помощью файла *SIMN.DBF* производится по пунктам: каждая запись файла соответствует одному пункту меню, независимо от уровня его вложения. Пункты меню могут быть трех типов:

- D - пункт меню, обеспечивающий выбор документа
- P - пункт меню, вызывающий процедуру (процедура может использовать в качестве параметров только переменные Public типа или константы)
- M - пункт вложенного меню

Поле системного номера *SIMN*, наличие которого обусловлено требованиями системы *ORD*, не содержит информации, необходимой для описания меню. Это поле можно заполнять, например, четырехразрядными шестнадцатеричными числами, начиная с 0000.

Поле метки пункта меню *SIML* содержит информацию о месте пункта в иерархии меню. Это поле заполняется слева направо, начиная с первой позиции. Количество введенных символов соответствует уровню пункта меню:

- пункты системного меню имеют один символ;
- подменю, обращения к документам или вызовы процедур первого уровня - 2 символа и так далее.

В качестве символов метки могут использоваться любые видимые символы кода, хотя удобнее использовать, например, только цифры и большие латинские буквы. Метка, начинающаяся с пробела (space), соответствует закомментированному пункту меню.

В пределах одного уровня порядок перечисления пунктов в меню соответствует лексикографическому порядку меток.

Метки подчиненных пунктов меню содержат на один символ больше, чем метки пунктов вышестоящего уровня. Причем их подчиненность конкретному вышестоящему пункту определяется *совпадением* первых символов метки подчиненного пункта с меткой вышестоящего пункта. Например, вышестоящий пункт имеет метку "3L2", тогда подчиненные ему пункты могут иметь метки "3L21", "3L22", "3L23", и т.д.

Порядок перечисления пунктов в меню подчиненного уровня определяется последним символом метки. Эти символы не обязательно должны следовать строго друг за другом по таблице кодов: для следующих друг за другом пунктов можно выбрать любые символы, относительное расположение которых в таблице ASCII кодов соответствует порядку пунктов меню.

Длина поля *SIML* в 8 символов обеспечивает 8 уровней меню, что вполне достаточно на практике.

В поле типа пункта меню *SIMA* указывается тип пункта меню: "D", "P" или "M" для выбора документа, вызова процедуры или построения подменю, соответственно. Все буквы должны быть латинскими прописными.

В поле *SIMT* записывается название пункта меню. Название может включать в себя знак амперсанда (&) для выделения символа, использующегося в качестве "горячего ключа" пункта меню. Амперсанд записывается перед выделяемым символом.

В поле *SIMC* записывается тип документа (в старших четырех байтах) для пунктов, типа "D", или вызов процедуры (функции) для пунктов меню типа "P". Вызываемая процедура/функция может либо не иметь параметров, либо использовать параметры Public типа, видимые в точке ее вызова.

Для пунктов типа "M" данное поле не используется.

Все пункты, описанные в файле *SIMN.DBF*, также включаются в ГенМеню. Причем при включении в него из названий пунктов вырезаются амперсанды и все, следующие за амперсандом до ближайшего пробела символы. Это сделано для того, чтобы названия пунктов в ГенМеню не содержали знаков амперсанда, в т.ч. и пункты, в которых "горячие ключи" вынесены в качестве отдельных символов в начало названия.

Примечание. Такое правило может создать определенные неудобства, вырезая вместо отдельного "горячего ключа" первое слово названия или его часть, следующую за амперсандом, если название пункта меню состоит из двух или более слов, а амперсанд включен в первое слово или вплотную примыкает к нему с начала.

В таблице Б.16. приведен пример содержимого файла *SIMN.DBF*, в котором записано пользовательское меню, показанное на рисунке Б.10.

В поле *SIMP* указывается слово NOEDIT, если к документу запрещен доступ на редактирование. В данном поле, также может содержаться информация, относящаяся к вопросам, связанным с безопасностью системы, но здесь эти вопросы не рассматриваются.

В полях *SIM1* и *SIM2* указывается специфическая информация, определяющая доступ к определенным таблицам (файлам), по каким-то причинам необходимая проектировщику при исполнении данного пункта меню.

Поле *SIMH* содержит идентификатор Help. Сама информация для справки готовится стандартными средствами WINDOWS.

Если в поле *SIMK* проставить информацию отличную от 0, то указанное значение определит максимально допустимое количество экземпляров документа данного типа.

Выход за границу указанного числа экземпляров документа будет автоматически контролироваться системой.

Таблица Б.16 Пример описания меню пользователя в файле *SIMN.DBF*

N	Метка пункта	Тип пункта	Название пункта	Процедура	Документ
<i>SIMN</i>	<i>SIML</i>	<i>SIMA</i>	<i>SIMT</i>	<i>SIMC</i>	<i>SIIM</i>
0001	1	M	&Справочники		
0002	11	D	&1 Группа материалов		0007
0003	12	D	&2 Материал		000G
0004	13	D	&3 Группировка товаров		000N
0005	14	D	&4 Поставщики-покупатели		0102
0006	15	D	&5 Банки		0110
0007	16	D	&6 Единицы измерения		000B
0008	17	D	&7 Единицы товара		000O
0009	18	D	&8 Валюты		000R
000A	19	D	&9 Справочник печат. образов		0P00
000B	1A	D	&A Подразделения фирмы		0104
000C	2	M	&Накладные		
000D	21	D	&1 Приход		0008
000E	22	D	&2 Расход		000E
000F	3	M	C&клад		
0010	31	D	&1 Товары на складе		000S
0011	32	D	&2 Штучный товар на складе		000Y
0012	33	P	&3 Просмотр остатков по тканям	OSTAT_TK(
0013	34	P	&4 Просмотр остатков по штучному	OSTAT_HT(
0014	35	P	&5 Проверка сальдо	AKOROST()	
0015	4	M	& Прайс лист		
0016	41	P	&1 Формировать "Прайс лист"	FORMPRS()	
0017	42	P	&2 Просмотр прайс листа по разделам	VIEWPRS()	
0018	43	P	&3 Формировать остатки по магазину	p_ost_tk()	
0019	44	P	&4 Продажи по клиентам	VIEVKLID()	
001A	45	P	&5 Продажи за день	Adaypro()	
001B	46	P	&6 Отчет о прибыли	Apribil()	

Использование файла *SIMN.DBF* позволяет не перекомпилировать программный файл в котором находится меню пользовательской и системной части разработанной информационной подсистемы

Примечание. Каждая информационная система содержит программный файл с именем <SUBSYS>OWIN.PRG, где SUBSYS – имя информационной подсистемы, в котором можно прописать содержимое файла SIMN, в последнем случае надобность в SIMN отпадает. Ниже приводится исходный текст LOWIN.prg.

```
#include "FiveWin.ch"
```

```
memvar oRdoWnd, sinvisibl, ordstek
```

```
FUNCTION LOrdMenu()
```

```
memvar subsys ,oMenu, SubSys1:=array(0,2),SubSys2:=0,SSubSys2
```

```
Local saItems, sI, sLng, sMnuAr
```

```
MENU oMenu
```

```
    MENUITEM "&"+sCodTrf("Информация")
```

```
        MENU
```

```
            MENUITEM "&"+sCodTrf("О подсистеме ") +subsys;
```

```

ACTION MsgInfo(sCodTrf("Новая подсистема")+Chr(13)+"CopyRights С & С
"+sCodTrf("Веди-Талвис"),sCodTrf('О подсистеме')+subsyst);
MESSAGE sCodTrf("Информация о подсистеме")+subsyst

```

```
SEPARATOR
```

```

MENUITEM "&"+sCodTrf("Выход")
ACTION MESSAGE sCodTrf("Завершение работы")
ENDMENU

```

```

MENUITEM "&"+sCodTrf("ГенМеню");
ACTION if(ORDSTЕК=NIL.OR.len(ORDSTЕК)=0,MnuNew(),);
MESSAGE sCodTrf("Сводное меню")

```

```
sMnuAr=sFileMenu() //меню из файла SIMN.dbf
```

```
sLng=len(sMnuAr)
```

```
for sI=1 to sLng
```

```
if len(rtrim(sMnuAr[sI,2]))=1//пункты верхнего уровня должны иметь метку из одного символа
sI=sMakeIt(sMnuAr,sI,chr(1))
```

```
endif
```

```
next
```

```

MENUITEM "&"+sCodTrf("Сервис") WHEN ORDSTЕК=NIL.OR.ASCAN(ORDSTЕК,{|X|
X[8]=2.OR.X[8]=4})=0

```

```
MENU
```

```
MENUITEM "&1 "+sCodTrf("СОЗДАНИЕ СТРАХОВОЙ КОПИИ");//
```

```
SEPARATOR
```

```
MENUITEM "&2 "+sCodTrf("ЧИСТКА УДАЛЕННЫХ ЗАПИСЕЙ");
```

```
ACTION PACKFL (); SEPARATOR
```

```
MENUITEM "&3 "+sCodTrf("АРХИВАЦИЯ ПОДСИСТЕМЫ");
```

```
ACTION MeterMoy( { | oMeter, oText, oDlg, lEnd |;
```

```
,sArch( oMeter, oText, oDlg, @lEnd ) },;
```

```
sCodTrf("Архивация документов подсистемы ....."), sCodTrf("Архивация документов
подсистемы ....."); SEPARATOR
```

```
SEPARATOR
```

```
MENUITEM "&4 "+sCodTrf("РАБОТА С АРХИВОМ")
```

```
SEPARATOR
```

```
MENUITEM "&5 "+sCodTrf("Поиск по запросам")
```

```
MENU
```

```
MENUITEM sCodTrf("&1 Поиск по документу");
```

```
HELPID 6;
```

```
ACTION (xdvM("3","sQStrt","sQStrt(1)"))
```

```
SEPARATOR
```

```
MENUITEM "&"+sCodTrf("3 Поиск по информац.запросу");
```

```
HELPID 12;
```

```
ACTION (xdvM("3","sQStrt","sQStrt(4)"))
```

```
SEPARATOR
```

```
MENUITEM "&"+sCodTrf("4 Информационный запрос")
```

```
sRMnu('0717')
```

```
ENDMENU
```

```
SEPARATOR
```

```
MENUITEM "&6 "+sCodTrf("Конфигурация") ACTION sCfg(2)
```

```
ENDMENU // ОБСЛУЖИВАНИЕ БАЗЫ ДАННЫХ
```

```
MENUITEM sCodTrf("Свя&зь") WHEN ORDSTЕК=NIL.OR.ASCAN(ORDSTЕК,{|X|
X[8]=2.OR.X[8]=4})=0 //09.01.01-исключаем возможность если док-т на редак
```

```
MENU
```

```
MENUITEM "&1 "+sCodTrf("Подготовка пакета отсылки");
```

```
ACTION sQStrt(1,'Q')
```

```
SEPARATOR
```

```
MENUITEM "&2 "+sCodTrf("Снятие отметок об отсылке");
```

```
ACTION sUnMrkHd()
```

```
SEPARATOR
```

```
MENUITEM "&3 "+sCodTrf("Добавление принятого пакета");
```

```
ACTION sEnrPost()
```

```
ENDMENU // Связь
```

```
oMenu:AddMdi() // Add standard MDI menu options
```

```
oMenu:AddHelp( sCodTrf("Система ORD "), "(c) C&Co ")
```

```
ENDMENU
```

```
RETURN oMenu
```

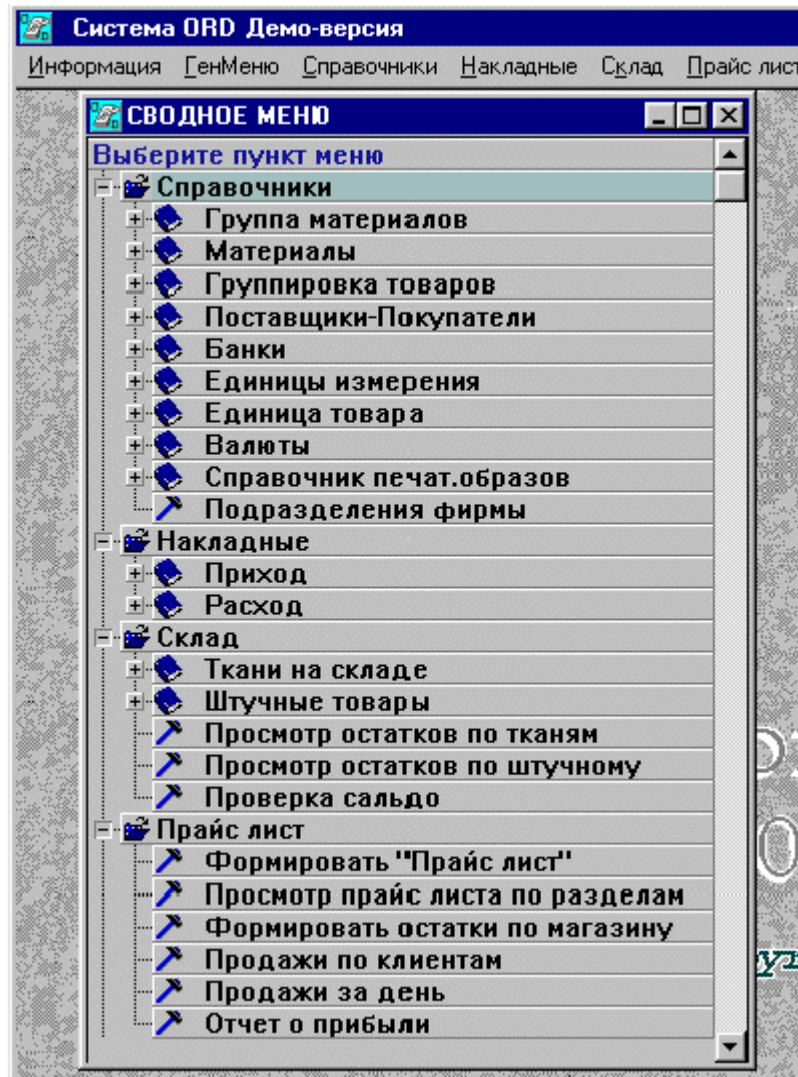


Рис. Б10 Пример меню подсистемы (пользовательская часть)

Все пояснения к исходному тексту приведенной программы можно найти в папке, куда при инсталляции было помещено окружение системы. В этой же папке находятся подробные справочники по языку программирования используемому в системе.

Приложение 3

В1. Общесистемные файлы

На каждой машине ЭВМ создается директорий с именем SUBU с такими же поддиректориями, как и во всех подсистемах системы.

Поддиректорий с именем SUDBF обязательно содержит два файла: *SUUU.DBF* и *SU<номер машины>.DBF*.

В.1. Файл *SUUU.DBF* (“ЭВМ, используемые в системе”)

Файл *SUUU.DBF* используется системой, как контрольный, для устранения ошибок, вызванных "забывчивостью" программиста (при перенастройке подсистем по машинам сети). В этом файле указываются имена электронных дисков ЭВМ сети: каждая запись файла содержит два поля: номер ЭВМ и имя ее электронного диска.

Пусть в системе используется восемь ЭВМ, тогда содержимое файла *SUUU.DBF*, например, для ЭВМ с номером 05 в случае использования в системе восьми ЭВМ, может иметь вид, показанный на рис В.1.

SUUU Номер машины	SI28 Имя электронного диска	SIUU Имя машины
06	F	S06
01	F	VIKA
02	H	OTDEL2
03	F	OTDEL1
04	F	S04
05	F	S05
06	F	S06
07	D	S07
08	D	S08

Рис. В.1. Содержимое файла *SUUU.dbf*, для случая восьми машин в системе

Первая запись описывает характеристики машины, на которой находится файл *SUUU.DBF*. Например, это машина 06 и имя электронного диска этой машины – F (поле *SI28*).

В следующих записях (по числу машин в порядке возрастания их номеров) описываются характеристики других машин.

Если *ORD* используется на одной машине, то содержимое файла *SUUU.DBF* для этого случая показано на рисунке В.2

SUUU-номер машины	SI28-имя электронного диска
05	H
05	H

Рис. В.2. Состав файла *SUUU* для одной машины

В.2. Файл *SU<номер ЭВМ>.DBF* (“Распределение общесистемных файлов в системе”)

Файлы подобного типа используются системой, если хотя бы одна из подсистем использует информационный файл, имя которого начинается с символа "U". Если эксплуатируется сетевой вариант *ORD*, файлы типа "U" позволяют регулировать производительность информационной системы в целом.

Файл *SU<номер ЭВМ>.DBF* содержит описание распределения общесистемных файлов типа "U" по машинам. Для всех машин структура и содержание (список "U"-файлов) одинаковы. Различаются только два последних символа в имени файла. Например, для машины 05 имя файла *SU05.DBF*, а для машины 01 – *SU01.DBF*.

Структура файла *SU<номер ЭВМ>.DBF* приведена на рисунке В.3:

Файл	Назначение	Поле	Тип	Длина	Точность	Сем.код	Спец.инд
SU05	Имя файла	sl20	C	9	0	20	
SU05	Число машин, хранящих файл	SUMA	N	2	0	0	
SU05	Номер хост-ЭВМ	SUXO	C	2	0	0	
SU05	Признак операции	SUMO	C	1	0	0	
SU05	Якорь записи файла	SUYK	C	5	0	0	
SU05	ЭВМ, проведшая процесс	SUL1	C	2	0	0	
SU05	Служебное	SUL2	C	1	0	0	
SU05	Время модификации	SUL3	C	6	0	0	
SU05	Номер док-та, ведущего процесс	SUL4	N	4	0	0	
SU05	Номер машины 01	SU01	C	1	0	0	
SU05	Номер машины 02	SU02	C	1	0	0	
SU05	Номер машины 03	SU03	C	1	0	0	
SU05	Номер машины 04	SU04	C	1	0	0	
SU05	Номер машины 05	SU05	C	1	0	0	
SU05	Номер машины 06	SU03	C	1	0	0	
SU05	Номер машины 07	SU04	C	1	0	0	
SU05	Номер машины 08	SU05	C	1	0	0	

Рис. В.3. Структура файла *SU<номер машины>.DBF* для случая восьми машин в системе

В исходном состоянии файл содержит описание распределения "U"-файлов по машинам системы.

Пусть в системе используется следующие U-файлы – *UOOO.DBF*, *UOVV.DBF*, *UFOR.DBF*, *UMCL.DBF*, *UM64.DBF*, *UM65.DBF*, *UM66.DBF*. Тогда, например, в файле *SU05.DBF* будет записана информация, представленная в таблице В.1.

Информация, записанная для файла *UOOO.DBF*, означает следующее:

- Файл хранится в системе в двух экземплярах (поле *SUMA*);
- Эталон файла хранится на машине 06 (поле *SUXO*). (Не обязательно, чтобы подсистема "O" также хранилась на машине 06. Система все равно всегда будет проверять все операции на файле *UOOO.DBF*, хранящемся на машине 06);
- В полях *SU01 – SU08*, соответствующих номерам машин, указано (знак "+"), что копия хранится на машине 08, а эталон – на машине 06.

В случае аппаратного сбоя администратор базы, на основании первоначального описания, должен восстановить копии файлов по их эталонам (используя копирование через

утилиты ОС) и уничтожить всю служебную информацию, прописанную системой в процессе ее эксплуатации. Ни в коем случае нельзя удалять служебную запись с информацией:

SSSS 0 HE УДАЛЯТЬ / HE 0 / У Д А Л Я Т Ь

Таблица В.1. Исходное состояние содержимого файла *SU05.DBF*

SI20	SUMASUXOS	SUMOSUYK	SUL1	SUL2	SUL3	SUL4	SU01	SU02	SU03	SU04	SU05	SU06	SU07	SU08	
U000	2	06					0					+		+	
U0VV	3	06					0	+				+		+	
UFOR	2	03					0			+		+			
UMCL	7	04					0	+	+	+	+	+		+	
UM64	7	04					0	+	+	+	+	+		+	
UM65	7	04					0	+	+	+	+	+		+	
UM66	7	04					0	+	+	+	+	+		+	
SSSS	0	HE		УДАЛЯТЬ	/	HE	0	/	У	Д	А	Л	Я	Т	Ь

По истечении некоторого интервала времени, в котором были модифицированы эталоны файлов *U000.DBF* и *U0VV.DBF*, в файле *SU05.DBF* может, например, оказаться следующая информация, представленная в таблице В.2:

Таблица В.2. Состояние файла *SU05.DBF* после модификации *U000.DBF* и *U0VV.DBF*

SI20	SUMASUXOS	SUMOSUYK	SUL1	SUL2	SUL3	SUL4	SU01	SU02	SU03	SU04	SU05	SU06	SU07	SU08	
U000	2	17					9					+		Y	
U0VV	3	18					10	+				+		Y	
UFOR	2	03					0			+		+			
UMCL	7	04					0	+	+	+	+	+		+	
UM64	7	04					0	+	+	+	+	+		+	
UM65	7	04					0	+	+	+	+	+		+	
UM66	7	04					0	+	+	+	+	+		+	
SSSS	0	HE		УДАЛЯТЬ	/	HE	0	/	У	Д	А	Л	Я	Т	Ь
	0		4	⊙2	05	054600	11							i	
	0		4	⊙4	05	054606	12							i	
	0		4	⊙4	05	054972	13							i	
	0		4	⊙8	05	054934	14							i	
	0		4	⊙6	05	055904	15							i	
	0		4	⊙<	05	055923	16							i	
	0		4	⊙7	05	055955	17							i	
	0		4	⊙=	05	060027	18							i	

Если в поле *SU<NN>*, где *NN* - номер машины, стоит "+" или "Y", то система гарантирует адекватность копии файла, расположенного на машине *NN*, его эталону.

Для ускорения работы системы следует периодически восстанавливать исходное состояние файлов *SU<NN>.DBF*, удаляя записи, появившиеся после записи *SSSS*. Эту операцию должен выполнять администратор базы данных. Удалив такие записи, администратор обязан восстановить исходное состояние *SU<NN>.DBF* файлов на всех машинах и проверить соответствие копий оригиналам.

За необходимостью привлечения администратора к чистке файлов типа *SU<NN>* система следит сама. Однако он может не дожидаться приглашения системы, а выполнить эту работу сам (приведя систему в состояние, показанное на рис. В.1).

Внимание! Если местоположение эталона "U" файла изменяется, то необходимо **вновь создать эталоны** всех документов.

В.3. Файл <SUBSYS>_X5.DBF (“Установки калькулятора”)

Система позволяет в любой точке использовать встроенный калькулятор и использовать (импортировать), в случае необходимости, результат работы калькулятора как исходное значение какого-либо информационного поля документа.

Содержимое файла задает начальные установки режимов работы калькулятора.

При входе в калькулятор из документа содержимое записи файла, соответствующей данному типу документа, определяет устанавливаемые режимы. В процессе работы с калькулятором эти режимы можно изменять, а при выходе из калькулятора – сохранить в файле.

Структура файла <SUBSYS>_X5.DBF (для подсистемы “F”) представлена в таблице В.3:

Таблица В.3. Структура файла <SUBSYS>_X5.DBF

Поле файла	Название поля	Содержимое поля	Тип	Длина
<i>SIIM</i>	Документ	Системный номер типа документа	C	4
<i>F_AF</i>	Режим	Режим отображения на экране	N	2
<i>F_HH</i>	Высота окна	Высота окна редактирования	N	2
<i>F_BB</i>	Ширина окна	Ширина окна редактирования	N	2
<i>F_HO</i>	Верх окна	Высота начала окна редактирования	N	2
<i>F_BO</i>	Начало окна	Левый край начала окна редактирования	N	2
<i>F_HT</i>	Высота таблицы	Высота начала таблицы переменных	N	2
<i>F_BT</i>	Начало таблицы	Левый край начала таблицы переменных	N	2

Назначение полей файла:

SIIM – системный номер типа документа

F_AF – режим отображения на экране:

1 - на экране отображены все части калькулятора: (фон, окно редактирования и таблица переменных);

2 - на экране отображено окно редактирования и таблица переменных (фон калькулятора выключен);

3 - на экране отображено только окно редактирования: (фон и таблица переменных выключены);

F_HH – Высота окна редактирования, уменьшенная на 2. Минимальное значение – 1 (т.е., 3 строки), максимальное – ограничено размером экрана с учетом высоты начала окна редактирования;

F_BB – Ширина окна редактирования. Минимальное значение - 19 колонок, максимальное – 77;

F_HO – Высота начала окна редактирования. Минимальное значение в режиме отображения 1 – 2, в других режимах – 0, максимальное – ограничено размером экрана с учетом высоты окна редактирования;

F_BO – Левый край начала окна редактирования. Минимальное значение – 0, максимальное – ограничено размером экрана с учетом ширины окна редактирования;

F_HT – Высота начала таблицы переменных. Минимальное значение - в режиме отображения 1 – 2, в других режимах - 0, максимальное - ограничено размером экрана с учетом высоты таблицы (числа переменных);

F_BT – Левый край таблицы переменных. Минимальное значение – 0, максимальное – 38 при числе переменных до 7, 37 – при числе переменных больше 7.

Подробное описание режимов работы калькулятора, доступно по клавише F1, при вызове калькулятора.

Файл <SUBSYS>_X5.DBF оставлен в системе для совместимости с DOS версией системы.

В.4. Файл *SSFP.DBF* (“Косвенные ссылки”)

Файл предназначен для автоматического переименования (с помощью утилиты DB.EXE) собственных, разделяемых и системных информационных файлов и полей этих файлов. (Режим переименования позволяет автоматизировать процесс перезаписи существующих прикладных программ, в случае изменения предметной области.)

Для этого в файле *SSFP.DBF* содержится информация о тех файлах системы (кроме управляющих системных файлов), в которых упоминаются имена каких-либо файлов системы и/или их полей, т.е. имеются ссылки на другие файлы и их поля. Структура файла *SSFP.DBF* приведена в таблице 6.4.

Таблица В.4. Структура файла *SSFP.DBF*

Поле файла	Название поля	Содержимое поля	Тип	Длина
SI20	Файл	Имя файла, содержащего ссылку	C	48
SI26	Поле	Имя ссылочного поля файла	C	4
SSSK	Атрибут	Тип ссылки	N	2

Значение типа ссылки определяется как сумма следующих четырех составляющих:

- 1 – ссылочное поле файла может содержать имя файла;
- 2 – ссылочное поле файла может содержать имя поля;
- 4 – имя файла или поля может быть в контексте;
- 8 – необходимо выравнивать длину имени (сохранять позиции всех подполей)

Пример содержимого файла *SSFP.DBF* представлен на рисунке В.4

Файл	Поле	Тип
V_11	JK11	7
V_11	JK12	5
BP86	BP88	1
BP86	BP89	1
SGCH	SI20	1
SGCH	SI26	2
SGCH	SH20	1
SGCH	SH26	2

Рис. В.4. Пример содержимого файла *SSFP.dbf*

В свою очередь, содержимое одного из файлов, указанных в файле *SSFP.DBF*, например, *V_11.DBF*, показано на рисунке В.5. Как видно из этого рисунка, поля JK11 и JK12 файла *V_11.DBF* содержат ссылки на другие поля.

Примечание. Использование файла *SSFP* не рекомендуется. Значительно проще описать используемые пользовательские функции в теле таблицы SIUEK, как независимые от используемых в ней данных. Однако, в виду того, что некоторые (давно созданные) фреймы используют файл *SSFP*, он сохранен в системе.

Документ	JK11 – Функция	JK12 – Функция таблицы	Тип
0032	jkvit1("G_V1")	jkvit2("G_V2")	2
0012	jttn1("G_TN")		3
0203	jplat1("F_LA")		3
0703	jtreb1("F_RE","gtre1",jtip)		4
0240	jpord1("F_OR")		3
0239	jrord1("F_OP")		3
0013	jttn1("G_TN")		3
0485	bBlInPrn("b_tc")		5
0410	jp("BGA5","BVA5")		5
0421	jp("BGC1","BVC1")		5
0401	jbpsch0		5
0411	jp("BGA6","BVA6")		5
0427	jp("BGC7","BVC7")		5
0408	jp("BGA7","BVA7")		5
0422	jp("BGA9","BVA9")		5
0420	jp("BGB0","BVB0")		5
0418	jp("BGZ2","BVZ2")		5

Рис. В.5. Содержимое файла В_11.DBF (процедуры печати подсистемы В)

В.5. Файл *SSLO.DBF* (“Понятия системы”)

Файл *SSLO.DBF* используется системой и ее администратором для конструирования новых понятий в системе. Этот файл применяется для построения непротиворечивого отображения понятий инфологического уровня в понятия самой системы.

Содержимое файла формируется автоматически при исполнении режима “F5-Создать” из его окна прокрутки (для этого конечно надо перейти в папку SUBS, где помимо общесистемных файлов хранится много полезной информации).

Пример содержимого файла *SSLO.DBF* представлен на рисунке В.6.

В окне прокрутки данного файла можно выполнить следующие действия:

- <F1> - получение справки по файлу *SSLO.DBF*;
- <F2> - переключение режимов видимости записей, помеченных на удаление;
- <F3> - добавление в конец файла пустой записи. Данный режим не рекомендуется использовать, т.к. содержимое файла формируется автоматически при нажатии клавиши F5;
- <F4> - удаление выбранной записи, при этом происходит физическое удаление (с упаковкой) записи и понятие становится недоступным для текущей машины;
- <F5> - создание (или расширение) списка понятий задействованного в системе (см. сообщения при исполнении данного режима).

Следует отметить, что список понятий, доступных подсистеме, формируется на основе имеющихся понятий всех подсистем, перечисленных в файле *SISOA.DBF* – распределение информации по подсистемам и по машинам сети.

Использовать данный режим следует в том случае, если в системе разработан новый документ, и он прошел режим “Согласования статусов” без противоречий. И только в этом случае его можно помещать в файл *SSLO*;

- <F10> - выход в основной режим (редактирования *SIORD.DBF* подсистемы "S").
- <F12> - вызов файла помощи *ORD*.

Сист.	Имя понятия	Основной файл	Якорь	NALK
0010	МАРКИ	SG53	SG53	
0012	ТОВАРНО-ТРАНСПОРТНАЯ НАКЛАДНАЯ	SGTX	SGTX	
0013	ТТН / разливная продукция/	SGTX	SGTX	
0014	СПРАВОЧНИК "ВОДИТЕЛИ АВТОТРАНСПОРТ	SG28	SG28	
0016	СПРАВОЧНИК "УСЛУГИ"	SGRM	SGRM	
0017	СПРАВОЧНИК "ПРОДУКЦИЯ НЛ"	SGRV	SGRV	
0019	СХЕМЫ ПРОВОДОК ПО ГОТОВОЙ ПРОДУКЦИИ	SGCH	SGCH	
0020	СПРАВОЧНИК "ТИПЫ ПРОДУКЦИИ"	SGTY	SGTY	
0021	ПЕРЕТАРИВАНИЕ	GPER	GPER	
0022	Для документа 0021	GPEV	GPEV	
0023	Для документа 0035	GSSB	GSSB	
0024	ОТЧЕТ СКЛАДА О ДВИЖЕНИИ ГОТОВОЙ ПР	GMMA	GMMA	

Рис. В.6. Пример содержимого файла SSLO.DBF

В.6. Файл *SSII.DBF* (“Сообщения системы *ORD*”)

Общесистемный файл *SSII.DBF* содержит описание контролируемых системой ошибочных ситуаций, а также обеспечивает возможность *динамического вызова внешнего программного кода в исходном виде*.

Файл доступен для расширения только администратору системы.

В.7. Файл *SS_U.DBF* (“Связь локальных сетей”)

Общесистемный файл *SS_U.DBF* содержит описание правил приема-передачи информации между локальными сетями, использующими общие данные (см. Help – F1, когда маркер курсора находится в таблице *SIORD* на имени файла *SS_U*).

Файл доступен для расширения только администратору системы.

В.8. Файл *SSIO.DBF* (“Контроль процессов системы *ORD*”)

Общесистемный файл *SSIO.DBF* содержит описание контролируемых системой ошибочных ситуаций, возникающих в процессе ее функционирования, и реакцию на них. В отличие от описанного выше файла *SSII.DBF*, сообщения которого, в основном, относятся к утилите *DB.EXE*, сообщения данного файла относятся к ситуациям, возникающим при исполнении приложения *ORD.EXE*.

Файл доступен для расширения только администратору системы.

Информация файла позволяет не только отслеживать действия любого пользователя, но и контролировать уровень доступа к информации.

Включение в работу “Журнала учета работы пользователей” определяются значением переменной *Sinvisible* в головной программе системы *ORD* (см. исходный текст *T.prg* в папке, откуда запускается исполняемый модуль).

Права доступа к любому типу документа (или экземпляру или полю) определяются парольной защитой для каждого пользователя системы.

Литература

1. РД40. РСФСР 050087. Проекты (работы) дипломные и курсовые. Правила оформления.
2. Копейкин М.В., Спиридонов В.В., Шумова Е.О. Методические указания к курсовому проектированию СПб.: СЗПИ 1998.
3. Копейкин М.В., Спиридонов В.В., Шумова Е.О. Базы данных. Объектно-реляционный подход: Учеб. пособие. - СПб.: СЗПИ, 1998. - 96 с.
4. Копейкин М. В., Спиридонов В. В., Шумова Е. О. Базы данных. Основы SQL реляционных баз данных: Учеб. пособие. – СПб.: СЗТУ, 2004. – 176 с.
5. Копейкин М. В., Спиридонов В. В., Шумова Е. О. Базы данных. Концепция баз данных. Учеб. пособие. – СПб.: СЗТУ, 2004. – 111 с.
6. Копейкин М.В., Спиридонов В.В., Шумова Е.О. Базы данных. Инфологические модели баз данных. Учеб. пособие. – СПб.: СЗТУ, 2004. – 187 с.
7. Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика. 2-е издание. – М.: Издательский дом “Вильямс”. 2000. – 1120 с.
8. Мальцев М.Г., Хомоненко А.Д., Цыганков В.М. Базы данных: Учеб. пособие. – СПб.: Изд-во Корона принт, 2002. – 672 с.
9. Копейкин, М.В. Базы данных. HTML и PHP . Методические указания к выполнению лабораторных работ для студентов бакалавриата направлений 09.03.01, 09.03.02 / М.В. Копейкин, В.В. Спиридонов, Е.О. Шумова. – СПб.: Изд-во Национальный минерально-сырьевой университет «Горный», 2016. – 59 с.

Средства обеспечения освоения дисциплины (ресурсы Internet)

10. Сайт WWW.ORD.COM.RU

Содержание

1. ОБЩИЕ ПОЛОЖЕНИЯ	4
2. ТЕМАТИКА КУРСОВОГО ПРОЕКТА	5
3. ЗАДАНИЕ НА ТИПОВОЙ КУРСОВОЙ ПРОЕКТ	6
4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ	7
4.1. Техническое задание (ТЗ)	8
4.2. Разработка управляющей части программного комплекса	12
4.3. Основные определения, используемые в системе	15
4.4. Разработка структуры документов предметной области	25
4.5. Проектирование программ в среде ORD	54
4.5.1. Компиляция и передача пользовательских программ в систему	68
4.5.2. Общие требования к именам программ, файлов и полей файлов в системе ORD	70
4.5.2.1. Структура имен программ	70
4.5.2.2. Структура имен файлов .DBF	70
4.5.2.3. Список системных управляющих файлов	72
4.5.2.4. Имена полей .DBF файлов	73
4.5.3. Распределение файлов по директориям	74
5. ТЕСТИРОВАНИЕ	76
Приложение 1	78
<i>А. Системное управление атрибутами документа</i>	78
А.1 Оформление документа	78
А.2 Представление простых полей документа	79
А.3 Представление полей классификаторов К- и Q-типа	80
А.4 Представление модельных (Н) классификаторов	82
А.5 Связь частей документа	85
А.6 Копирование полей документа	88
А.7 Пользовательские и системные функции	100
А.8 Дополнительные служебные признаки	112
Приложение 2	114
<i>Б. Структура управляющих системных таблиц</i>	114
Б.1. Таблица <i>SISOA</i> (“Адреса подсистем”)	114
Б.2. Таблица <i>SIORD</i> (“Справочник файлов подсистемы”)	116
Б.2.1. Правила описания логической структуры файла в <i>SIORD</i>	117
Б.3. Таблица <i>SIDO</i> (“Атрибуты документов”)	118
Б.4. Таблица <i>SIIM</i> (“Список типов документов подсистемы”)	119
Б.5. Таблица <i>SIUEK</i> (“Экранные образы документов”)	123
Б.6. Таблица <i>SIAOF</i> (“Списки файлов документов”)	129
Б.7. Таблица <i>SIAGF</i> (“Логическая структура документов”)	131
Б.8. Таблица <i>SIMN</i> (“Меню подсистемы”)	137
Приложение 3	143
<i>В1. Общесистемные файлы</i>	143
В.1. Файл <i>SUUU.DBF</i> (“ЭВМ, используемые в системе”)	143
В.2. Файл <i>SU<номер ЭВМ>.DBF</i> (“Распределение общесистемных файлов в системе”)	144
В.3. Файл <i><SUBSYS>_X5.DBF</i> (“Установки калькулятора”)	146
В.4. Файл <i>SSFP.DBF</i> (“Косвенные ссылки”)	147
В.5. Файл <i>SSLO.DBF</i> (“Понятия системы”)	148
В.6. Файл <i>SSII.DBF</i> (“Сообщения системы <i>ORD</i> ”)	149
В.7. Файл <i>SS_U.DBF</i> (“Связь локальных сетей”)	149
В.8. Файл <i>SSIO.DBF</i> (“Контроль процессов системы <i>ORD</i> ”)	149
Литература	151
Содержание	152

Учебное издание

Базы данных

Методические указания по курсовому проектированию
для студентов бакалавриата направления 09.03.01

Составители:

КОПЕЙКИН Михаил Васильевич
СПИРИДОНОВ Виктор Валентинович
ШУМОВА Елена Олеговна

Ответственный за выпуск *М.В. Копейкин*