

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Уфимский государственный авиационный технический университет»**

Кафедра автоматизированных систем управления

ПРОЕКТИРОВАНИЕ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

**Лабораторный практикум
по дисциплинам**

**«Информационные системы управления документооборотом»,
«Системы электронного документооборота»**



Уфа 2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Уфимский государственный авиационный технический университет»

Кафедра автоматизированных систем управления

ПРОЕКТИРОВАНИЕ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

Лабораторный практикум
по дисциплинам

«Информационные системы управления документооборотом»,
«Системы электронного документооборота»

Учебное электронное издание сетевого доступа

© УГАТУ

Уфа 2022

Авторы-составители: Н. О. Никулина, А. Н. Набатов, А. И. Малахова

Проектирование систем электронного документооборота: лабораторный практикум по дисциплинам «Информационные системы управления документооборотом», «Системы электронного документооборота» [Электронный ресурс] / Уфимск. гос. авиац. техн. ун-т ; [авт.-сост. : Н. О. Никулина, А. Н. Набатов, А. И. Малахова]. – Уфа : УГАТУ, 2022. – URL: https://www.ugatu.su/media/uploads/MainSite/Ob%20universitete/Izdateli/El_izd/2022-162.pdf

Цель лабораторного практикума – ознакомление с пользовательским интерфейсом и возможностями программного средства Domino Designer, предназначенного для проектирования систем электронного документооборота на программной платформе IBM Notes.

Предназначен для студентов старших курсов, занимающихся изучением систем электронного документооборота.

Рецензент д-р. техн. наук, проф. Л. Р. Черняховская

При подготовке электронного издания использовались следующие программные средства:

- Adobe Acrobat – текстовый редактор;
- Microsoft Word – текстовый редактор.

Авторы-составители: *Никулина Наталья Олеговна,
Набатов Александр Нурович,
Малахова Анна Ивановна*

Верстка *Р. М. Мухамадиева*

Программирование и компьютерный дизайн *О. М. Толкачёва*

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

Подписано к использованию: 04.08.2022
Объем: 4,56 Мб.

ФГБОУ ВО «Уфимский государственный авиационный технический университет»
450008, Уфа, ул. К. Маркса, 12.
Тел.: +7-908-35-05-007
e-mail: rik@ugatu.su

СПИСОК СОКРАЩЕНИЙ

API – Application Programming Interface (интерфейс прикладного программирования)

BMP – Bitmap Picture (формат хранения растровых изображений)

CSP – Content Services Platforms (система обслуживания информации)

GIF – Graphics Interchange Format (растровый формат графических изображений)

HTML – HyperText Markup Language (язык гипертекстовой разметки)

HTTP – HyperText Transfer Protocol (протокол передачи гипертекста)

IBM – International Business Machines

ISO – International Standardization Organization (международная организация по стандартизации)

JPEG – Joint Photographic Experts Group (растровый формат графических изображений)

OLAP – Online Analytical Processing (аналитическая обработка в реальном времени)

OLE – Object Linking and Embedding (технология связывания и внедрения объектов в другие документы и объекты)

АРМ – автоматизированное рабочее место

БД – база данных

СЭД – система электронного документооборота

ВВЕДЕНИЕ

Документы – это основные информационные ресурсы любой организации, работа с ними требует правильной постановки задачи. Документы обеспечивают информационную поддержку принятия управленческих решений на всех уровнях и сопровождают все бизнес-процессы. Документооборот – это непрерывный процесс движения документов, объективно отражающий деятельность организации и позволяющий оперативно ей управлять. Длительный поиск нужного документа, потери, дубликаты, задержки с отправкой и получением, ошибки персонала составляют далеко не полный перечень проблем, возникающих при неэффективном построении документооборота. Всё это может сильно затормозить работу организации.

Эффективный документооборот является обязательной составляющей эффективного управления, он исключительно важен для правильной организации финансового и управленческого учёта.

Упорядочивание документооборота начинается с составления реестра используемых в организации документов. Для каждого типа документа определяется маршрут движения, от момента возникновения документа до уничтожения или отправки в архив. Каждый этап жизненного цикла документа подробно описывается: определяется инициатор, ответственный за выполнение каждого этапа, куратор, список подписывающих, список рассылки. Такой подход позволяет упорядочить документооборот, исключить дублирующие документы и функции, и как следствие – оптимизировать работу в целом.

Для организации электронного документооборота в настоящее время существует большое количество разнообразных по функциональным возможностям и ценовым характеристикам программных средств. Но зачастую установка системы «под ключ» не приносит желаемого эффекта, поскольку на предприятии существуют некоторые особенности, которые требуют настройки готовой системы под нужды управления предприятием. В этом случае следует обратить особое внимание на класс систем, являющихся своеобразной «платформой» для организации документооборота на предприятии. Одной из таких систем является система организации коллективной работы с документами IBM Notes.

ЛАБОРАТОРНАЯ РАБОТА № 1

ПРОЕКТИРОВАНИЕ ФОРМ ДОКУМЕНТОВ

1.1. Цель работы

Целью работы является ознакомление с пользовательским интерфейсом и возможностями программного средства Domino Designer, предназначенного для проектирования форм документов.

1.2. Теоретические сведения

Самой важной компонентой IBM Notes являются базы данных. Основная единица хранения базы данных Notes – **документ**, который является фактически эквивалентом бумажного документа. Разработчик базы данных заранее определяет, какой тип документов должен храниться в базе данных, и проектирует одну или несколько форм для создания этих документов.

Основными элементами баз данных Notes являются формы. С их помощью создаются документы с информацией, которая хранится в базе данных. **Форма** – это некоторый бланк, который должен заполнить пользователь, чтобы поместить документ в базу данных. Каждая база данных Notes имеет свой набор форм, подготовленных разработчиком базы. Каждая форма имеет свое имя, отображающее, как правило, какой именно документ готовится по этой форме. После открытия базы данных пользователю предоставляется возможность выбрать нужную форму из имеющихся в базе. Например, база данных «Отдел кадров» может иметь формы: «Личная карточка», «Образование», «Паспорт» и т.д.

Основными элементами форм являются **поля** – при создании документов пользователь видит их на экране как некоторые области, помеченные уголками. В поля вводятся данные, которые должны храниться в базе. Пользователь может получить подсказку в нижней строке экрана, какие именно данные и в каком виде вводить в то или иное поле. Информация такого рода является динамической для той или иной формы.

Также в формах присутствует некоторая текстовая информация, которую в режиме заполнения формы изменить нельзя. Обычно к такой информации относятся названия полей и разные надписи в форме. Информацию этого рода называют статической, изменить ее можно лишь в режиме редактирования формы, имея определенные права для работы с базой.

Форма может содержать и другие элементы, улучшающие наглядность формы и удобство работы с ней – это таблицы, а также графические кнопки и объекты, нажав на которые, можно получить некоторые варианты заполнения полей, или выполнить некоторые действия над формируемым документом, или получить справочную информацию. Эти элементы позволяют выполнить определенные действия над документами и полями. В них могут быть заложены действия в виде простой операции Notes, @формулы или программы на языке LotusScript. Гиперобъекты в виде всплывающих текстов предоставляют дополнительную информацию по использованию формы или документа.

Взаимодействие документов реализуется путем включения в формы вложений и связей. Вложения позволяют присоединить к форме любой файл. Пользователи смогут в каждом документе, созданном по данной форме, извлечь файл или запустить его на выполнение. Связи позволяют обеспечить взаимодействие между документами, представлениями и другими базами данных Notes. Эти связи могут активизироваться при чтении документов и предоставлять дополнительную информацию.

1.3. Порядок выполнения работы

1.3.1. Создание БД без использования шаблона

Запустите IBM Domino Designer. Выберите пункт «Создать новую БД» (*File/Application/New*) (рис. 1).

В появившемся диалоговом окне (рис. 2) присвойте название базе данных, задайте имя файла (по умолчанию совпадает с названием базы данных) и местоположение (при отсутствии сервера выберите «Локальный компьютер»). В поле «Шаблон» выберите «-Нет-». Название базы данных должно отражать предметную область.

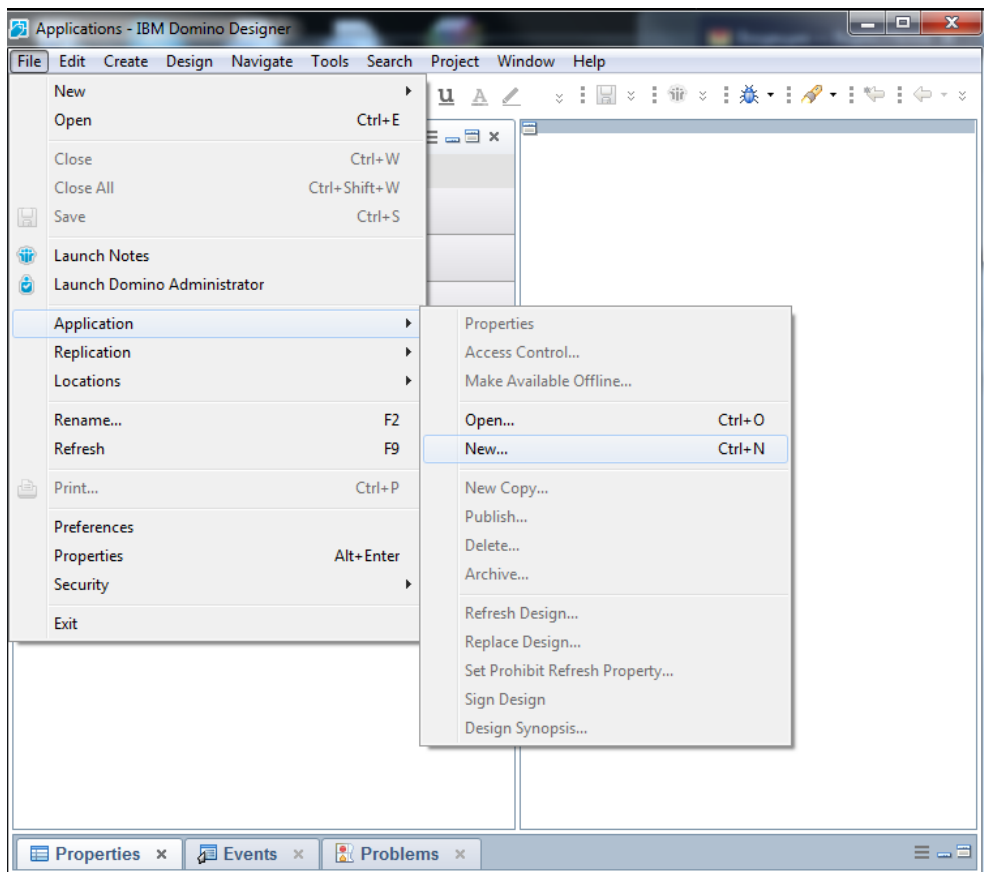


Рис. 1. Создание новой БД

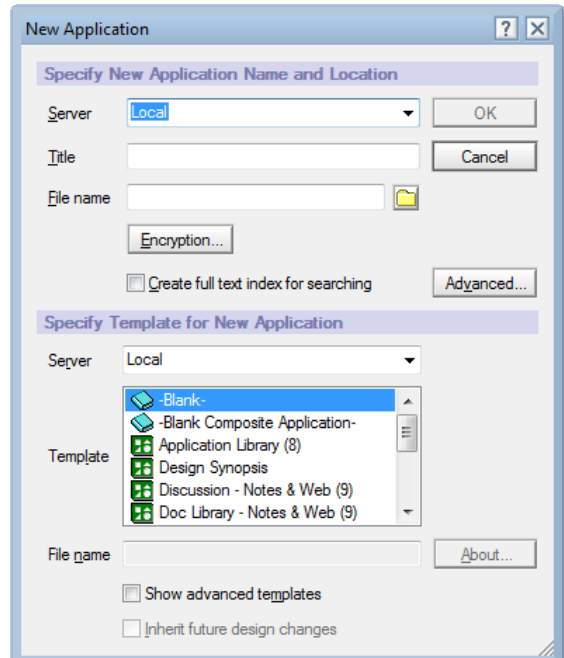
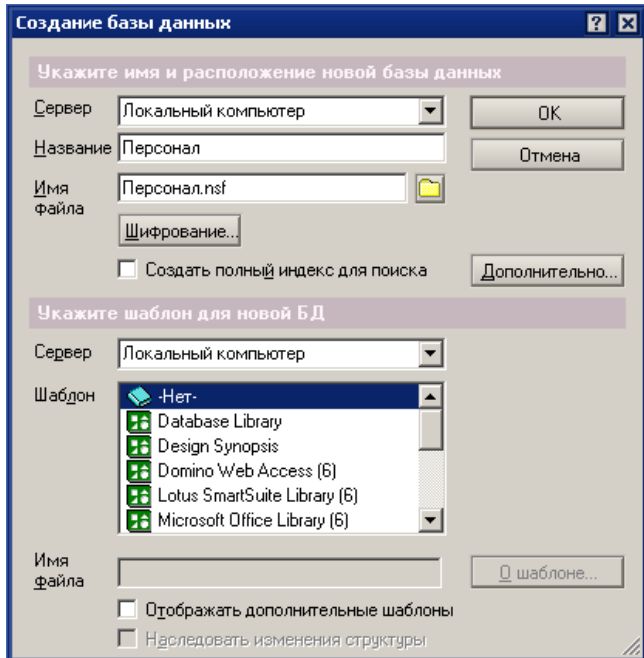


Рис. 2. Окно создания базы данных

Оболочка базы данных создана. Чтобы впоследствии ее открыть в режиме разработки, выберите пункт «Открыть существующую БД» после запуска IBM Domino Designer.

1.3.2. Проектирование формы

В базе данных должно быть не менее 2 различных форм документов. Вид электронной формы должен быть максимально приближен к виду бумажного шаблона документа.

При создании новой базы данных Notes по умолчанию предлагает создать форму – открывается соответствующее окно с заголовком «Создать форму».

Для создания форм также можно воспользоваться командой меню *Создание/Конструктор/Форма (Create/Design/Form)* (рис. 3).

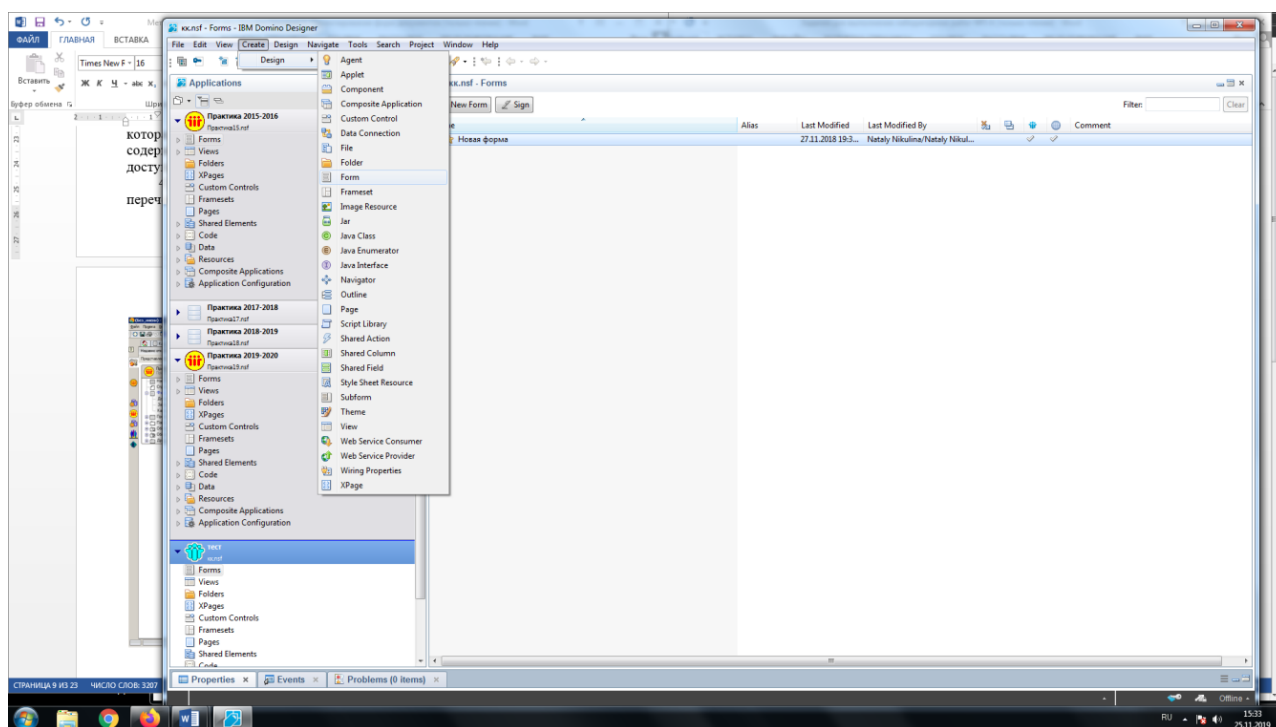


Рис. 3. Создание формы документа

На экране появятся четыре области (рис. 4):

1) область формы (верхняя часть экрана) – чистый лист, который в процессе проектирования преобразуется в шаблон документа;

2) область конструктора (правая нижняя часть экрана), в которой отображаются свойства элементов формы;

3) область объектов и списков (левая нижняя часть экрана), которая является вспомогательной для области конструктора и содержит списки полей базы данных, а также функций и команд, доступных при проектировании элементов форм;

4) закладка базы данных (левая часть экрана), содержащая перечень элементов базы данных.

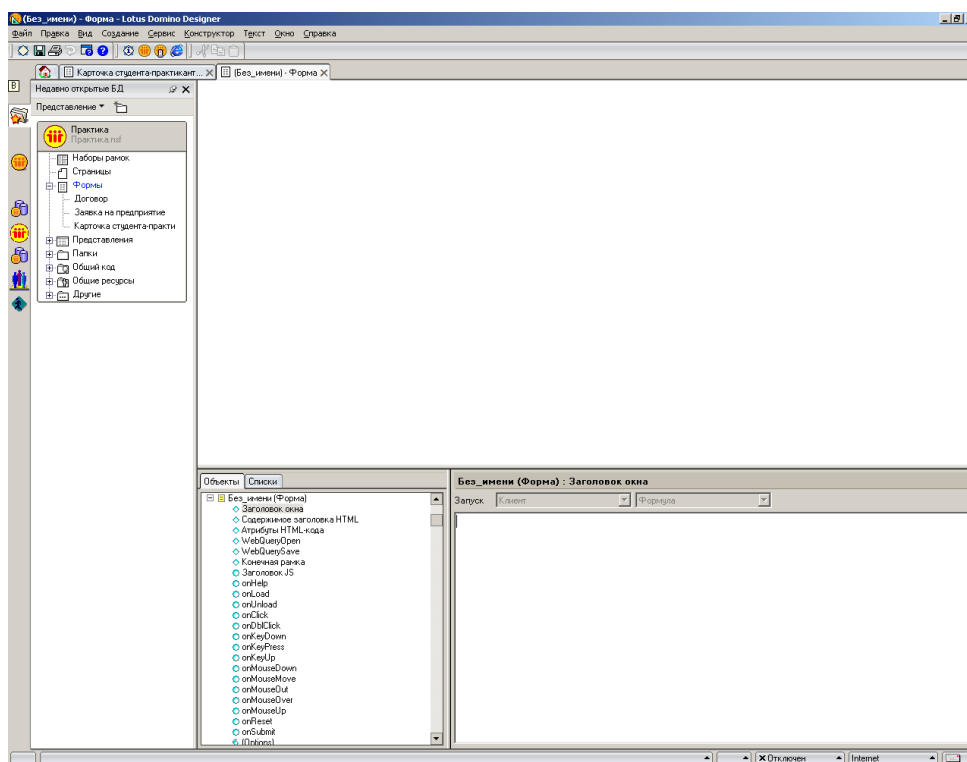


Рис. 4. Окно проектирования формы

Процесс проектирования формы включает определение свойств формы и создание необходимых элементов.

Отладка формы с целью проверки вносимых изменений производится при помощи команды меню *Конструктор/Просмотреть в Notes (Design/Preview in Notes)* (рис. 5).

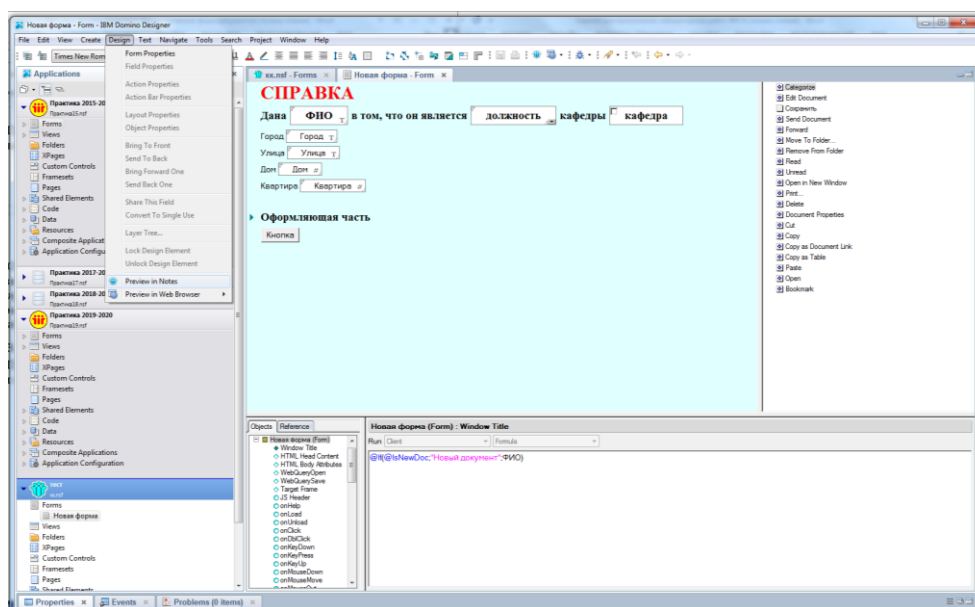


Рис. 5. Команда проверки разработанной формы

Это дает разработчику возможность представить, что будет видеть пользователь на экране при работе с данной формой.

Определение свойств формы

Проектирование формы начинается с определения ее свойств путем выполнения команды *Конструктор/Свойства формы (Design/Form Properties)*.

На вкладке «**Сведения о форме**» можно задать следующие атрибуты (рис. 6).

Имя формы – название формы документа (например, «Акт о выполнении работ», «Контракт» и т. д.).

Тип формы – предназначен для выбора типа документов. Возможны следующие типы документов: «Документ», «Ответ на документ», «Ответ на ответ». Большинство документов имеют тип «Документ». Это означает, что документы, созданные с помощью данного типа формы, были созданы в качестве исходного, независимого документа. Документ типа «Ответ» является подчиненным документу типа «Документ», документ типа «Ответ на ответ» является подчиненным документу типа «Ответ». Notes поддерживает иерархию ответных документов до 32 уровней.

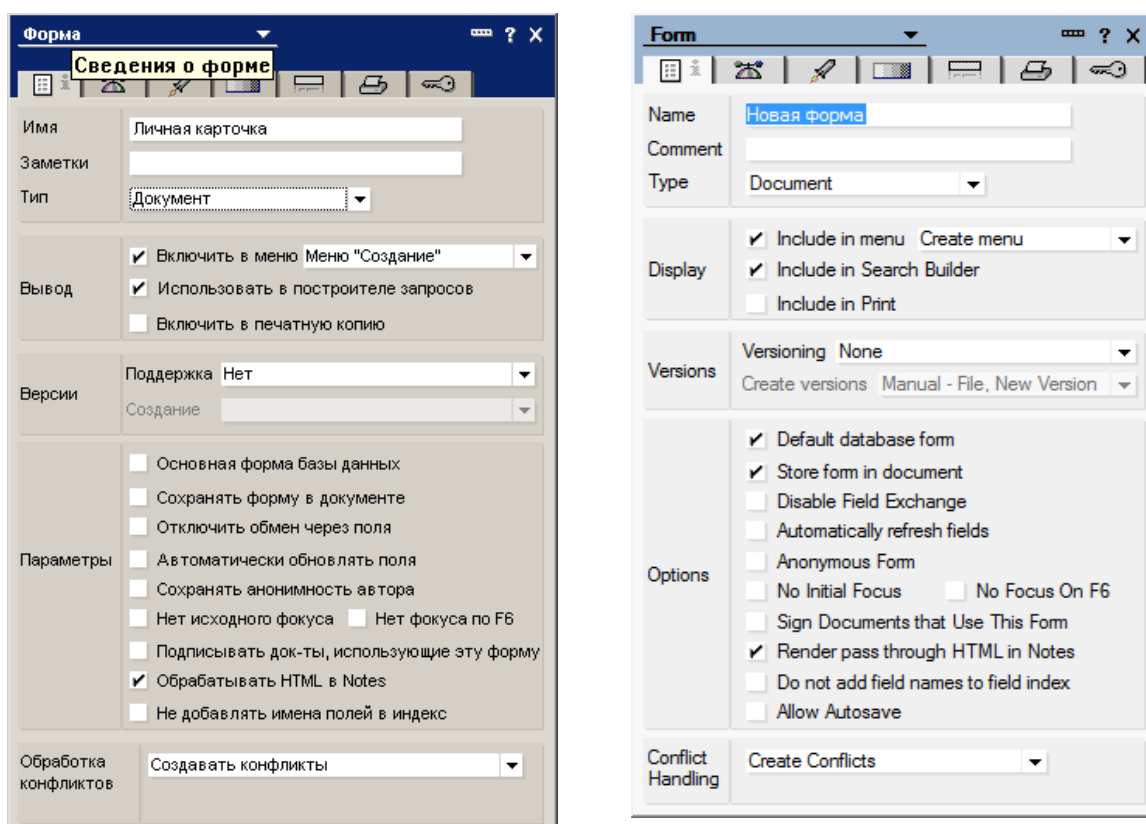


Рис. 6. Определение свойств формы

Флажок «Включить в меню» позволяет включить форму в список форм меню «Создание». В этом случае при подготовке документа пользователь может выбрать форму непосредственно из этого меню.

Список «Поддержка версий» предназначен для выбора варианта создания и сохранения новых версий документа. Рекомендуется выбрать вариант «-Нет-».

Раздел *Параметры* позволяет определить, что происходит при создании, открытии и закрытии документов, использующих данную форму.

Флажок «Основная форма базы данных» – это форма по умолчанию. Форма по умолчанию предназначена для тех случаев, когда Notes не знает, какую форму использовать при работе с документом. Например, при копировании документа из одной базы данных в другую Notes должен привести документ в соответствие с одной из форм текущей базы. Если в текущей базе нет формы, по которой создавался копируемый документ, то Notes будет пытаться отформатировать документ по форме, используемой в текущей базе по умолчанию. Если такая форма отсутствует, Notes не сможет выполнить копирование, поэтому одна из форм базы данных обязательно должна быть основной.

Флажок «Сохранять в документе» – эта опция позволяет сохранять копию формы вместе с документом, созданным по данной форме. Это предотвращает ситуации, когда Notes не может скопировать документы из одной базы в другую из-за отсутствия соответствующей формы.

Предназначение остальных флажков выходит за рамки данной лабораторной работы, их следует оставить без изменения.

Флажок «Обработка конфликтов» дает возможность настроить многопользовательскую работу с одним документом. Это становится возможным, если в документе разными пользователями были отредактированы разные поля. Если разные пользователи отредактировали одно поле, один из документов сохраняется как конфликтный.

После задания атрибутов формы ее нужно сохранить, выбрав из меню команду *Файл/Сохранить*. В результате выполнения этой команды имя формы появится в списке форм папки *Конструктор/Формы* на закладке базы данных Notes.

Редактирование созданных форм осуществляется по двойному щелчку мышью на ее названии в списке форм папки *Конструктор/Формы*.

Проектирование элементов формы

Вся информация в документе делится на статическую и динамическую. Проектирование формы нужно начать с размещения статического (неизменяемого) текста. Статическая информация – это ***надписи***, которые делают форму более удобной в использовании. Обычно это названия полей, заголовочная неизменная часть документа или некоторая справочная информация. Эта информация вносится разработчиком и в дальнейшем недоступна для редактирования пользователем при заполнении формы документа. К надписям можно применять обычные операции форматирования – выравнивание, задание цвета, типа и размера шрифта. Атрибуты статического текста можно задать в меню *Текст/Свойства текста*.

Динамическая информация создается с помощью полей. Разработчик определяет формат вносимой в поля информации. В результате проектирования форма документа должна иметь следующий вид (рис. 7).

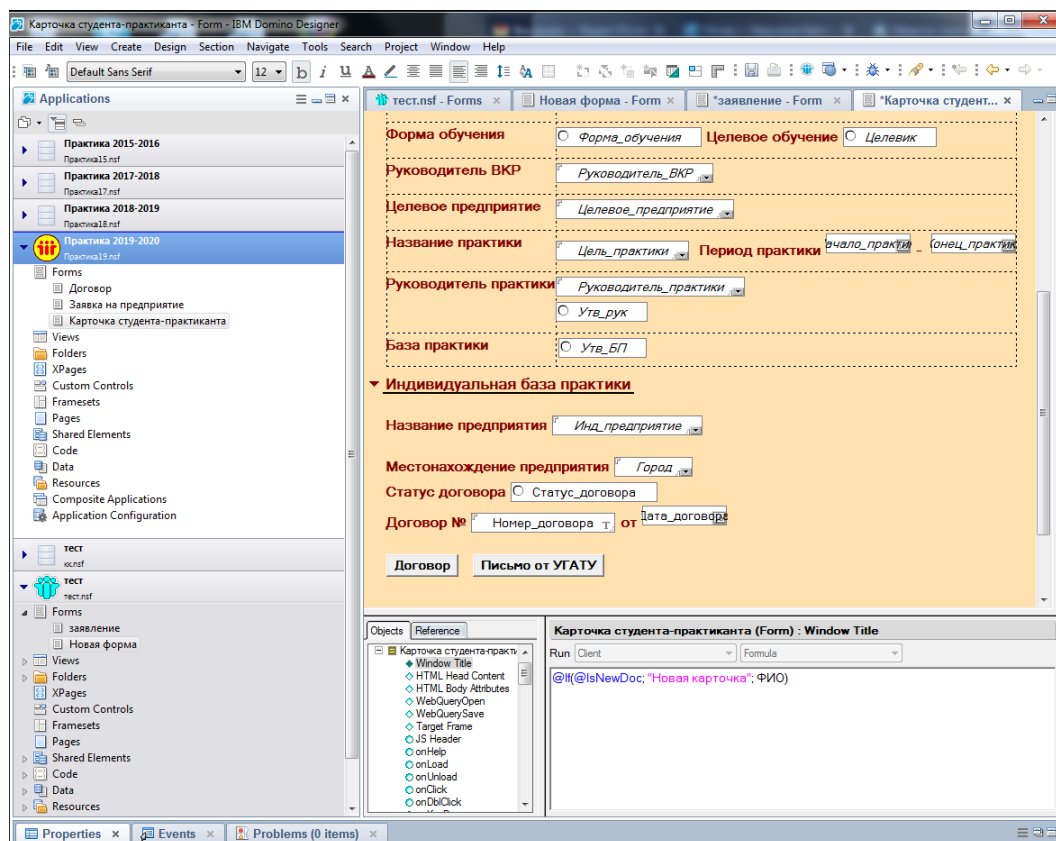


Рис. 7. Форма документа с различными элементами

Создание поля

Поля – это элементы формы, предназначенные для ввода и отображения данных различного типа. Поля могут находиться в любом месте формы. Они могут быть уникальными для данной формы или общими для нескольких форм базы данных. Общие поля удобно создавать отдельно от какой-либо формы, а затем вставлять в нужные формы. Можно зашифровать данные в полях, добавить электронную подпись, использовать скрытые поля для вычисления данных. Полям можно назначить определенные текстовые атрибуты (цвет, размер и разновидность шрифта), чтобы сделать данные более наглядными.

Подведите курсор к тому месту формы, где планируете разместить поле. Выполните команду *Создание/Поле (Create/Field)*. Откроется диалоговое окно *Свойства поля* (рис. 8):

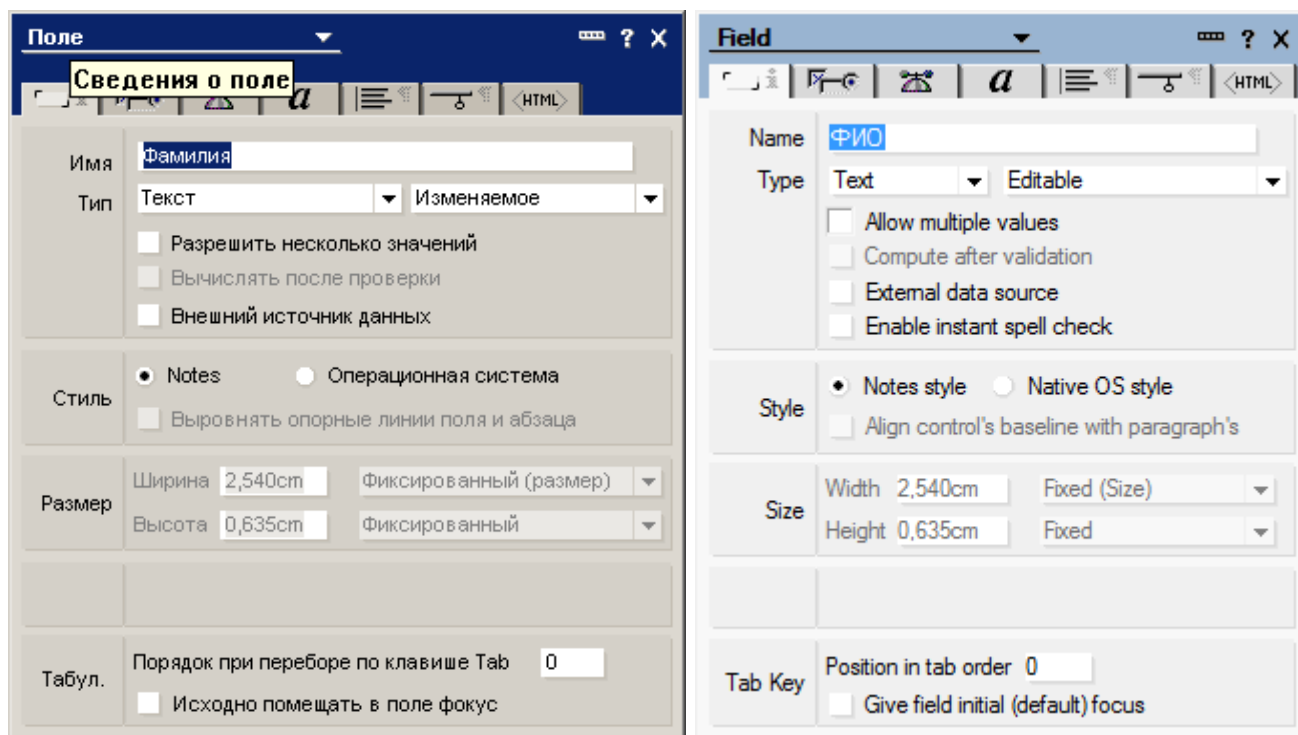


Рис. 8. Окно для задания свойств поля

Атрибуты поля

На вкладке «Сведения о поле» можно задать самые общие свойства поля:

Имя – название поля (можно использовать буквы, цифры, символ «_», нельзя использовать пробел). Имена полей в формах, используемых в текущей базе данных, должны быть уникальны.

Тип включает два параметра: тип данных, содержащихся в поле (по умолчанию «Текст») и тип самого поля, определяющий способ получения значения поля (по умолчанию «Изменяемое»).

Рассмотрим более подробно различные варианты типов полей:

– **Текст** – содержит текстовую информацию, допускается ввод любых символов;

– **Время/дата** – отображает время/дату в заданном стиле (Notes или в виде стандартного календаря);

– **Число** – содержит числовую информацию в заданном формате;

– следующие пять типов полей относятся к *перечисляемым* – в полях такого типа содержится несколько значений, которые заранее задаются разработчиком. Пользователь в дальнейшем при заполнении формы выбирает нужные значения нажатием кнопки мыши.

Значения поля задаются разработчиком в окне *Свойства поля* на вкладке «Управление (Control)» (рис. 9). При этом разработчик может выбрать один из 5 возможных вариантов определения списка перечисляемых слов (раздел *Значения*).

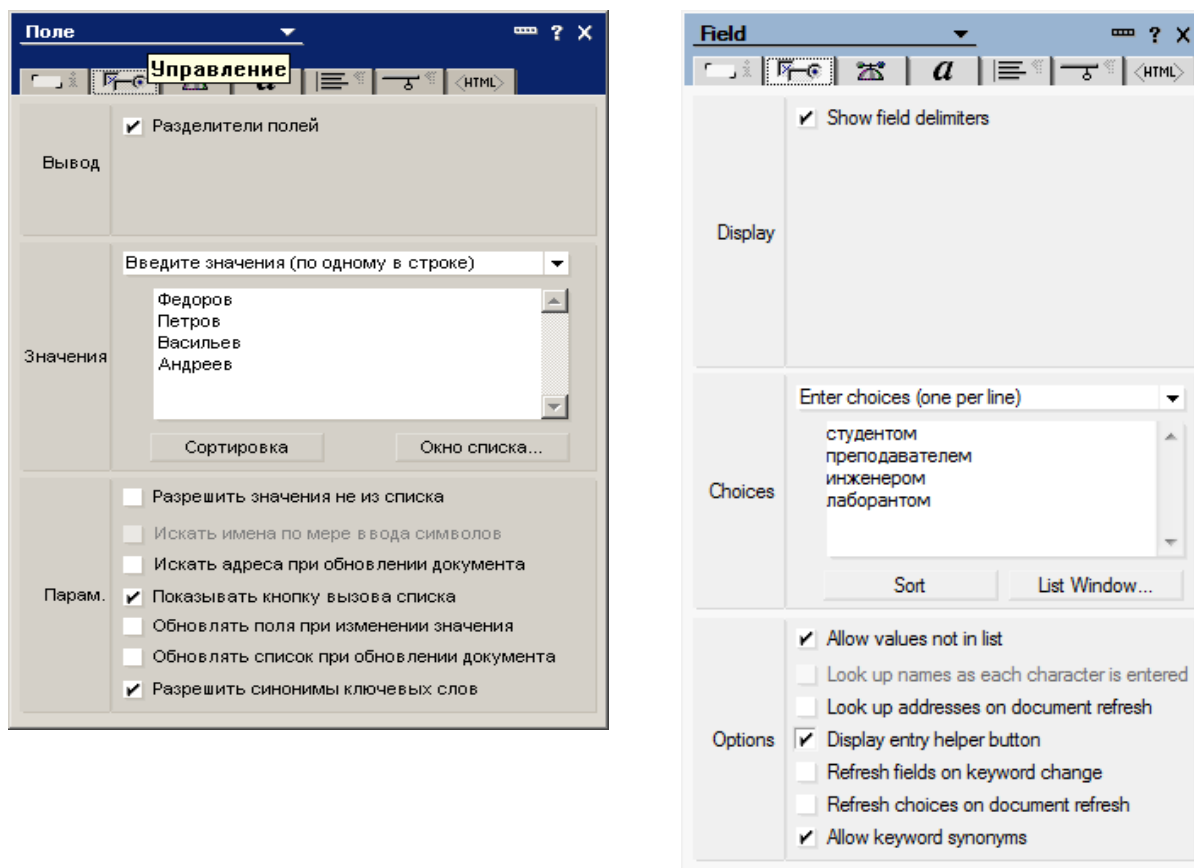


Рис. 9. Вкладка «Управление» для задания атрибутов поля

При необходимости можно разрешить пользователю вводить значения не из списка, включив соответствующий флажок. Рассмотрим различные форматы отображения информации в зависимости от типа поля:

– **Список с окном (*Dialog list*)**. После выбора такого варианта в форме около названия поля появится кнопка с треугольником. При нажатии этой кнопки откроется окно со списком перечисляемых слов (рис. 10);

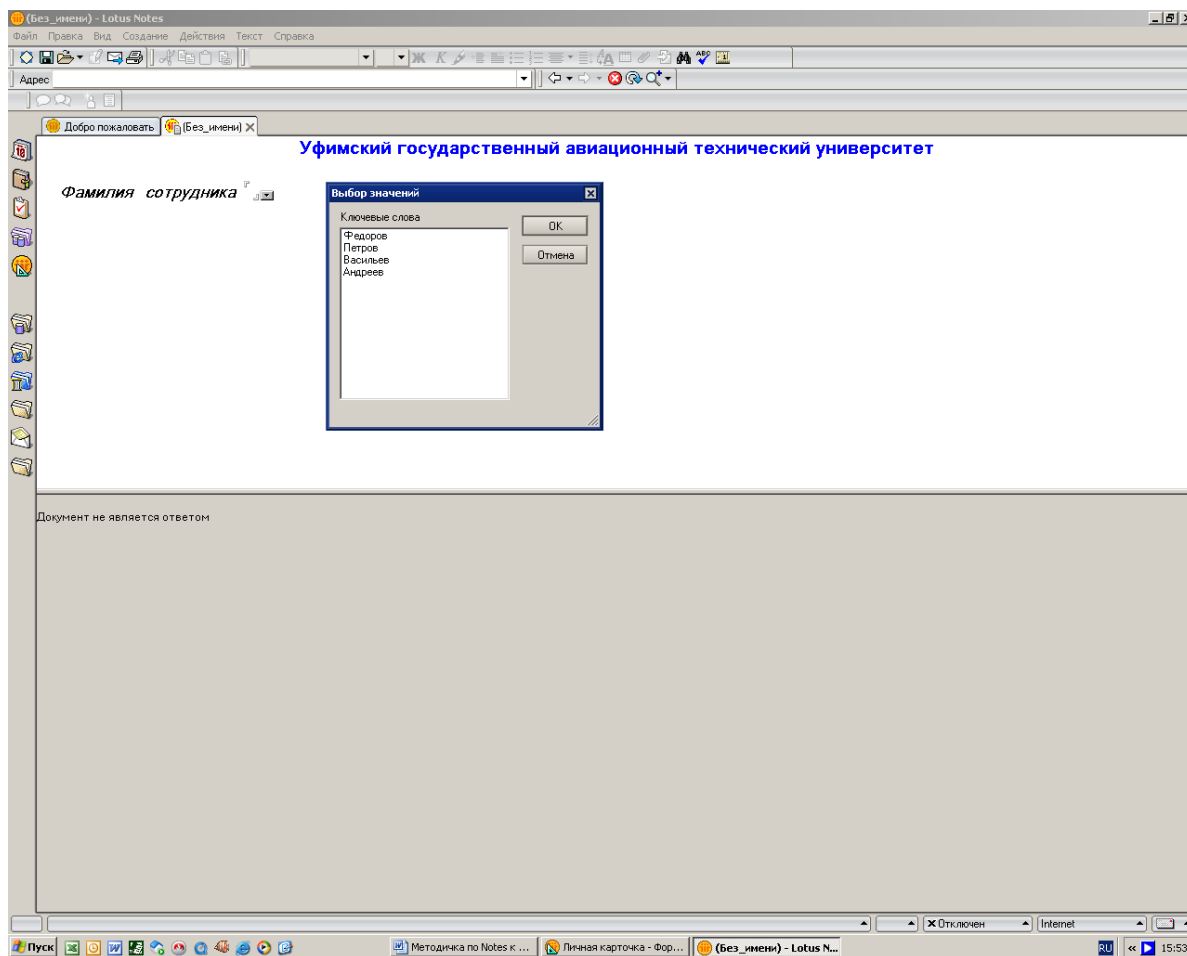


Рис. 10. Вид документа, заполняемого пользователем, при выборе типа поля **Список с окном**

– **Набор флажков (*Checkbox*)**. Выбор этого варианта позволяет в зависимости от ситуации включать или выключать опции (может быть выбрано несколько значений);

– **Набор переключателей (*Radio button*)**. Позволяют выбрать одну из нескольких взаимоисключающих возможностей;

– **Список (*Listbox*)**. Аналогичен набору флажков, но значения отображаются в поле с помощью скроллинга (по одному);

– **Поле со списком (*Combobox*)**. Аналогичен набору переключателей, но значения в поле отображаются по одному;

– **Авторы** – содержит список имен, групп, ролей и определяет, кто может редактировать документ, созданный по данной форме;

– **Имена** – обеспечивает при сохранении документа преобразование введенных имен в более полный формат, позволяющий использовать эти имена в почтовых адресах;

– **Читатели** – содержит список имен, групп, ролей и определяет, кто может читать документ, созданный по данной форме.

Формат отображения информации для каждого из типов полей определяется на вкладке *Управление* (см. рис. 9).

Кроме типа данных справа необходимо указать тип самого поля, т.е., каким образом будет определяться значение поля. Возможны следующие варианты:

– **Изменяемое (*Editable*)** – указывает, что значение поля должно быть введено пользователем вручную;

– **Вычисляемое (*Computed*)** – указывает, что значение поля вычисляется при сохранении документа, а также при выполнении команды *Вид/Обновить*;

– **Вычисляемое при открытии (*Computed for display*)** – указывает, что значение поля вычисляется каждый раз при открытии документа в режиме редактирования или чтения. Такие поля используются для отображения данных, зависящих от текущего состояния документа, например, от результатов промежуточных вычислений;

– **Вычисляемое при создании (*Computed when composed*)** – указывает, что значение поля вычисляется один раз при создании документа. Этот тип полей используется, если начальные значения полей должны оставаться постоянными.

Флажок «*Разрешить несколько значений*» позволяет указать, что поле может иметь несколько значений (не все типы полей позволяют вводить несколько значений).

Для повышения степени дружелюбности интерфейса рекомендуется задавать атрибут «*Справка по полю*» на вкладке «*Дополнительно*» (рис. 11).

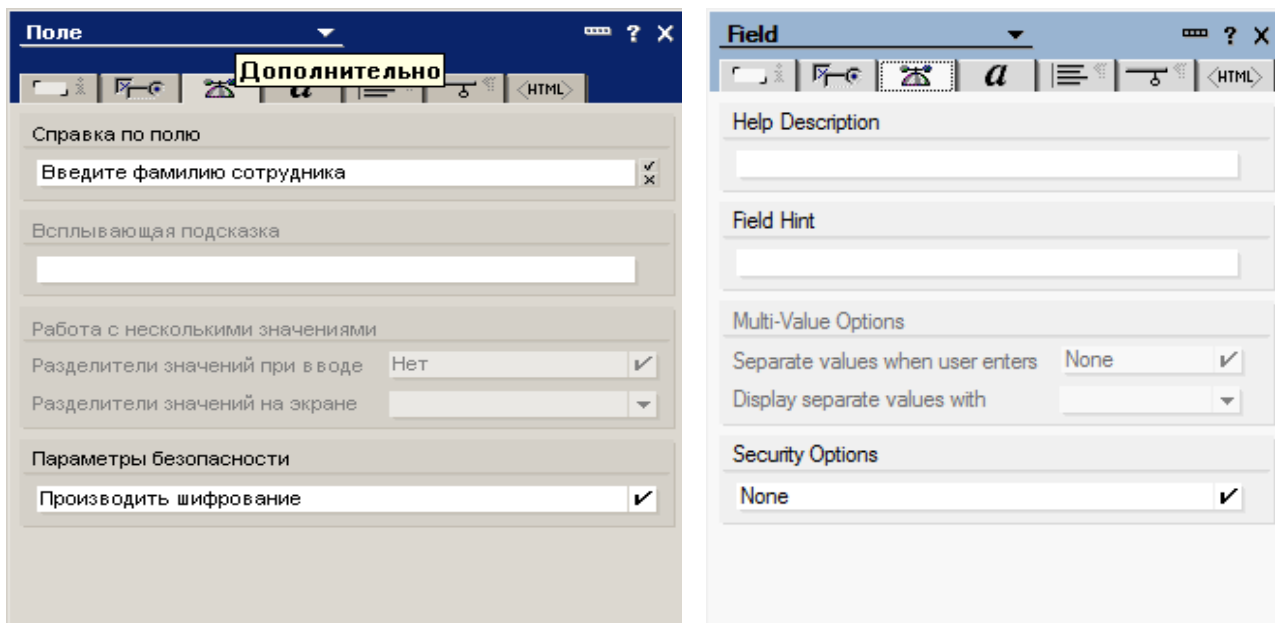


Рис. 11. Задание справки по полю

В результате при заполнении поля пользователем в нижней части экрана будет отображаться введенная фраза в качестве строки подсказки. Это может быть информация о назначении и способе заполнения поля. Для отображения строки подсказки на экране в IBM Notes в режиме редактирования выполните команду *Вид/Отобразить/Справка по полям (View/Show/Field Help)*.

Создание подформы

Подформы позволяют объединить постоянно используемые элементы формы в некоторую совокупность, доступную для разработки других форм. Подформу можно добавить в форму при ее проектировании. Использование подформ позволяет экономить время на разработку форм. Любые изменения в полях подформ приводят к соответствующим изменениям в формах, где они используются.

Создание подформы аналогично созданию формы. Выполните следующие действия:

1. *Создание/Конструктор/Подформа*. Откроется окно проектирования подформы, для создания которой используются те же средства, что и для создания формы.

2. Задайте атрибуты подформы – *Конструктор/Свойства подформы* (рис. 12).

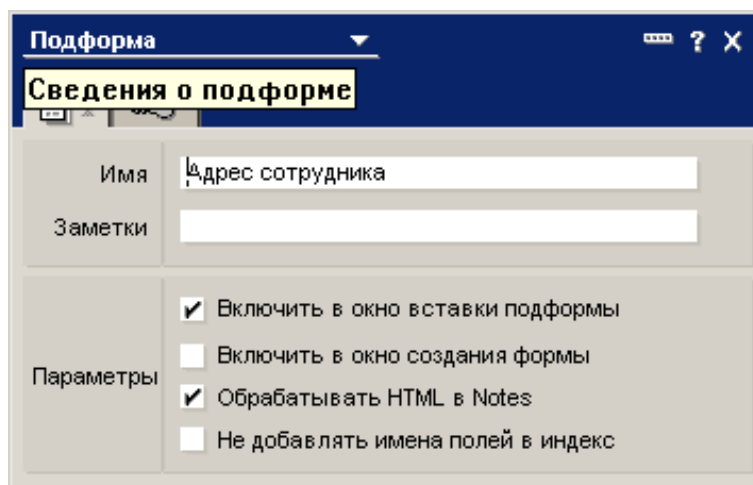


Рис. 12. Свойства подформы

Флажок «Включить в окно вставки подформы» – указывает, что созданная подформа включается в список ресурсов, который появляется в окне «Вставка подформы». Это окно открывается в результате выполнения команды *Создание/Ресурс/Добавить подформу*.

Флажок «Включить в окно создания формы» – указывает, что при создании новой формы по команде *Создание/Конструктор/Форма* автоматически должно открываться окно «Вставка подформы».

После того, как подформа создана, ее можно использовать при разработке или редактировании форм текущей базы данных. Находясь в окне проектирования формы, укажите курсором позицию, где должна находиться подформа. Выполните команду *Создание/Ресурс/Добавить подформу*. Для изменения расположения подформы в форме пользуйтесь буфером обмена.

Создание раздела

Разделы используются в формах с большим количеством полей и статического текста, затрудняющих просмотр формы на экране. Разделы можно сделать свертываемыми и разворачиваемыми, что позволяет сделать форму более компактной и удобной для просмотра. Можно создать раздел с ограниченным доступом.

Для создания обычного раздела выполните следующие действия:

1. Откройте нужную форму в текущей базе данных.
2. В форме выделите текст, поля и другие элементы, которые будут входить в раздел.

3. Выполните команду *Создание/Раздел/Обычный*. В результате создается раздел с заголовком, соответствующим первой строке выбранного фрагмента формы. Если поместить курсор в пустую строку и выполнить ту же команду, будет создан раздел с заголовком «Раздел без имени».

Для редактирования раздела в главном (верхнем) меню появляется пункт «Раздел».

1.3.3. Создание документа «О базе данных»

Каждая база данных должна содержать документ, позволяющий получить определенные сведения о ней – какого типа информация содержится в базе данных, кому она предназначена, кем разработана. Для создания подобного документа сделайте следующие действия:

- откройте базу данных в Domino Designer;
- в области навигатора выберите *Другие/Ресурсы базы данных*;
- создайте документ «О базе данных».

При открытии базы данных пользователем IBM Notes именно этот документ должен отображаться в первую очередь. Для этого в Domino Designer выберите команду *Файл/База данных/Свойства* (рис. 13).

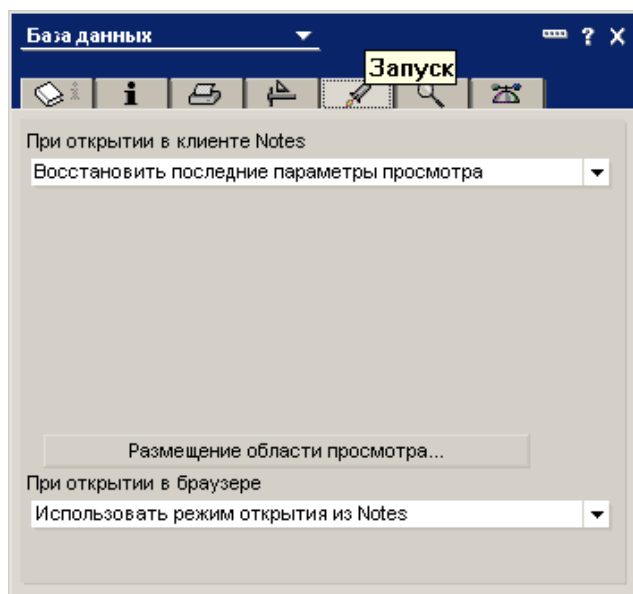


Рис. 13. Окно свойств базы данных

Затем на вкладке «Запуск» выберите из списка «*При открытии в клиенте Notes*» пункт «*Открыть документ «О базе данных»*».

1.3.4. Создание закладки и значка БД

Каждая база данных должна иметь значок, который помогает быстро отыскать ее в рабочем пространстве. Для этого служит закладка базы данных. Закладка содержит псевдоним базы данных, который по умолчанию совпадает с ее названием, и значок. Можно отредактировать значок, который присваивается по умолчанию, или создать новый.

Для редактирования существующей закладки сделайте следующие действия:

- щелкните правой кнопкой мыши в области навигатора вашей базы данных и выберите команду «*Переименовать закладку...*»;
- убедитесь, что название базы данных, имя файла и закладка разные;
- щелкните правой кнопкой мыши в области навигатора вашей базы данных и выберите команду «*Сменить значок закладки...*»;
- в появившемся окне (рис. 14) вы можете выбрать картинку из предложенного списка, а можете создать свою в GIF-, JPEG- или BMP-формате;

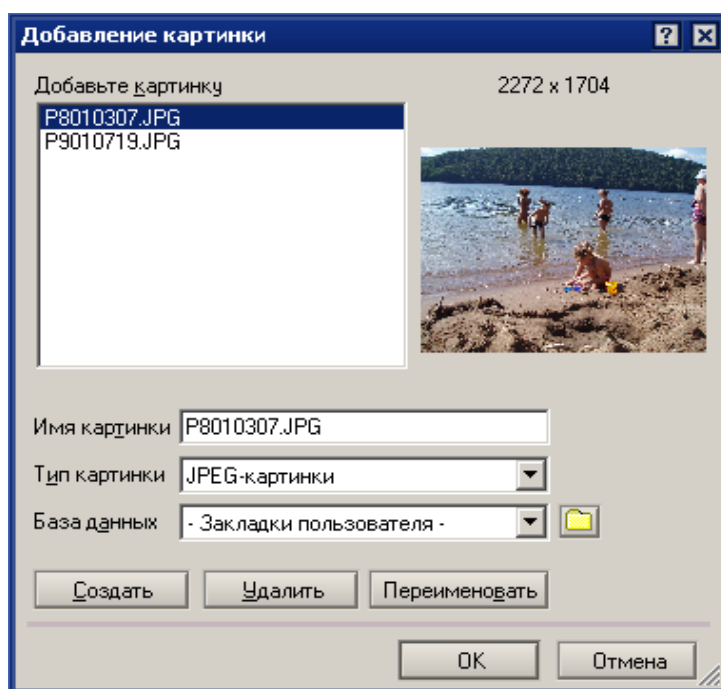


Рис. 14. Окно создания значка базы данных

- разместите вашу закладку на «Панели закладок», перетаскив ее мышкой (область панели закладок – узкая серая полоска – располагается слева от области навигатора).

1.3.5. Заполнение форм (создание документов)

Спроектированные и сохраненные формы предназначены для создания документов пользователями. Заполнение форм возможно только в режиме IBM Notes при выполнении команды *Создание*/*<Имя формы>* (рис. 15).

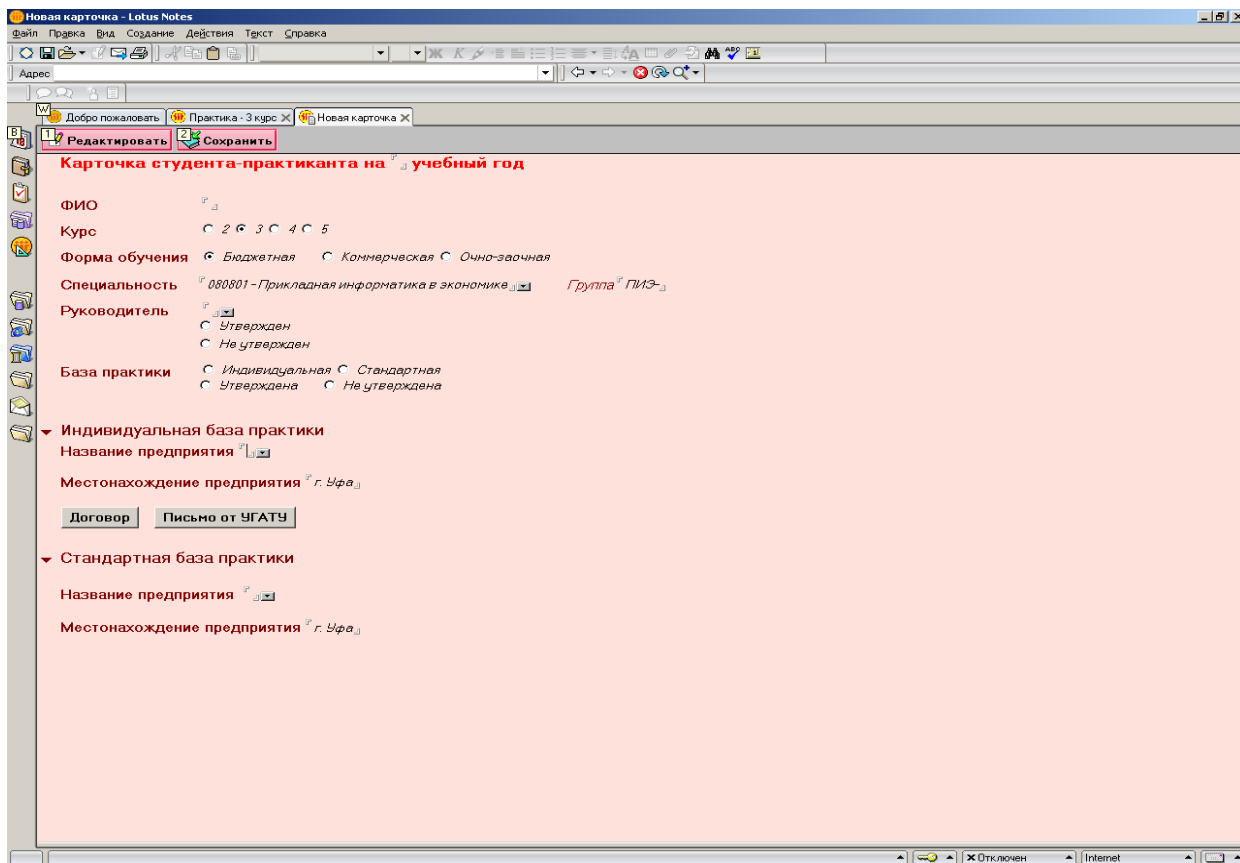


Рис. 15. Вид документа, заполняемого по спроектированной форме

1.4. Требования к отчету

1. Созданная в соответствии с вариантом задания база данных должна содержать две формы документов.
2. Формы документов должны различаться по оформлению.
3. В формах должны быть поля следующих типов данных: текст, число, дата, список с окном, набор флажков, набор переключателей.
4. Формат статического текста и формат значений полей должны различаться.
5. В одной из форм документов должна содержаться подформа, в другой форме – раздел.

6. При открытии базы данных в IBM Notes должен открываться документ «О базе данных», в котором указываются фамилии разработчиков, дата создания БД и номер варианта задания.

7. База данных должна иметь эргономичный интерфейс, обеспечивающий пользователю удобство заполнения документов.

Контрольные вопросы

1. Что такое форма документа, какие элементы содержит окно проектирования форм?

2. Из каких элементов состоит форма?

3. Что содержит окно навигатора базы данных?

4. Какие типы данных могут содержать поля формы?

5. Для чего предназначены подформа и раздел?

ЛАБОРАТОРНАЯ РАБОТА № 2

ПРОЕКТИРОВАНИЕ ПРЕДСТАВЛЕНИЙ И ПАПОК

2.1. Цель работы

Целью работы является изучение особенностей разработки представлений базы данных с использованием программного средства Domino Designer.

2.2. Теоретические сведения

Представления и папки – это определенным образом организованные списки документов базы данных Notes, с краткой информацией о содержимом этих документов. Каждая база данных должна иметь, по крайней мере, одно представление. Представление документов может содержать список всех документов базы или только некоторой части документов, отобранных в представление по определенному критерию. По сути, представление – это результат фильтрации всех документов, содержащихся в базе данных, по группе признаков.

В представлениях и папках имеются два главных элемента:

– *строки*, отображающие документы и категории (*категория* – это слово, фраза или число, используемая для группировки документов в представлении);

– *столбцы*, отображающие определенные поля из документов.

Какая конкретно информация из документов, и в каком виде она отображается, зависит от того, как спроектированы представления.

В представлениях не могут отображаться данные из форматируемых полей, а также графические объекты.

Процесс проектирования представления включает следующие этапы:

- 1) определение свойств представления;
- 2) задание критериев отбора документов;
- 3) создание столбцов и определение их свойств;
- 4) определение содержимого столбцов.

2.3. Порядок выполнения работы

Прежде чем приступить к выполнению лабораторной работы, необходимо создать документы (не менее 3-х по каждой форме) в базе данных, разработанной на предыдущей лабораторной работе.

На рис. 16 приводится вид базы данных IBM Notes с набором представлений (слева в области навигатора) и перечнем документов, принадлежащих одному из представлений.

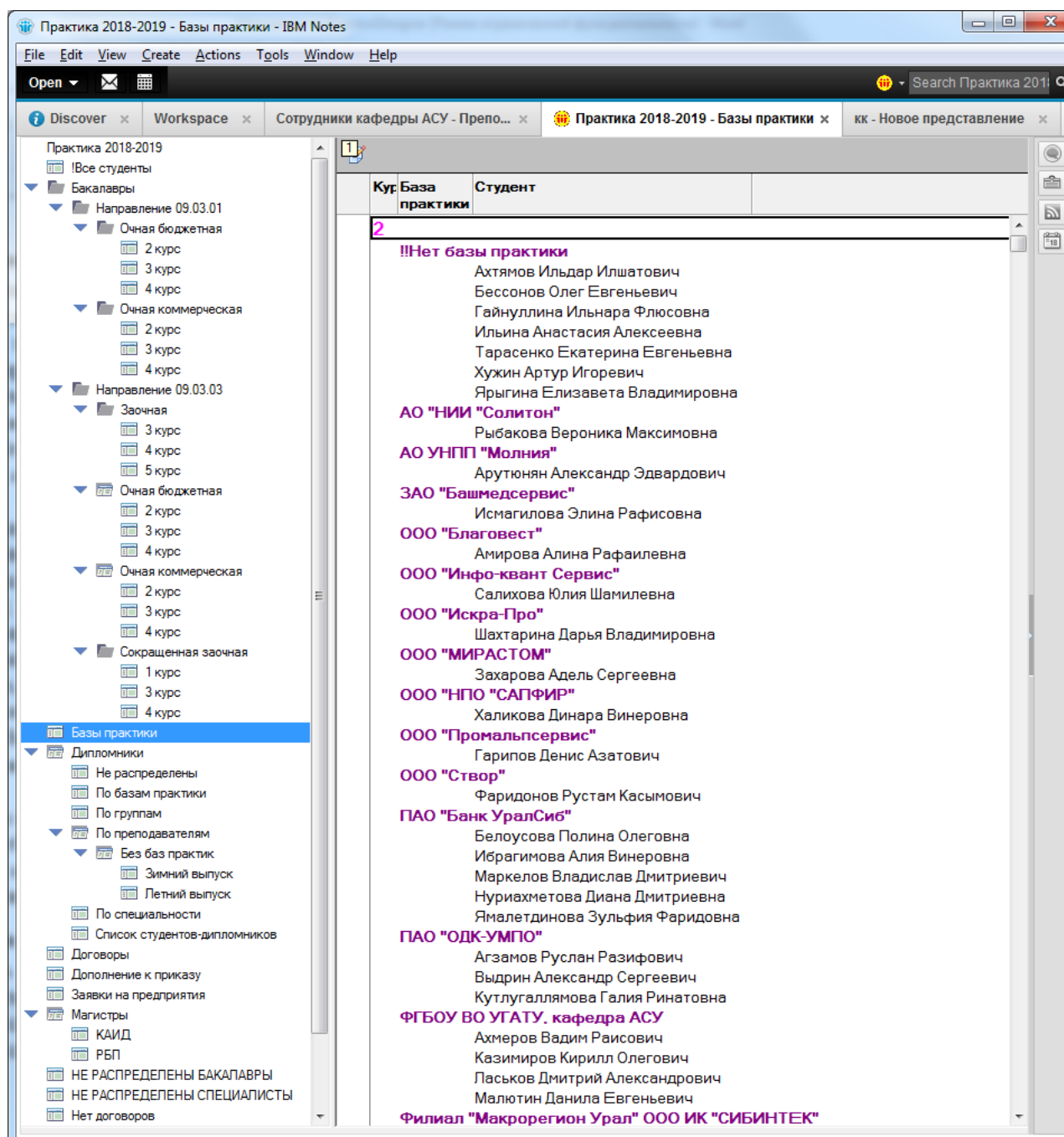


Рис. 16. Вид базы данных с набором представлений

На рис. 17 приведен вид того же представления в IBM Domino Designer.

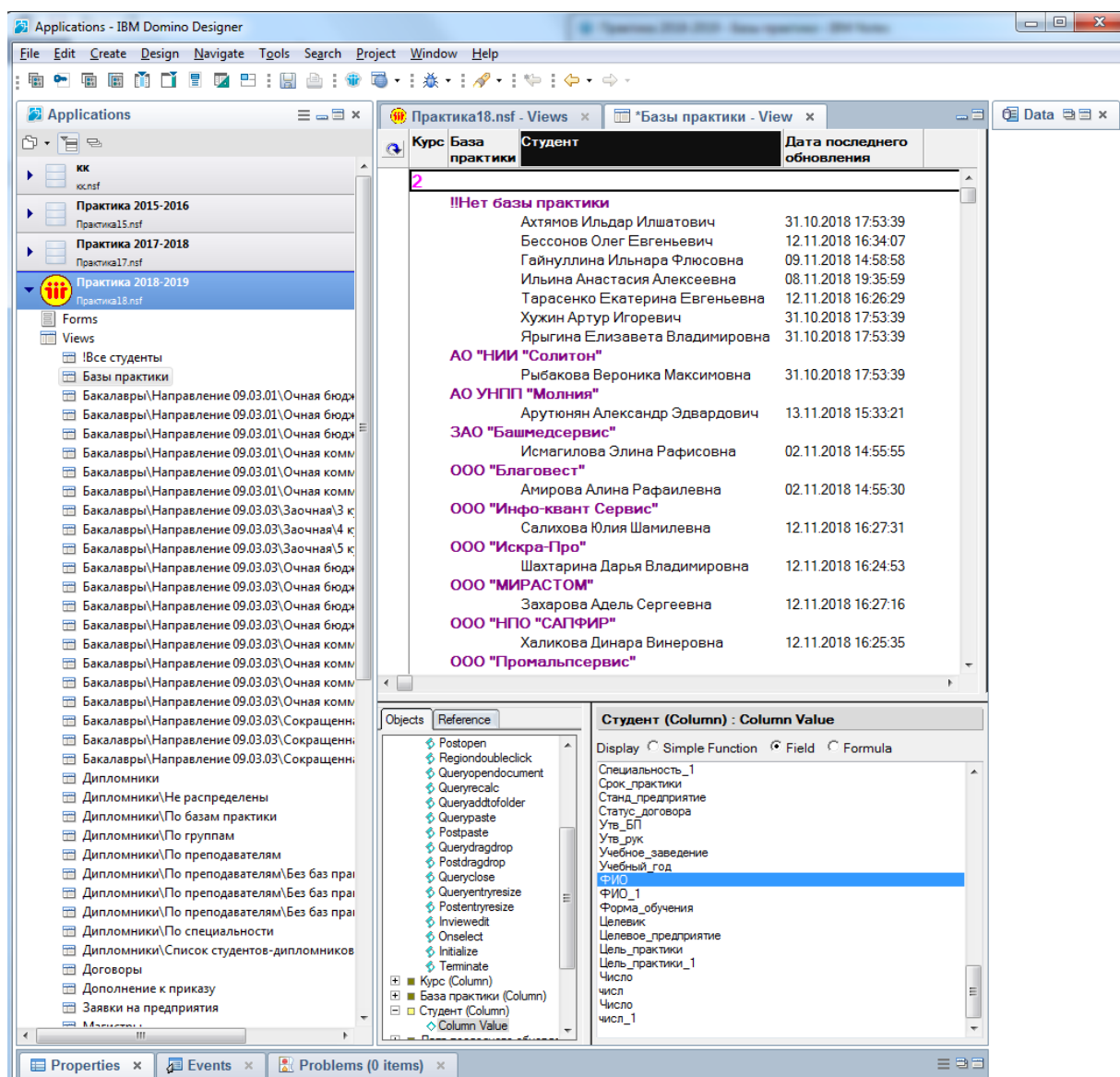


Рис. 17. Вид представления в IBM Domino Designer

В ходе лабораторной работы необходимо:

1. Отредактировать представление *Untitled* (Без имени), созданное в базе данных по умолчанию.
2. Спроектировать 3 новых представления базы данных таким образом, чтобы одно из них было вложено в другое.
3. В каждом представлении должно быть не менее 4–5 столбцов.
4. В каждом представлении документы должны быть сгруппированы по 2–3 признакам (как на рис. 16).

2.3.1. Определение свойств представления

Откройте в IBM Domino Designer разработанную вами базу данных (см. лабораторную работу № 1).

Выберите в области навигатора базы данных строчку *Views* (*Представления*). На экране появится строчка с параметрами представления, создаваемого в базе данных по умолчанию (рис. 18). Звездочка слева от имени представления означает, что это представление по умолчанию, его удалить нельзя.

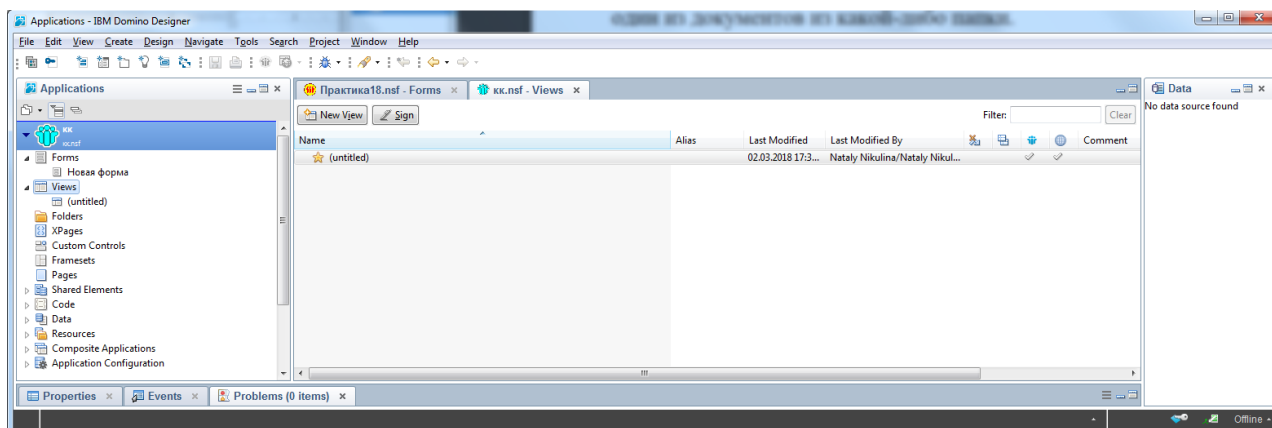


Рис. 18. Представление базы данных по умолчанию

Это представление *Untitled* (*Без имени*), в котором содержится только один столбец – порядковый номер документа в базе данных. В этом представлении отображаются все документы, имеющиеся в базе данных вне зависимости от формы, в которой они разработаны. На рис. 19 видно, что в базе данных находится 4 документа.

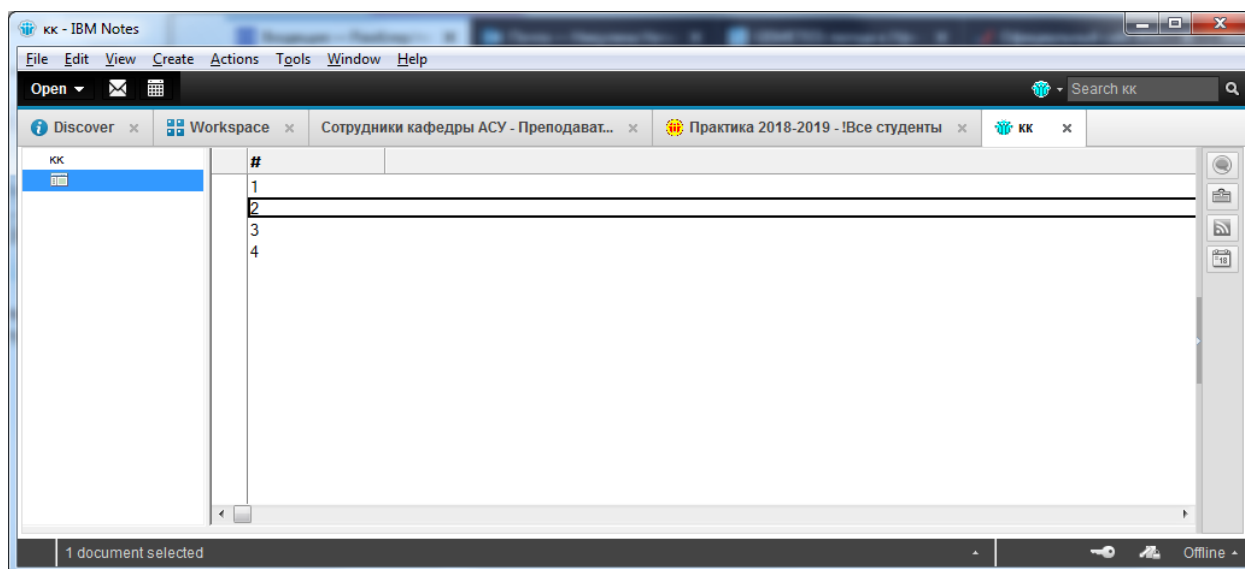


Рис. 19. Вид ненастроенного представления в базе данных

Для редактирования представления в навигаторе дважды щелкните мышкой на его названии и в меню *Design* выберите команду *ViewProperties* (рис. 20). Появится окно свойств представления, аналогичное окну свойств формы (рис. 21).

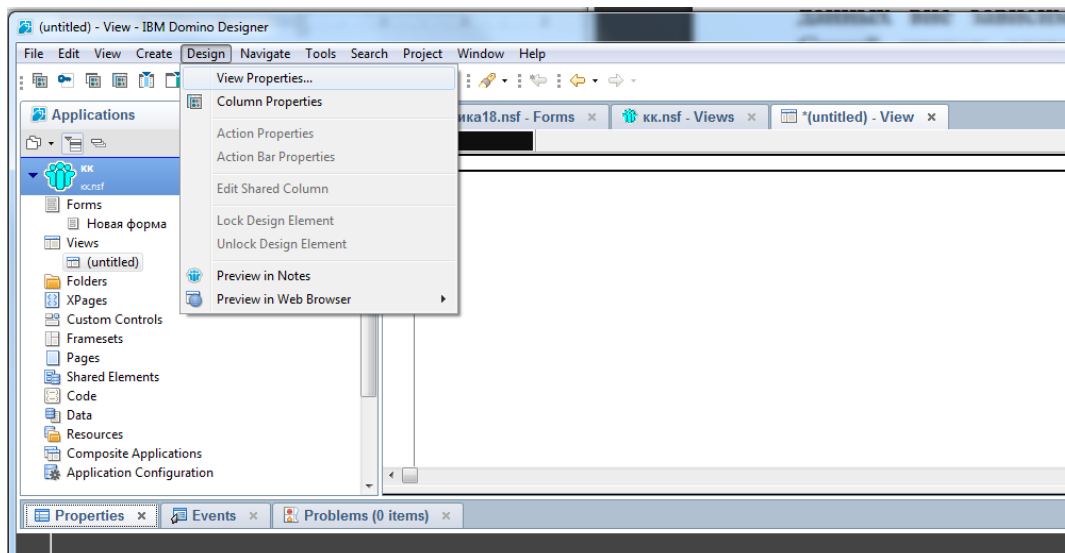


Рис. 20. Редактирование свойств представления

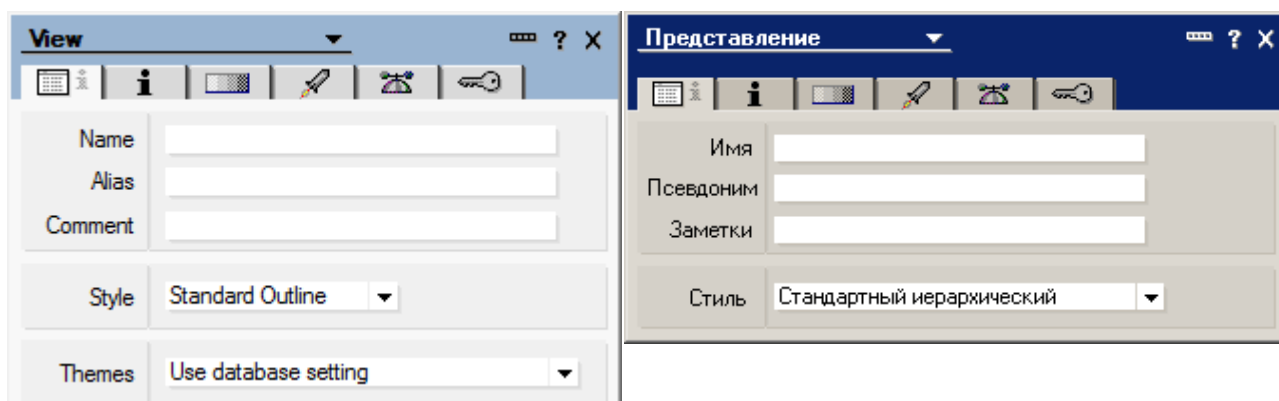


Рис. 21. Окно свойств представления

Окно свойств представления содержит 6 вкладок, в которых содержатся различные атрибуты и параметры представления. Изменения, сделанные в окне свойств представления, распространяются на все представление в целом.

На вкладке «**Сведения о представлении**» задаются имя представления, его псевдоним (служит для задания синонимов имени представления и предназначены только для разработчиков), заметки (для напоминаний разработчику), а также тип представления (стандартный – в виде иерархии документов и в виде календаря).

Вкладка «**Параметры**» содержит следующие флажки (рис. 22):

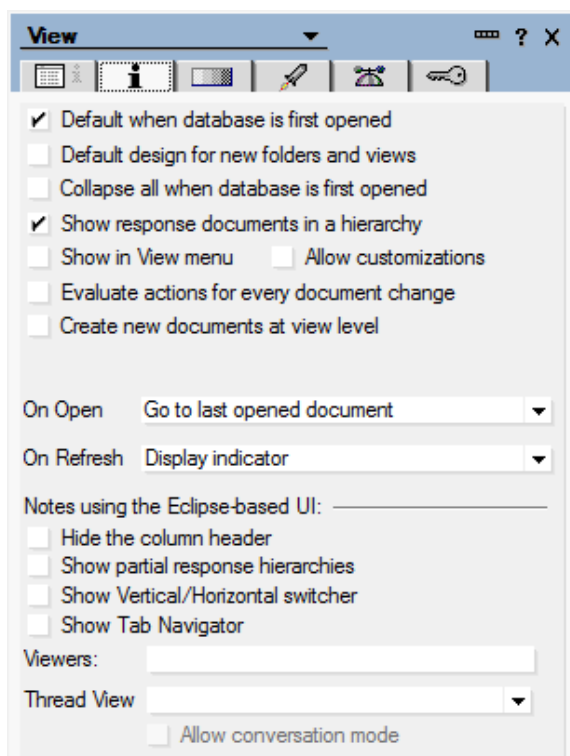


Рис. 22. Вкладка «**Параметры**»

Использовать при первом открытии базы данных – указывает, что это представление по умолчанию, оно загружается по умолчанию при первом открытии базы данных. Каждая база данных должна иметь представление по умолчанию.

Использовать для новых папок/представлений – указывает, что структура данного представления наследуется по умолчанию для создания новых представлений.

Свертывать все при первом открытии базы данных – все категории сворачиваются до самого верхнего уровня, все ответные документы сворачиваются под главными.

Отображать в меню - указывает, что это представление нужно включить в меню *View (Вид)*. Перечень представлений, включенных в меню *View (Вид)* можно увидеть только из IBM Notes.

Также можно задать, какие действия должны происходить при открытии и обновлении базы данных.

Вкладка «**Стиль**» позволяет настроить внешний вид столбцов и отображаемых данных в представлении – цвет, размер и стиль шрифта, высота строк, обрамление.

В дальнейшем к окну «Представление» можно перейти, выполнив команду *Design/ViewProperties*.

Задайте имя представления, установите нужные параметры и перейдите к пункту 3.2.

2.3.2. Создание столбцов представления и определение их свойств

В столбцах представления находится информация, помогающая пользователю определить нужный документ. Одни и те же столбцы могут быть использованы в разных представлениях в одной и той же базе данных.

После задания имени представления откроется окно проектирования представления, аналогичное окну проектирования формы. Оно состоит из двух основных частей (рис. 23).

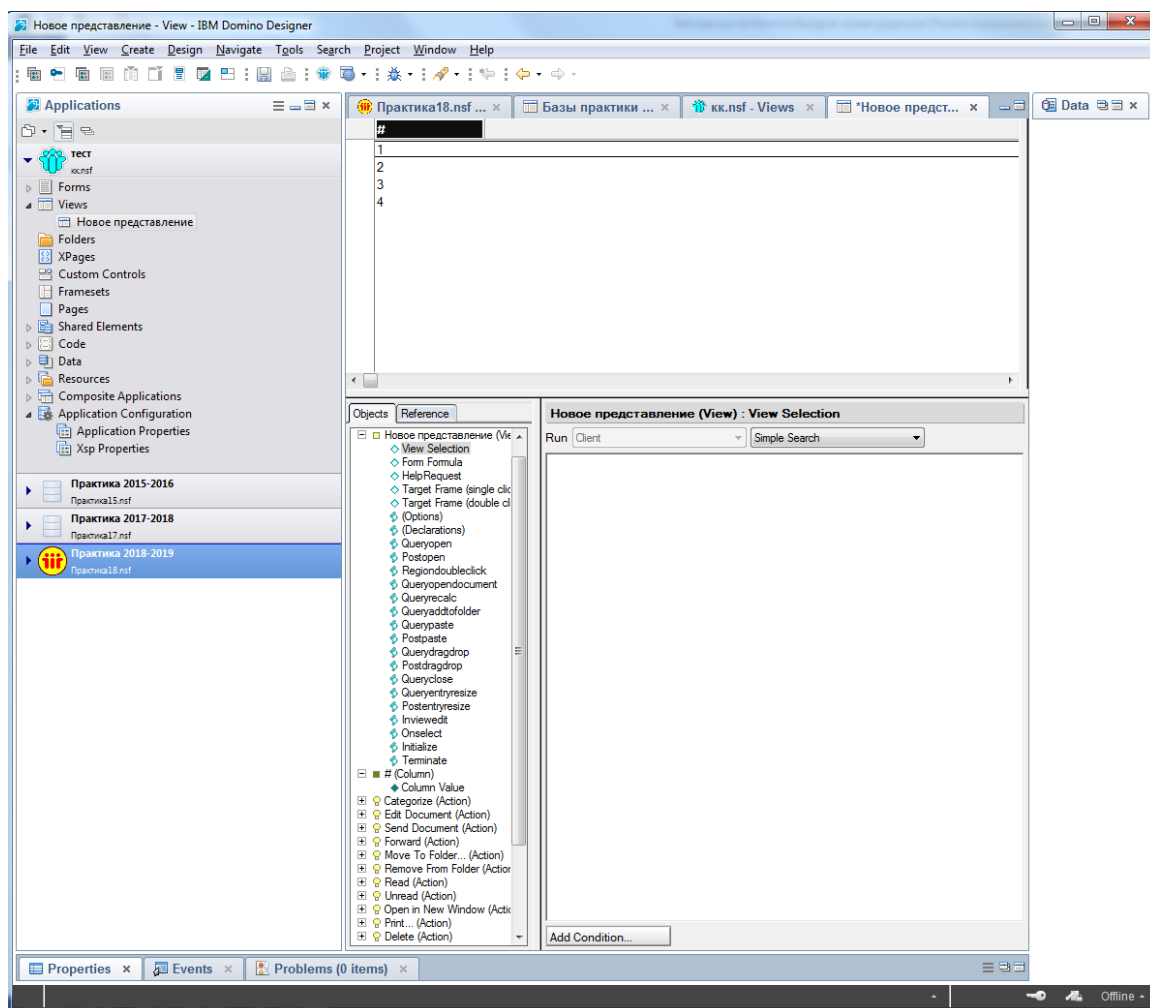


Рис. 23. Окно проектирования представления

В верхней части находится область столбцов представления. В ней можно добавить, переместить или удалить столбец, настроить его размер, задать или изменить атрибуты заголовка столбца и атрибуты текста отображаемых документов. Если в базе уже есть документы, то можно во время проектирования следить за тем, как они отображаются в представлении. Для этого нужно выполнить команду *View/Refresh (Вид/Обновить)* или нажать на синюю круговую стрелку, расположенную в крайнем левом столбце.

В нижней части экрана находится область конструктора, в которой задаются критерии отбора документов и определяются значения столбцов. Область конструктора делится на две части: в левой части отображаются объекты базы данных, списки существующих полей базы данных, функций и команд, правая часть представляет собой поле для записи формул на языке *Lotus Script* или отображения простых функций *Notes*.

Для создания столбцов можно использовать команду *Create/InsertNewColumn (Создание/Вставить столбец)* (столбец добавится перед выделенным в данный момент столбцом) или *Create/AppendNewColumn (Создание/Добавить столбец)* (столбец добавится после выделенного в данный момент столбца) (рис. 24).

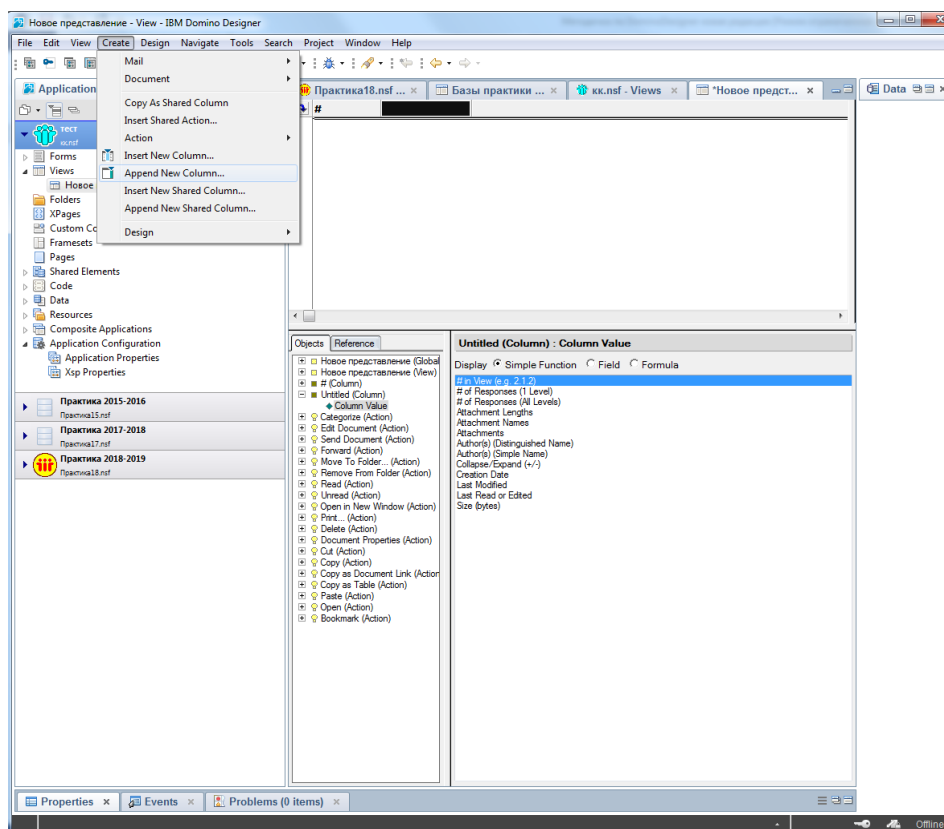


Рис. 24. Добавление столбца

Можно пользоваться контекстным меню под правой клавишей мыши. Для удаления столбца нужно выделить его и нажать клавишу *Del*.

Для определения свойств столбца предназначено окно *Column* (*Столбец*) (рис. 25). Чтобы открыть это окно, выделите столбец и выполните команду *Design/Column Properties* или дважды щелкните мышью по пустому заголовку столбца.

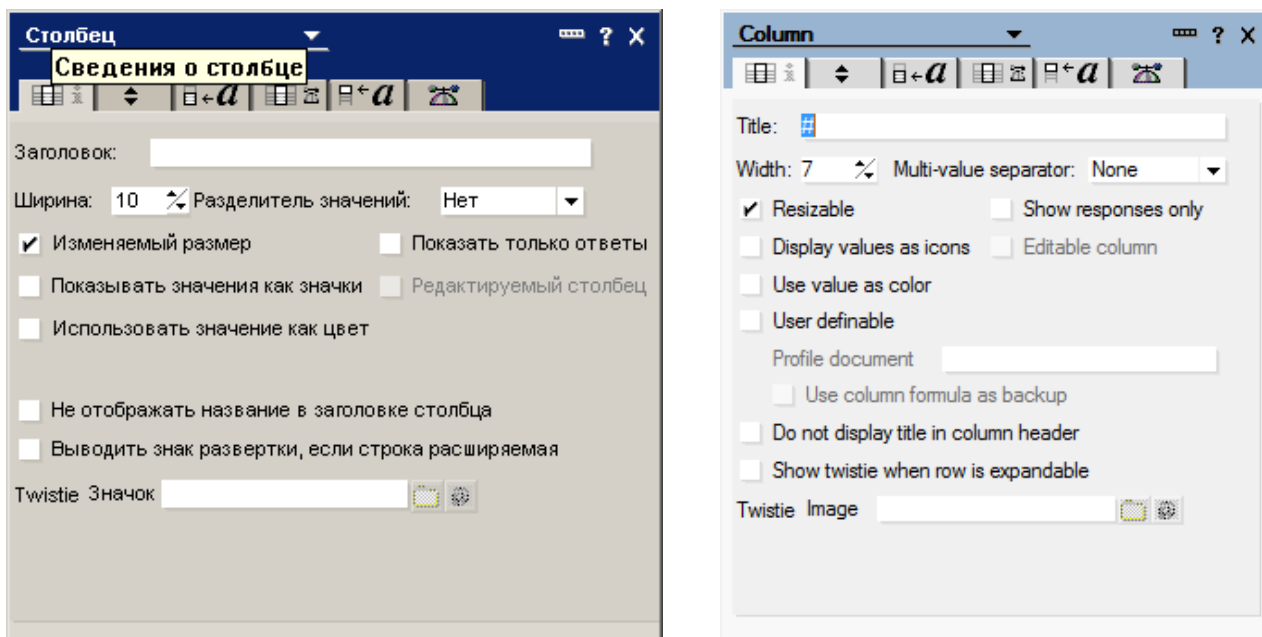


Рис. 25. Окно свойств столбца

Окно свойств столбца имеет 6 вкладок.

На вкладке «**Сведения о столбце**» находятся следующие атрибуты:

– *Заголовок* – заголовок столбца (может и не совпадать с названием поля базы данных или простой функцией Notes, откуда берется информация для заполнения столбца).

– *Ширина* – для установки ширины столбца (можно пользоваться мышью для выравнивания ширины столбца по размеру заголовка).

– *Флажок «Изменяемый размер»* – если не выбрать этот флажок, пользователь при открытии представления не сможет изменить размер столбца мышью.

– *Список «Разделитель значений»* используется для выбора разделителя, если в столбце отображается какой-либо список.

– Флажок «Показать только ответы» – устанавливается только в одном столбце, если в представлении должна отображаться иерархия ответных документов.

– Флажок «Выводить знак развертки, если строка расширяемая» – позволяет поместить в представление специальные значки в виде треугольников, указывающие, что строку можно развернуть или свернуть.

На вкладке «Сортировка» (рис. 26) задаются параметры, определяющие все, что связано с упорядочиванием списков документов в представлении, а также параметры, позволяющие отображать некоторые результаты итоговых вычислений по столбцам.

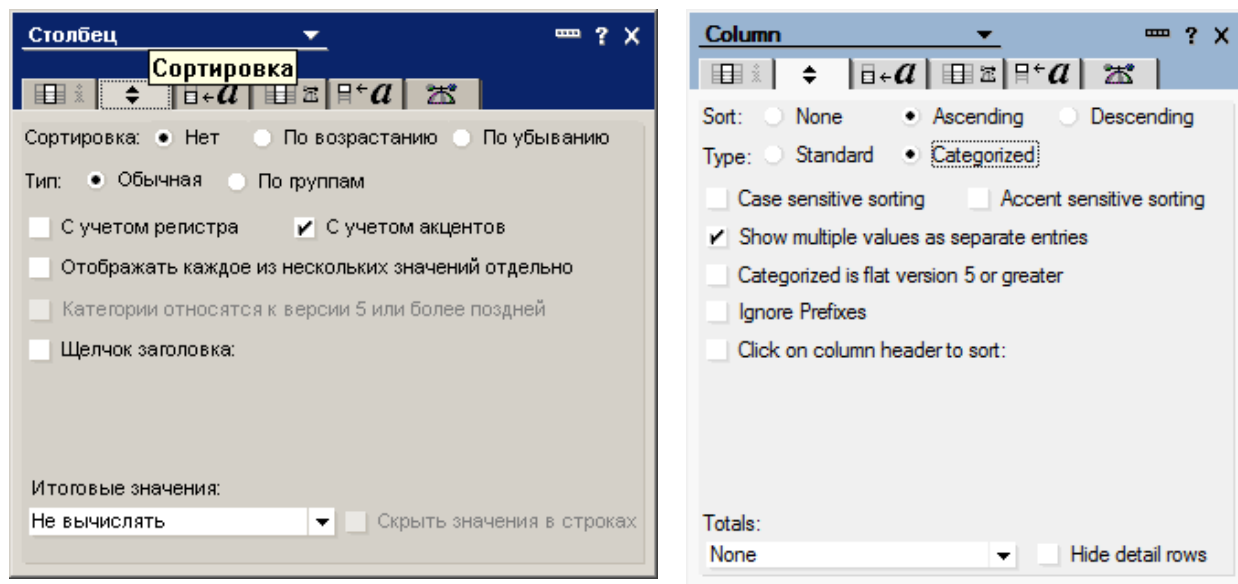


Рис. 26. Сортировка информации в столбце

Требует пояснения флажок «Щелчок заголовка». Установка этого флажка обеспечивает сортировку столбца «на лету». После его установки в заголовке столбца появляется треугольник, при нажатии на который происходит сортировка документов. При установке этого флажка становится доступным список вариантов направления пересортировки: сортировка по возрастанию, сортировка по убыванию, сортировка в обратном порядке, смена представления.

При выборе одного из первых трех вариантов пересортировки становится доступным список *Вторичная сортировка по столбцу*.

При выборе варианта *Смена представления* можно переключаться в другое представление, просто нажав мышью по соответствующему значку.

Вкладка «**Шрифт**» задает *атрибуты текста документов* при отображении содержимого столбцов.

Вкладка «**Заголовок**» позволяет задать атрибуты текста для заголовка столбца.

Вкладка «**Дополнительный формат**» (рис. 27) позволяет настраивать формат вывода данных в соответствии с их типом. Можно настроить дополнительный формат для трех типов данных – «Число», «Дата/время», «Имена».

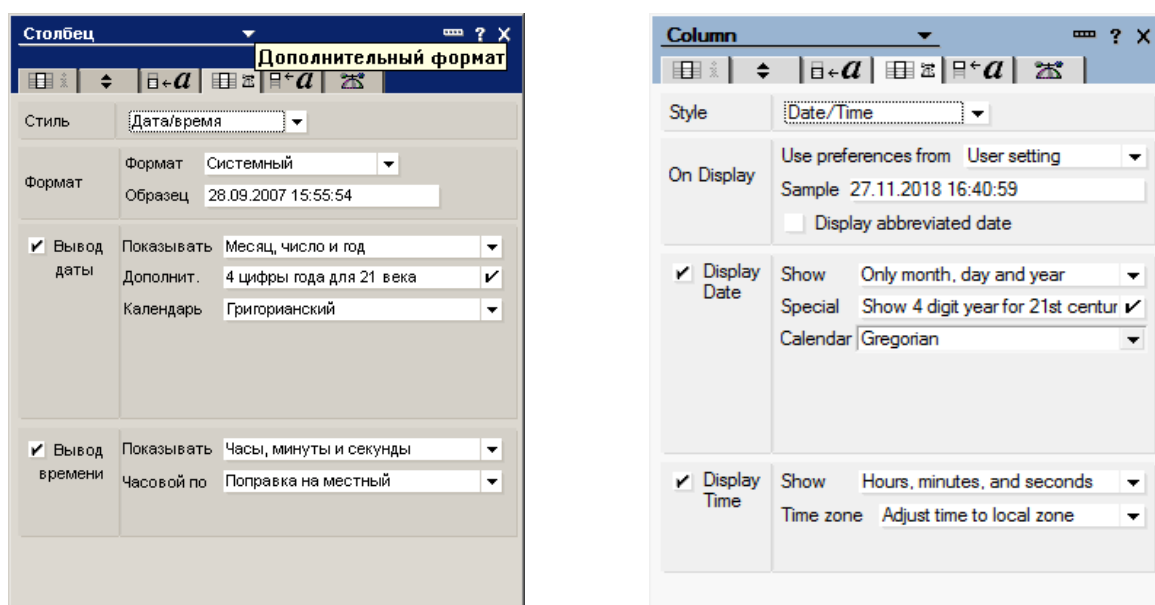


Рис. 27. Настройка формата вывода данных в столбце

На последней вкладке «**Дополнительно**» есть флажки, позволяющие сделать столбец невидимым для пользователя (рис. 28).

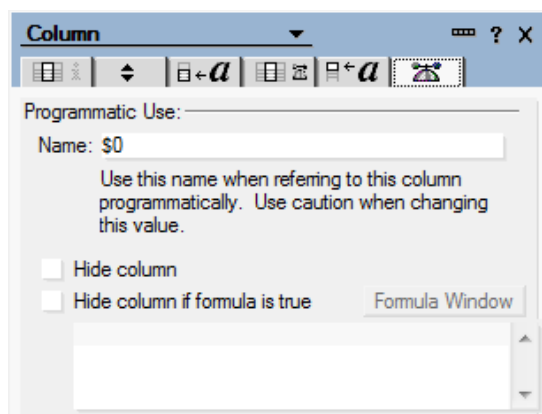


Рис. 28. Вкладка «**Безопасность**»

2.3.3. Определение содержимого столбцов

После создания столбца нужно определить его содержимое (какая информация о документе или из документа будет в нем отображена). Для этого в правой нижней части области конструктора в окне # (*Column*): *Column Value* установите переключатель **Display** в одно из следующих состояний (***) – имя столбца): *Simple Function*, *Field* или *Formula* (простая функция, поле или формула).

Простая функция – в столбце будет отображаться результат выполнения простой функции Notes. При этом в списке в области конструктора появятся 13 простых функций (табл. 1).

Таблица 1

Простые функции Notes

Простая функция	Возвращаемый результат
Автор(ы) (составное имя)	Полное составное имя автора документа, например, Denis Ivanov\ Intertrust corp.\RU
Автор(ы) (простое имя)	Имя автора документа, например, Denis Ivanov
Вложения	Количество вложенных файлов
Дата изменения	Дата и время последней модификации документа
Дата создания	Дата и время создания документа
Длины вложений	Список размеров вложенных файлов
Имена вложений	Список имен вложенных файлов
Порядковый номер	Номер документа в представлении
Последнее чтение/правка	Дата и время последнего доступа
Размер (байт)	Размер документа в байтах
Свернуть/развернуть	Развертываемость категории
Число ответов (1 уровень)	Количество ответных документов 1 уровня
Число ответов (все уровни)	Количество всех ответных документов

Пример отображения в столбце с заголовком # порядкового номера документа, формирующегося с помощью простой функции *Порядковый номер*, приведен на рис. 19.

Поле – в столбце будет отображаться содержимое некоторого поля из имеющейся в базе данных формы. При этом в списке в области конструктора появятся имена всех полей всех форм базы (рис. 29). Выберите название поля, значение которого вы хотите отобразить (достаточно просто указать на нужное поле мышью).

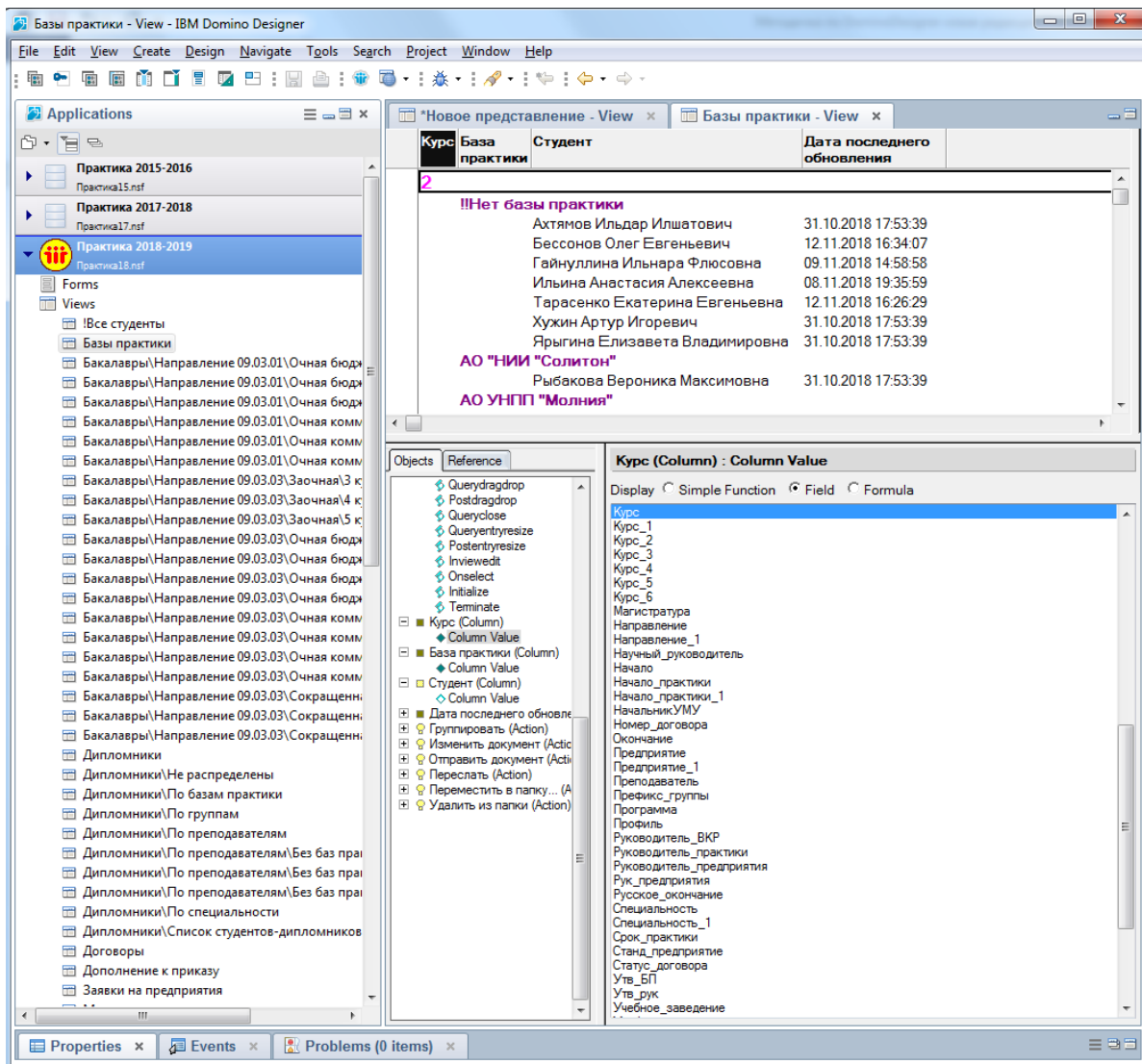


Рис. 29. Определение содержимого столбца как поля из документа

Формула – в этом случае в столбце отразится результат вычисления по некоторой формуле, введенной в том же самом окне (рис. 30).

2.3.4. Задание критериев отбора документов в представление

При создании представления в нем по умолчанию будут отображаться **все** документы базы данных. Однако можно задать один или несколько критериев отбора документов для отображения их в представлении. Критерий отбора документов определяется в области конструктора в левом нижнем углу окна проектирования представления – в небольшом окне *Objects & Reference (Объекты и списки)*.

Выберите вкладку *Objects*, далее – строчку *View Selection (Отбор документов)*. В правой нижней части области конструктора появится окно ***** (View): View Selection** (рис. 31).

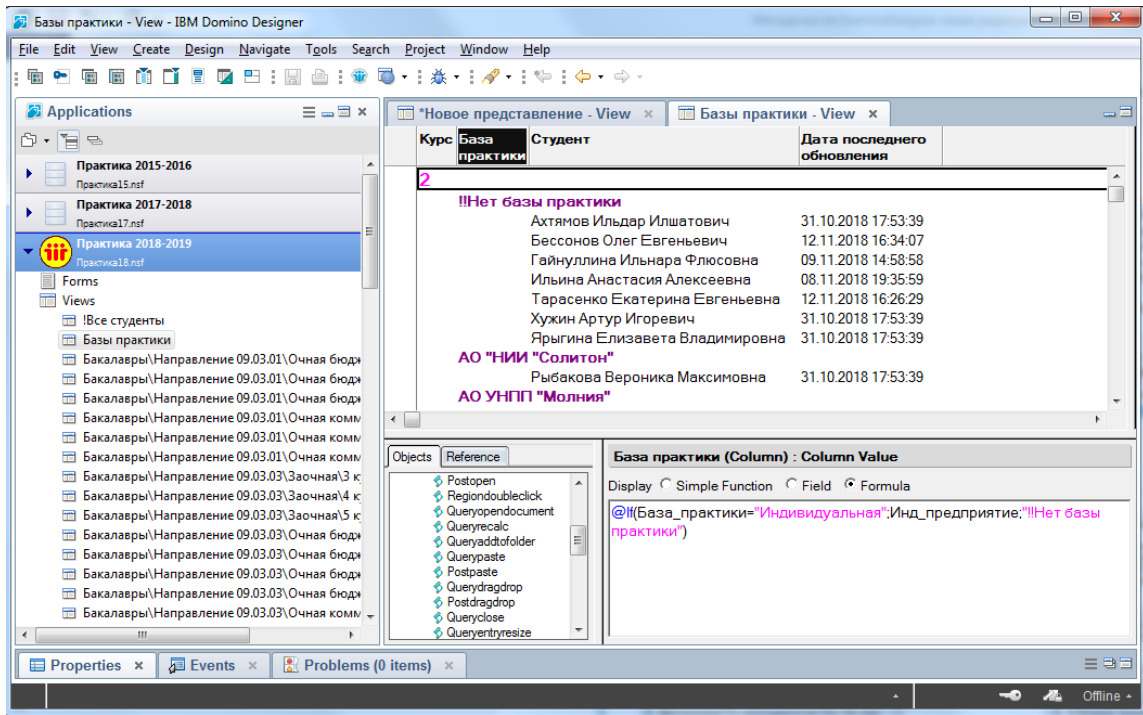


Рис. 30. Определение содержимого столбца через формулу

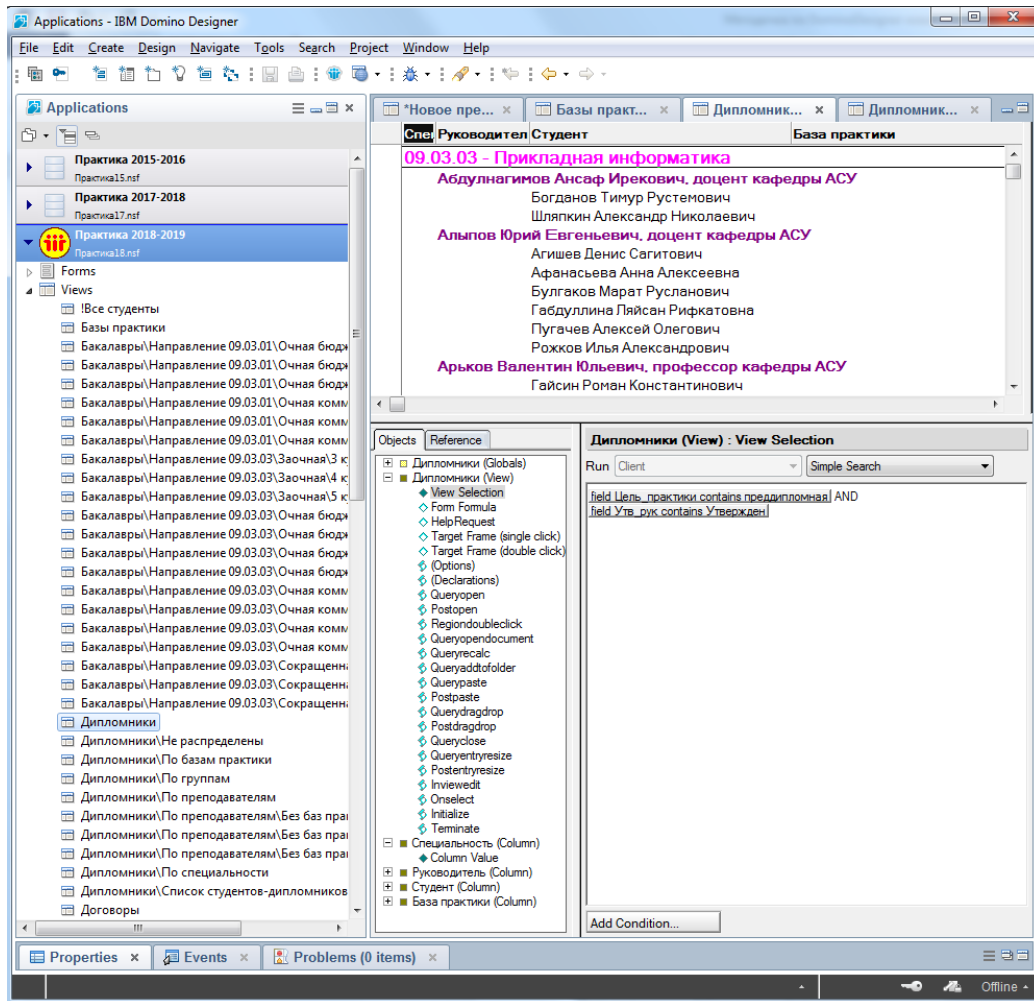


Рис. 31. Отбор документов в представление

Существует два варианта использования этой области для задания критерия отбора документов в представление:

1. *Простой поиск*. В этом случае на экране внизу появляется кнопка *Add Condition* (*Добавить условие*). При нажатии на нее открывается окно диалога, с помощью которого можно задать условия отбора документов (рис. 32).

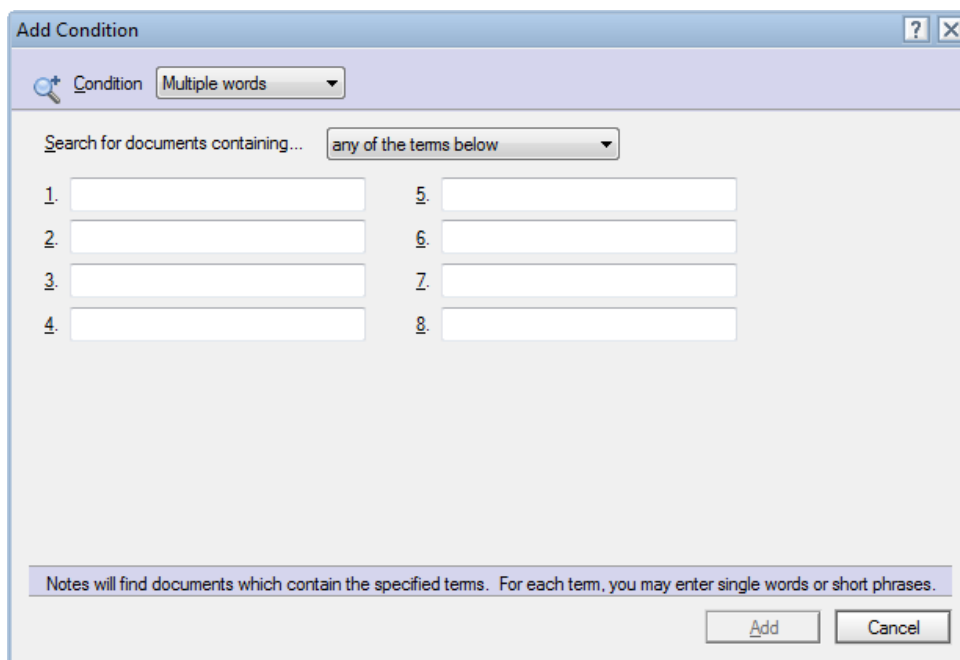


Рис. 32. Окно задания условий отбора документов

Например, в представление можно отобразить только документы, созданные по определенной форме. Для этого из списка *Condition* (*Условие*) нужно выбрать вариант *By form* (*По форме*) и нажать кнопку *Add* (*Добавить*) (рис. 33).

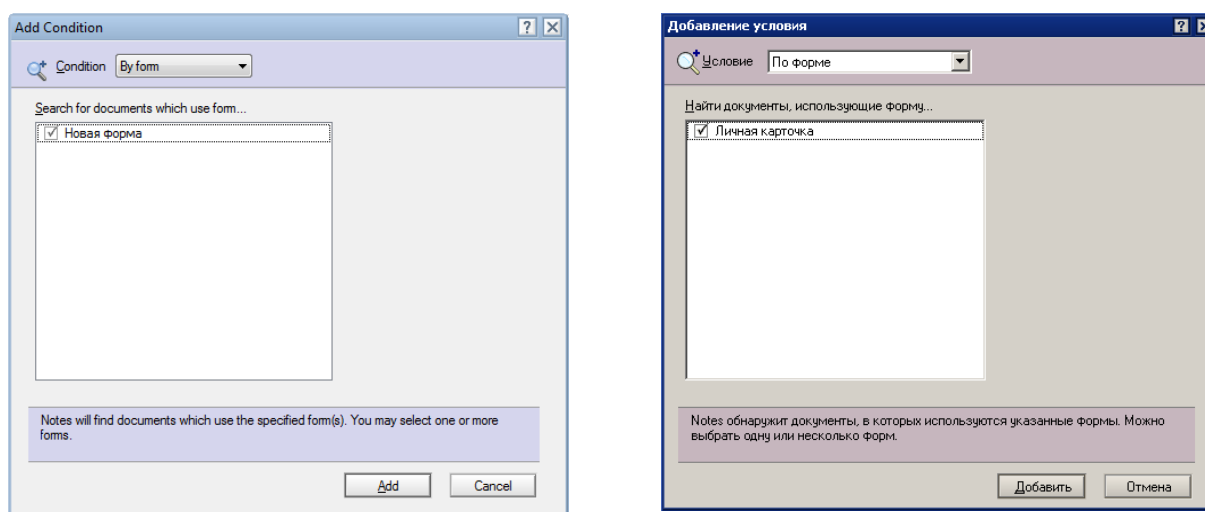


Рис. 33. Окно задания условий отбора документов по выбранной форме

2. *Формула*. В этом случае в поле ввода можно задать формулу отбора документов. По умолчанию в этом поле находится формула *SELECT @All*, в соответствии с которой выбираются все документы. Нужную формулу отбора документов можно выбрать из списка функций и полей в окне «Объекты и списки» на вкладке *View Selection* – можно выбрать простой поиск *Simple Search*, а можно задать формулу (*Formula*), используя язык формул *Lotus Script* (рис. 34).

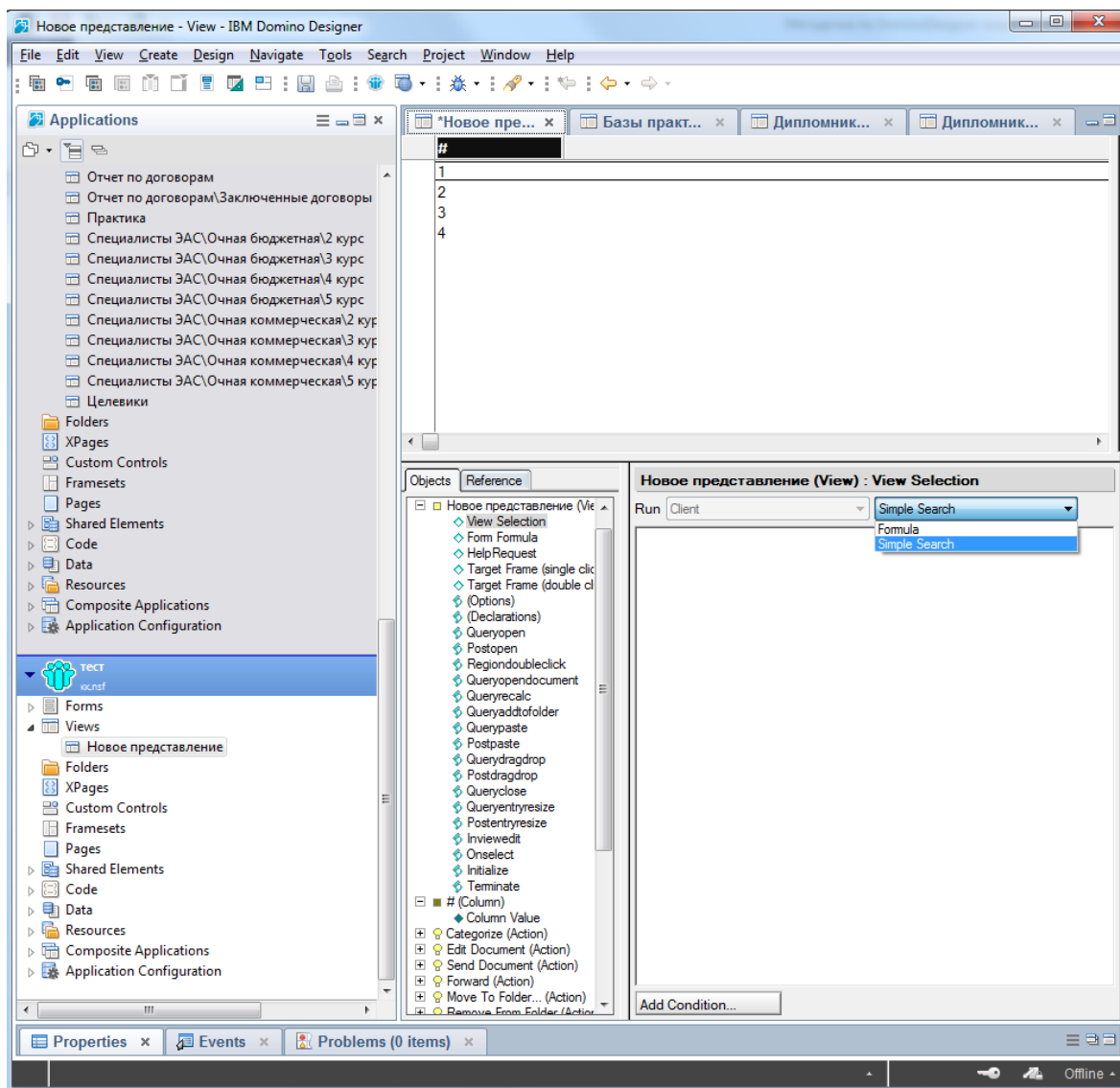


Рис. 34. Выбор типа формул для отбора документов в представление

Поиск *Simple Search* позволяет создавать простые формулы, используя поля базы данных и логические операции (рис. 35). При этом набор логических операций зависит от типа данных поля, а количество условий может быть любым.

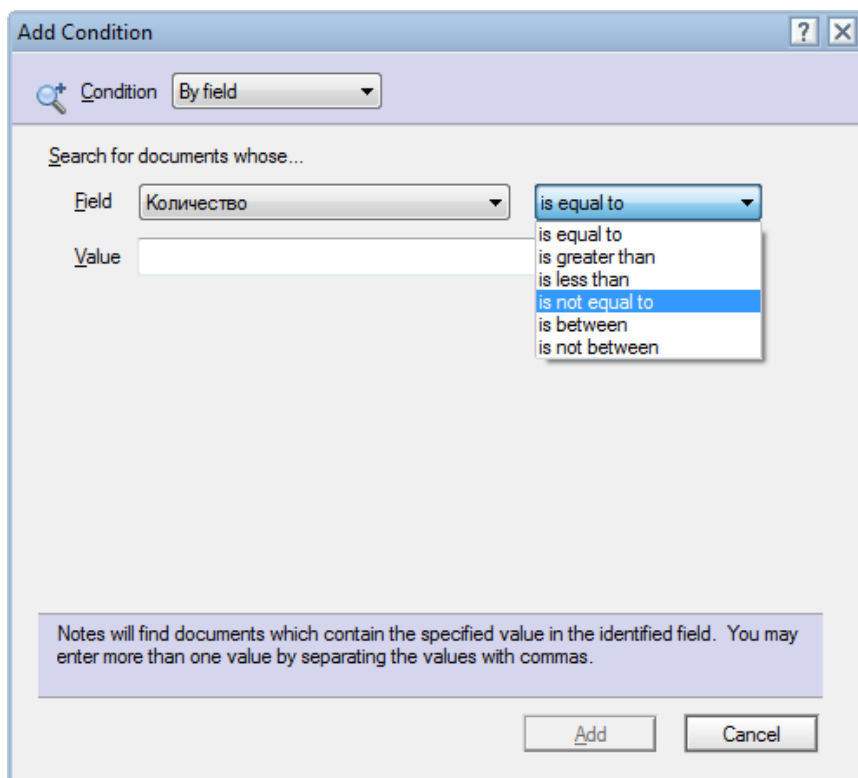


Рис. 35. Выбор условий для отбора документов в представление

2.3.5. Создание нового представления

Для создания нового представления перейдите в область навигатора базы данных и щелкните мышкой на строчке *Views* (*Представления*). Появится окно со списком представлений. Щелкните по кнопке **New View** *Создать представление*. Появится окно для задания свойств представления и критерия отбора документов (рис. 36). Если в базе данных уже имеется сформированное главное представление, то *Notes* предложит вам создать новое представление по его образцу.

Задайте или измените параметры представления.

Имя – должно отражать суть отображаемой информации.

Список *Тип представления* определяет возможности просмотра создаваемого представления пользователями *Notes*.

Личное – представление доступно только своему создателю. Автор представления такого типа имеет на него все права.

Общее – представление доступно для просмотра всем пользователям, но модифицировать его могут только пользователи с правами управляющего или разработчика, а также пользователь с правами редактора, для которого в таблице прав доступа установлен флажок «Создание общих папок и представлений».

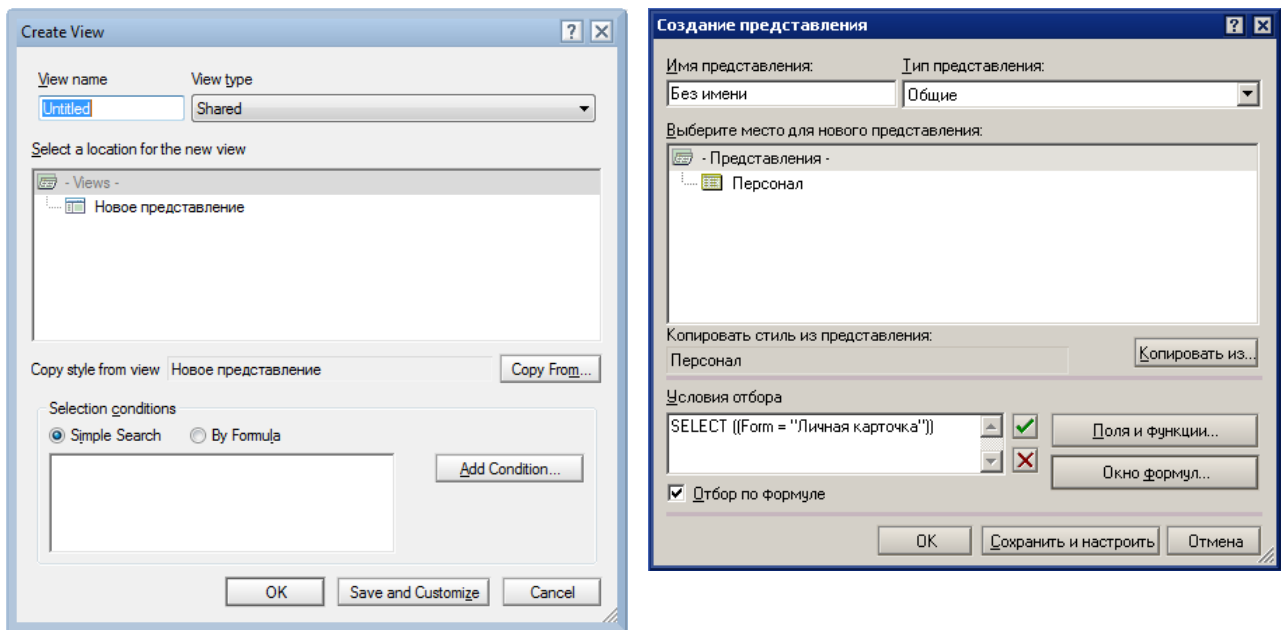


Рис. 36. Окно создания нового представления

Установление флажка «*Общее, личное при открытии*» означает, что пользователь, открыв представление, автоматически создает свою копию представления и в дальнейшем оно никак не связано с оригиналом.

Выберите место для нового представления – позволяет создавать иерархию представлений. Выделите то представление, которое будет являться главным для создаваемого представления. Для размещения представления на верхнем уровне иерархии выберите самую верхнюю строчку. Таким образом, можно создавать иерархию представлений для лучшего упорядочения информации в базе данных. Например, на верхнем уровне иерархии находится представление **Приказы**, где отображается некоторая общая информация об этом классе документов. На следующем уровне иерархии приказы могут быть сгруппированы в два представления: **Приказы\По дате** и **Приказы\По подразделениям**. В Domino Designer при размещении представления на нижнем уровне иерархии в имя представления автоматически копируется имя вышестоящего представления и добавляется значок «\». В IBM Notes пользователь видит значок развертки (зеленый треугольник) возле вышестоящего представления в области навигатора базы данных (см. рис. 16).

Далее вам предлагается *скопировать стиль* из представлений, уже имеющихся в базе данных. Это облегчает работу по созданию нового представления.

Задайте условия отбора документов в представление (аналогично предыдущему пункту).

Кнопка *Add Condition (Добавить условие)* при нажатом флажке *Simple Search* вызывает появление окна «Условия поиска» (рис. 31), с помощью которого также можно задать критерий отбора документов. Эта кнопка становится доступной при неустановленном флажке *By formula (Отбор по формуле)*.

Установка флажка *By formula (Отбор по формуле)* добавляет в окно «Создание представления» две кнопки (рис. 37): *Fields&Functions... (Поля и функции)* (рис. 38) и *Formula Window (Окно формул)* (рис. 39). Эти кнопки открывают окна, с помощью которых можно задать критерий отбора документов посредством формул *Notes* и с использованием имеющихся полей.

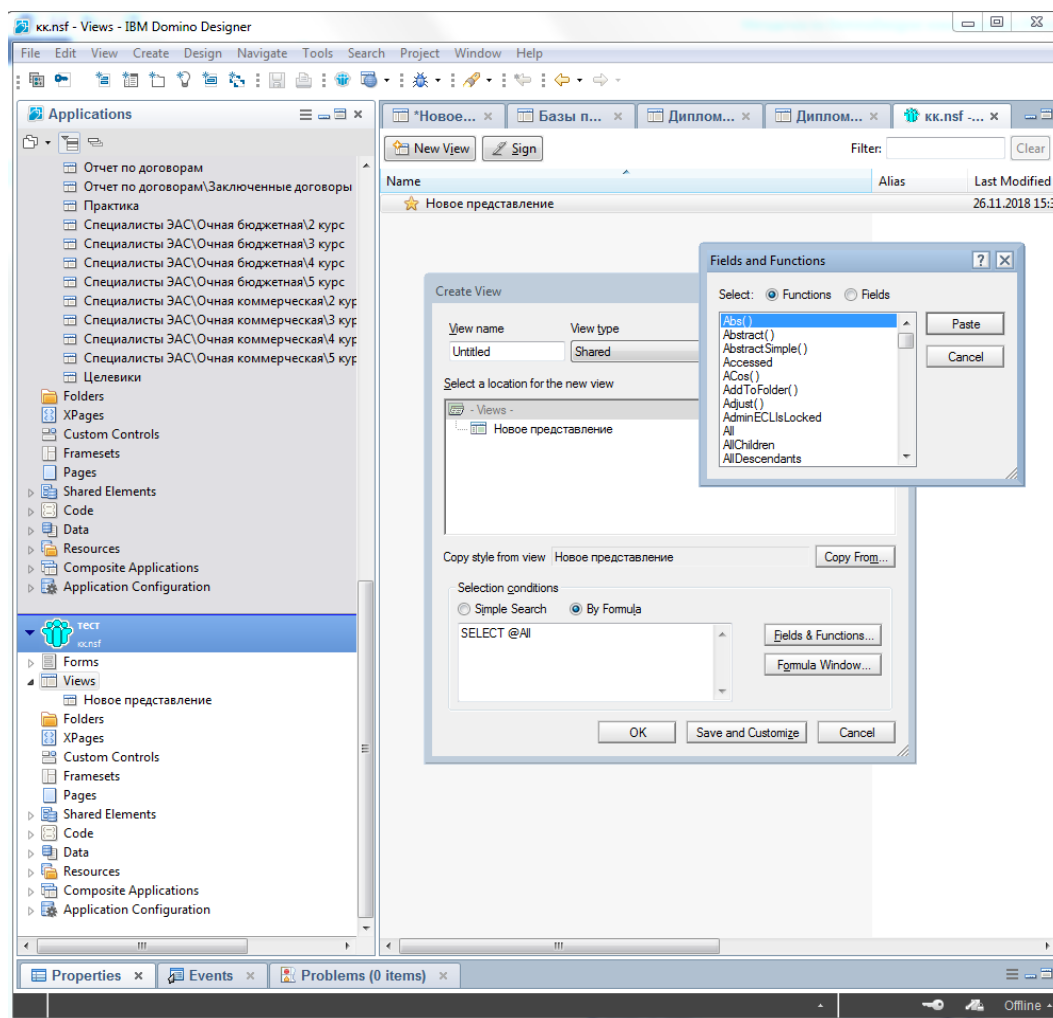


Рис. 37. Варианты отбора документов в новое представление

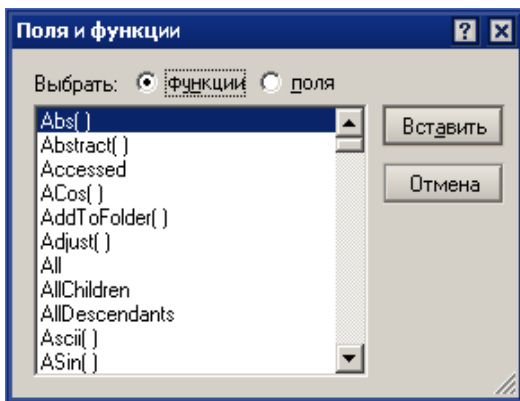


Рис. 38. Окно полей и функций

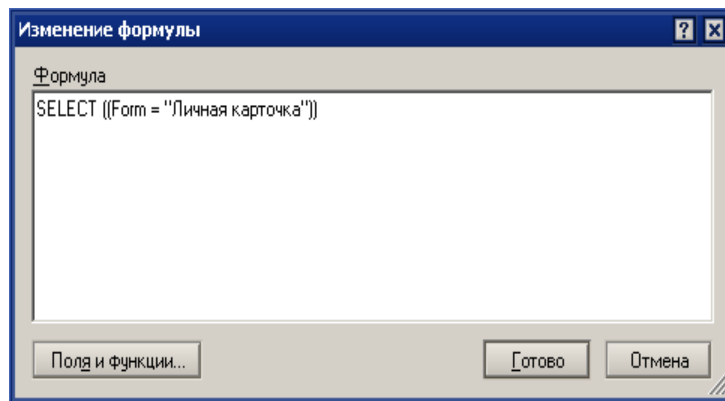


Рис. 39. Окно для создания формул

Кнопка *Сохранить и настроить* позволяет перейти к проектированию нового представления (пп. 3.2–3.3).

2.3.6. Сортировка и использование категорий

При разработке представления целесообразно определить хотя бы один столбец, по значению которого будут упорядочены списки документов. В противном случае документы в представлении будут располагаться в том порядке, в котором они добавлялись в базу.

Если на вкладке **«Сортировка»** окна **«Столбец»** (см. рис. 26) выбран тип сортировки *По группам*, то Notes позволяет сгруппировать документы, имеющие в столбце одинаковые значения, в группы, у которых появляется общий заголовок – **категория**.

В представлении категории отводится отдельная строка, под которой будет помещен список документов, относящихся к данной категории. Для повышения читабельности столбец, по которому группируются документы, надо расположить первым.

Например, в базе данных «Практика» можно сгруппировать списки студентов по месту прохождения практики (рис. 40). Это означает, что все документы со значением столбца, в котором отображается база практики, имеют одинаковое значение. В то же время можно эти же самые документы сгруппировать по признаку руководителя практики, затем по форме обучения студентов (коммерческая или бюджетная), а затем по специальности.

Таким образом, может быть получена иерархия категорий до 32 уровней вложенности.

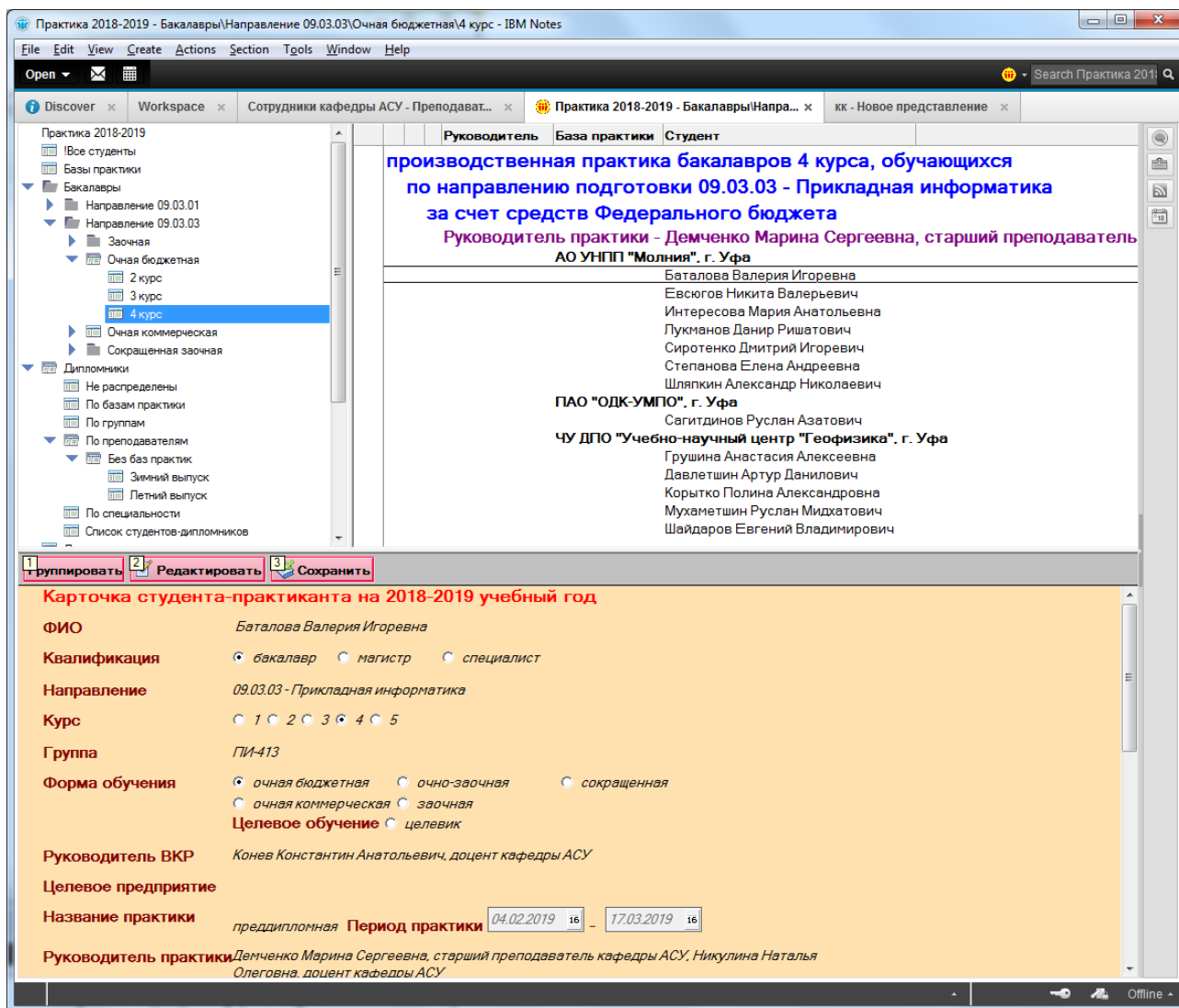


Рис. 40. Вид представления базы данных при использовании категорий

Для создания категории в представлении необходимо выполнить следующие действия:

1) выберите в представлении место для столбца, по значению которого будут группироваться документы;

2) на вкладке «Сортировка» (рис. 26) для столбца выберите значение переключателя *Сортировка – По возрастанию* или *По убыванию*;

3) выберите значение переключателя *Тип – По группам*;

4) на вкладке «Сведения о столбце» установите флажок «Выводить знак развертки, если строка расширяемая»;

5) выберите цветовое и шрифтовое оформление данных для улучшения отображения документов в представлении.

Для создания иерархии категорий нужно включить в представление несколько столбцов, по значениям которых будут группироваться документы. При этом следует учитывать порядок размещения этих столбцов в представлении, т.е., столбец, который содержит более общую категорию для всех документов, должен стоять первым.

2.3.7. Проектирование папок и работа с документами в папках

Папки аналогично представлениям содержат некоторые списки документов.

Основное отличие папок от представлений заключается в отсутствии критерия отбора документов в папку. В начальном состоянии папка пуста, но в любой момент времени любой документ может быть добавлен в папку, удален из папки или перемещен из одной папки в другую. Перемещение или добавление документов в папку можно осуществить с помощью мыши или команд меню *Действия*. При этом набор команд меню *Действия* отличается в зависимости от того, находитесь ли вы в представлении или в папке.

Проектирование папок отличается от проектирования представлений только отсутствием необходимости определения критерия отбора документов.

Для создания новой папки в области навигатора базы данных щелкните мышкой на строчке *Папки*. Откроется окно с перечнем папок. Дальнейшие действия и возникающие окна для проектирования папок ничем не отличаются от действия и элементов аналогичных окон для проектирования представлений (рис. 41).

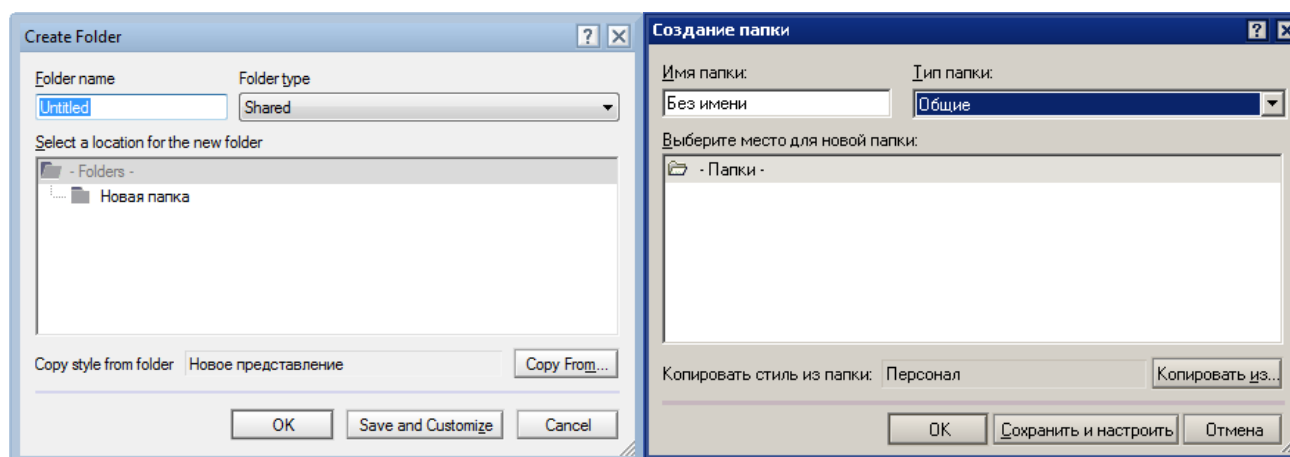


Рис. 41. Окно создания папки

Перемещение документов в папку может осуществляться только пользователем в IBM Notes.

Для перемещения документов нужно, находясь в области представления, выбрать мышью один или несколько документов (при наличии категорий в представлении курсор должен указывать на нижний уровень иерархии). Затем выбрать в меню команду *Actions/Move To Folder* (*Действия/Переместить в папку*) (рис. 42).

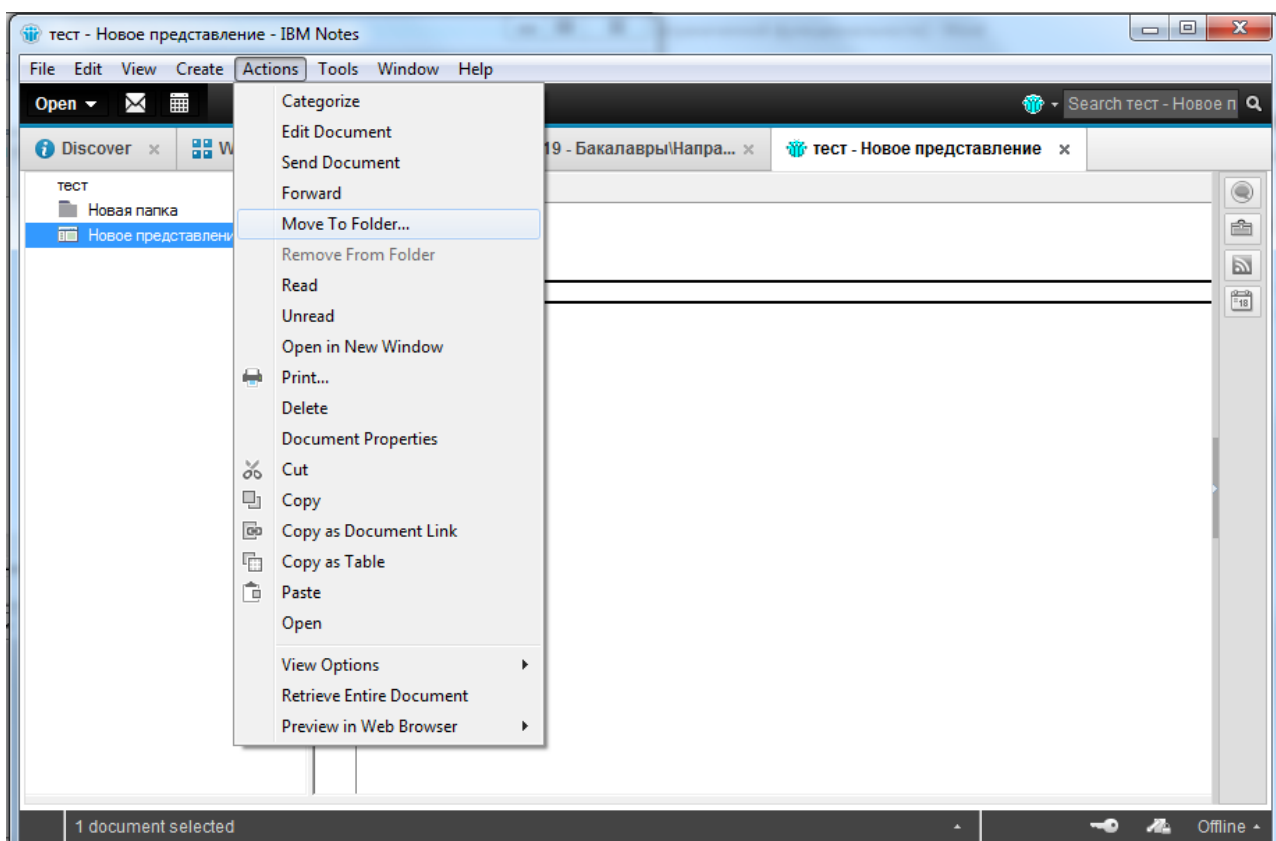


Рис. 42. Перемещение документов в папку

Документы можно также перемещать и копировать из папки в папку.

Удаление документов из папок осуществляется в окне соответствующей папки по команде из меню *Действия/Удалить из папки*. При этом документ удаляется из папки, а в представлении он только помечается на удаление.

2.4. Требования к отчету

1. В результате выполнения лабораторной работы база данных должна содержать не менее 10 документов, 3 представления, 2 папки.

2. Представления должны максимально полно отражать информацию из документов, собранных в базе данных.

3. Представления должны содержать не менее 4–5 столбцов, среди которых 2 столбца являются категорией, один является простой функцией *Notes*, 2 отображают значения полей базы данных. Необходимо использовать разные стили для оформления заголовков столбцов, данных и категорий.

Контрольные вопросы

1. Что такое представление базы данных?
2. В чем отличие представления от папки?
3. Что такое категория?
4. Сколько уровней вложенности может иметь иерархия категорий?
5. Приводит ли к физическому удалению документа удаление его из папки?
6. Приведите примеры простых функций *Notes*.

ЛАБОРАТОРНАЯ РАБОТА № 3

ДЕЙСТВИЯ, ФУНКЦИИ И КОМАНДЫ NOTES

3.1. Цель работы

Целью работы является изучение некоторых наиболее распространенных функций и команд Notes, применяемых при разработке базы данных, а также возможностей автоматизации работы пользователей в процессе ведения базы данных.

3.2. Теоретические сведения

Функции представляют собой оператор, который определенным образом обрабатывает данные и присваивает значения переменной.

Функции можно объединить в следующие основные группы.

Текстовые – позволяют управлять текстом и преобразовывать данные, представленные в текстовом формате, в другие форматы и наоборот. Позволяют контролировать длину поля, выделять любое количество символов из поля.

Математические – работа с числовыми данными.

Время и дата – действия над значениями времени и даты. Очень полезны для вычислений значений по умолчанию.

Логические – позволяют осуществить выбор. @If – позволяет строить достаточно сложные логические конструкции, т. к. обладает свойством вложенности.

Функции обработки документов – позволяют проводить статистическую обработку документа (например, сколько соединений имеет документ).

3.2.1. Как получить информацию о функции или команде

При разработке в Domino Designer форм, представлений, агентов в левом нижнем углу экрана появляется область объектов и списков. На вкладке *Reference (Списки)* находится выпадающее меню с тремя опциями (рис. 43):

- поля базы данных;
- @-команды в формуле;
- @-функции в формуле.

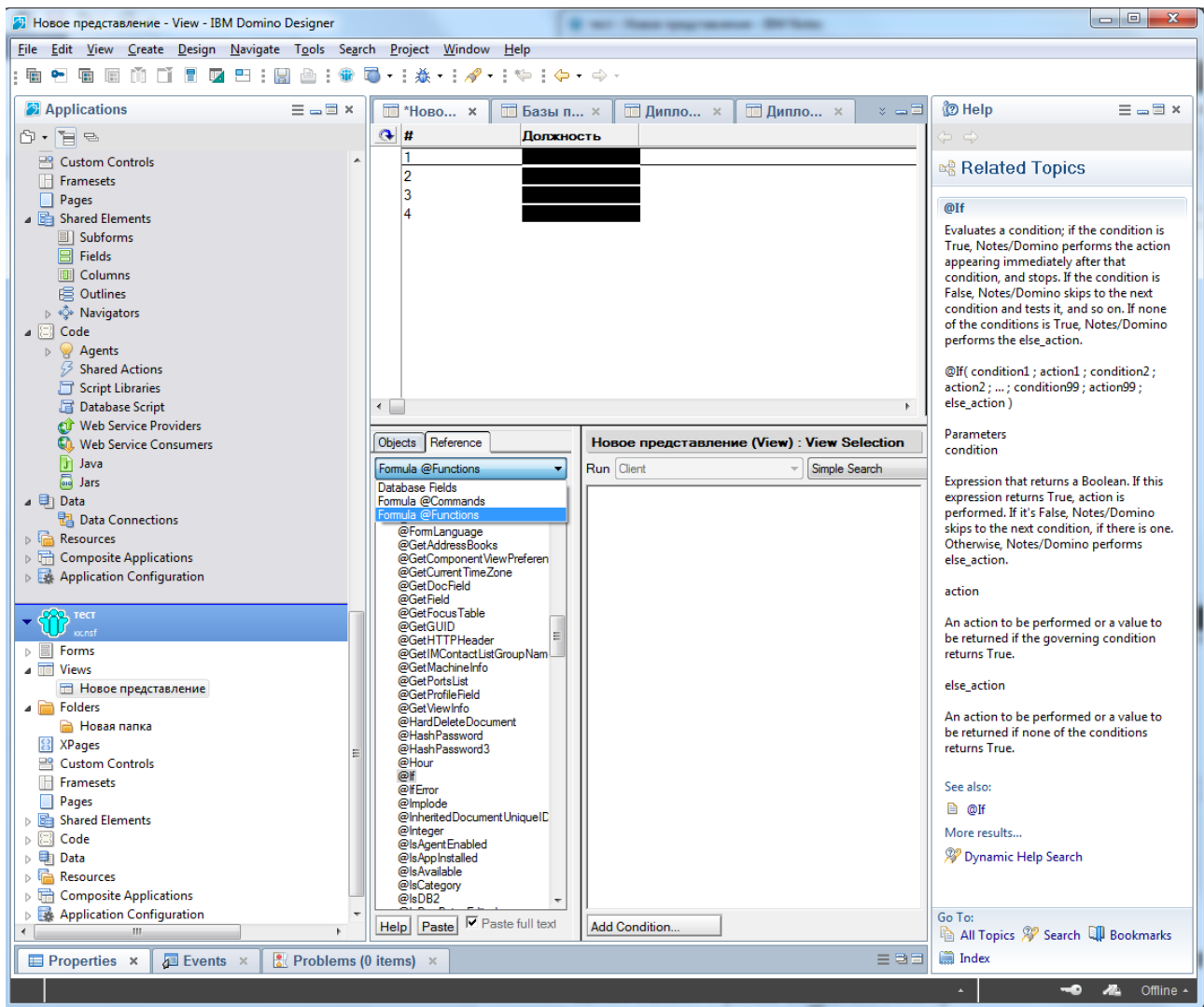


Рис. 43. Раздел Reference для записи формул

Открыв нужный список, найдите функцию или команду, укажите на нее мышью и нажмите F1 или клавишу *Справка*. Вы получите подробные инструкции на английском языке о синтаксисе команды или функции с примерами ее применения.

3.3. Порядок выполнения работы

В ходе выполнения лабораторной работы необходимо:

1. Создать панель действий для формы (использовать как минимум одно системное действие и одно собственное, например, кнопку с командой *Сохранить документ* (@FileSave)).
2. Создать панель действий для представления (использовать как минимум одно системное действие и одно собственное,

например, кнопку с командой *Редактировать документ* (*@EditDocument*)).

3. Создать агента для базы данных, который изменял бы значение какого-либо поля в документах или перемещал бы документы из представления в папку.

4. Использовать функцию *@If* для задания заголовка окна документа и проверки какого-либо условия.

5. Связать два документа в одной базе данных, используя функции и команды *Notes* и гиперобъект-кнопку.

6. Вложить документ *MS Word* в документ *Notes*.

7. Связать два документа из разных баз данных, связать документ с представлением из другой базы данных, связать документ с другой базой данных.

3.3.1. Панель действий для формы

При проектировании форм можно создавать действия – специальные кнопки, которые расположены на панели действий, расположенной в верхней части окна с документом (см. рис. 15).

Панель действий присутствует на экране постоянно, она не прокручивается вместе с текстом документа. Набор действий в панели является специфичным для каждой формы и определяется разработчиком. Следует помнить, что панель действий для формы определяет только те действия, которые можно производить с текущим открытым документом.

Если открыть окно проектирования формы и выполнить команду *View/Action Pane* (*Вид/Область действия*), то справа появится область со списком действий (рис. 44).

Если область пуста, в ней нужно щелкнуть правой кнопкой мыши и из контекстного меню выбрать команду *Insert System Actions* (*Добавить системные действия*) (рис. 45). В *Notes* есть несколько стандартных predefined действий (они помечены звездочкой), которые нельзя удалить.

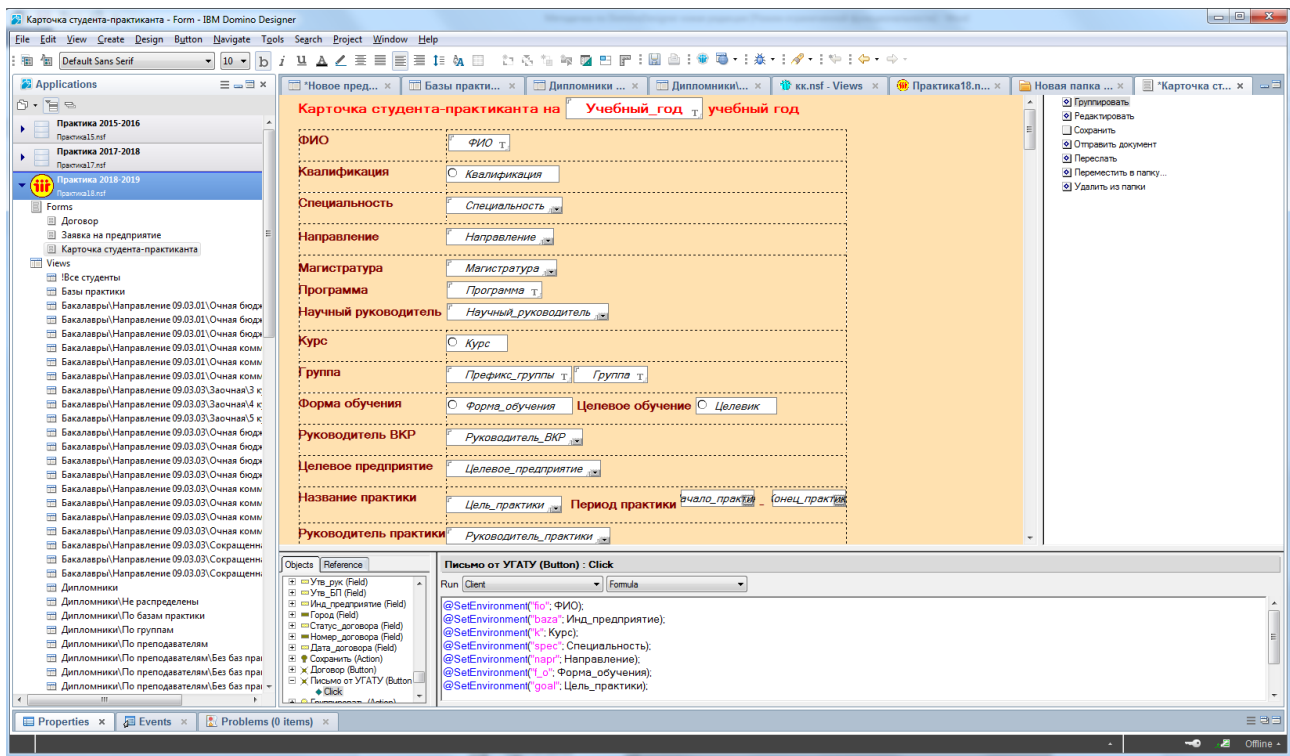


Рис. 44. Форма документа с областью действий

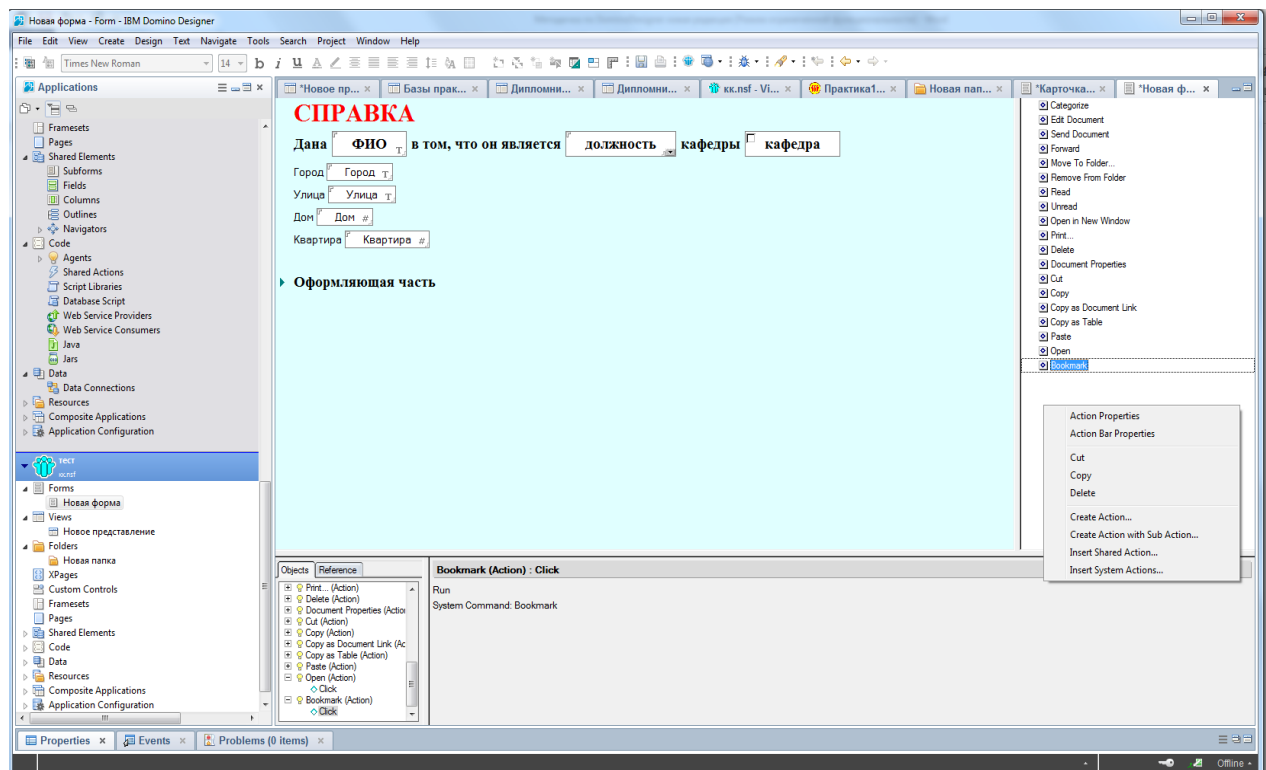


Рис. 45. Добавление системных действий на панель действий

Назначение некоторых predetermined действий Notes сведено в табл.. 2.

Назначение предопределенных действий

Действие	Назначение
Изменить документ	Переключение между режимами чтения/редактирования документа
Группировать	Классификация выбранных в представлении документов
Переслать	Отправка почтой изображения документа
Переместить в папку	Перемещение выбранных документов в папку
Удалить из папки	Удаление документов из открытой в данный момент папки
Отправить документ	Отправка выбранных документов почтой

Для вынесения на панель действий нужного действия из этого списка выделите его и выберите команду *Design/Action Properties* (*Конструктор/Свойства действия*) (рис. 46).

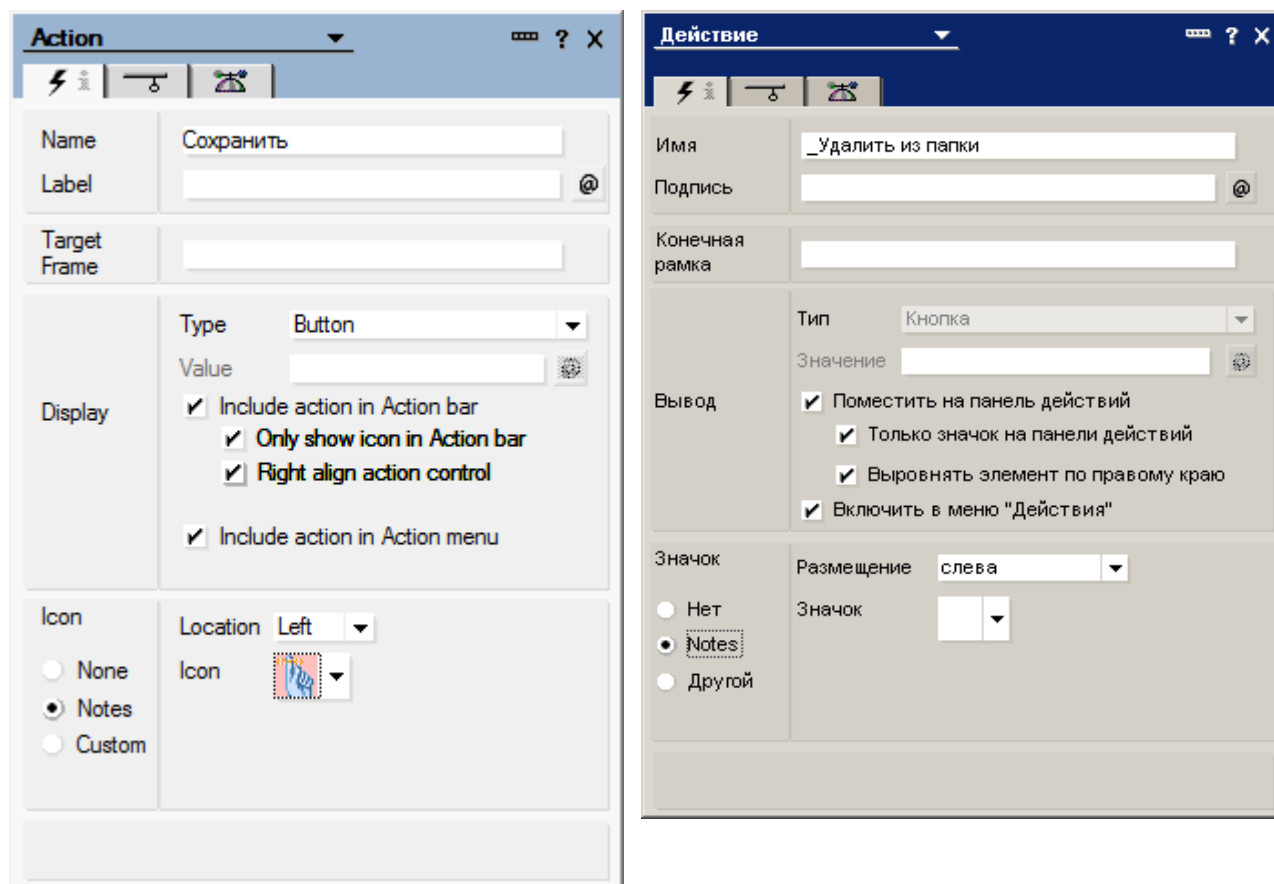


Рис. 46. Окно свойств действия

Для отображения действия в документе на вкладке *Сведения о действии* щелкните на флажок «*Поместить на панель действий*».

ВНИМАНИЕ! Кнопка с нужным действием отображается только в документе, с которым работает пользователь в IBM Notes, для разработчика в форме Domino Designer она не видна. Для проверки правильности действий нужно выполнить команду *Design/Preview in Notes (Конструктор/Просмотреть в Notes)* или создать новый документ.

В область действий можно добавить свои действия, разработанные только для этой базы данных. Для этого нужно проделать следующие действия:

- 1) откройте форму, в которой надо создать действие;
- 2) выполните команду *Create/Action/Action... (Создание/Действие/Действие...)*. На панели действий появится новое действие (рис. 47).

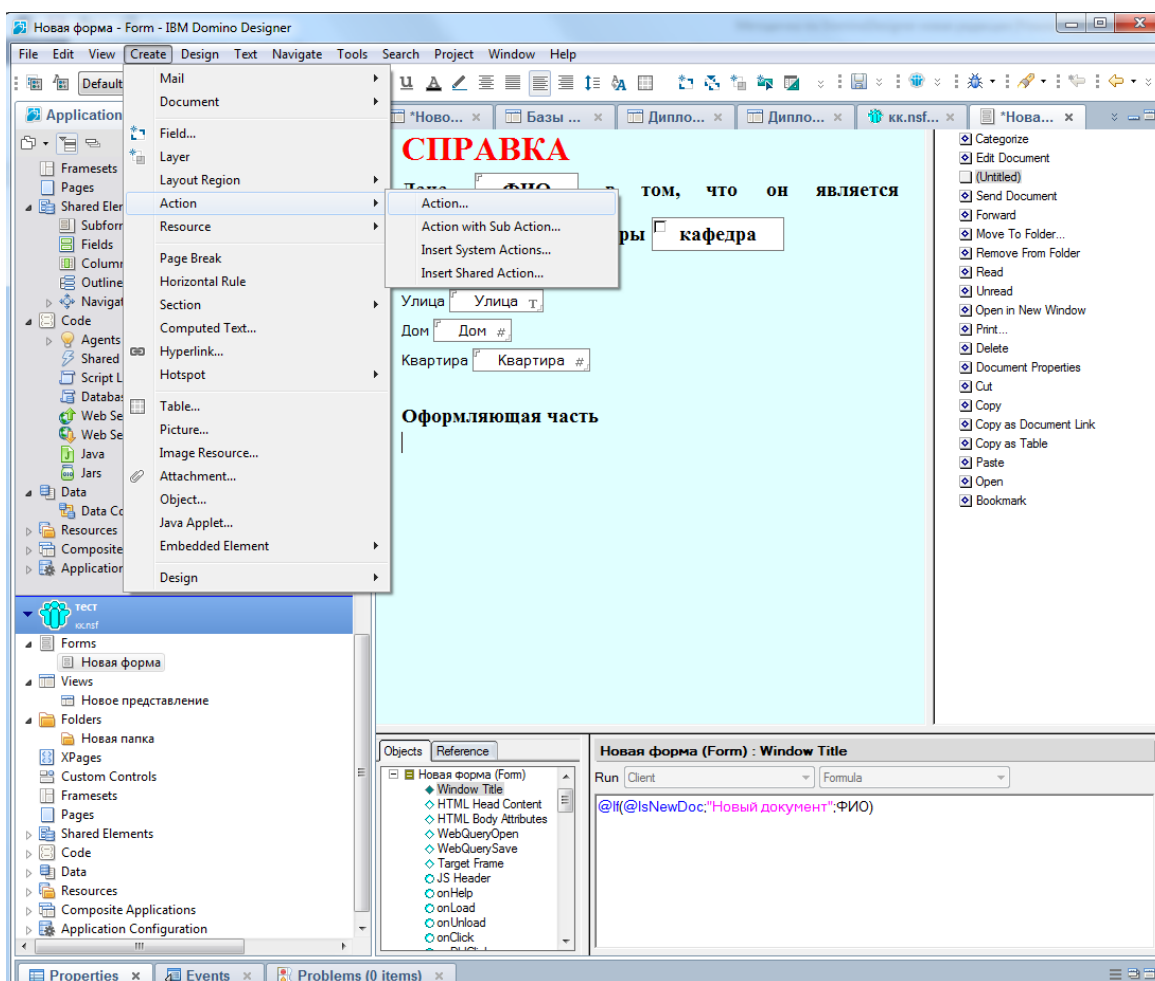


Рис. 47. Добавление кнопки на панель действий формы

3) определите, что нужно сделать при нажатии кнопки. Для этого сначала дайте название действию, а затем в области конструктора (*Reference*) выберите простое действие либо напишите формулу с использованием @-функций и/или @-команд;

4) в окне «Свойства действия» на вкладке *Сведения о действии* задайте основные атрибуты действия – название, значок для кнопки, включите флажки «*Включить в меню Действия (Include action in Action bar)*» и «*Поместить на панель действий (Include action in Action menu)*», укажите позицию в списке (рис. 45).

Вид панели действий можно настроить, выбрав команду *Design/Action Bar Properties (Конструктор/Свойства панели действий)* (рис. 48).

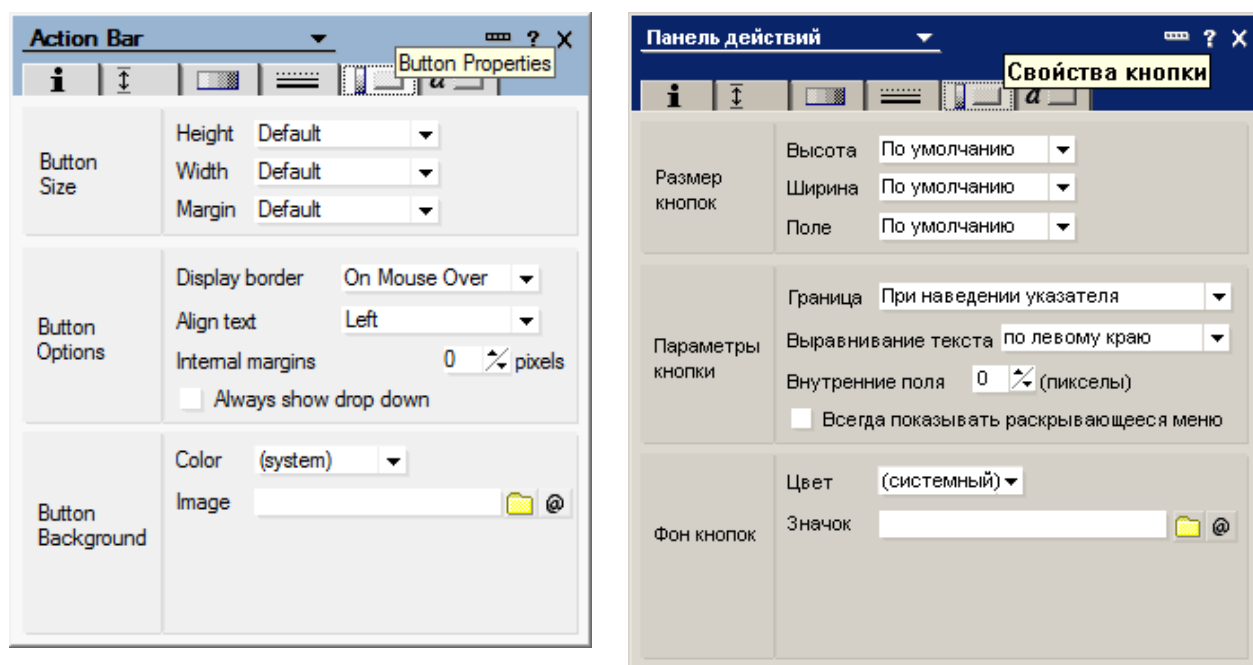


Рис. 48. Окно свойств панели действий

3.3.2. Панель действий для представления

При проектировании представлений можно создавать действия, расположенные в панели действий представления (см. рис. 16).

Панель действий представления – это непрокручиваемая область в верхней части окна представления. Для каждого представления разрабатывается свой набор кнопок действий. Следует помнить, что панель действий для представления содержит только те действия, которые можно выполнить над документами, не открытыми в данный момент.

Панель действий представления создается аналогично панели действий формы.

3.3.3. *Агенты*

Каждая база данных имеет в области навигатора папку «Агенты». В этой папке перечислены агенты, к которым пользователь имеет доступ в соответствии со своим уровнем. *Агент* – это программа, написанная разработчиком на языке *Lotus Script* или с применением простых функций и формул, автоматизирующая выполнение каких-либо действий в базе данных. Агенты могут служить для модификации содержания полей сразу нескольких документов, отправки почтовых сообщений, поиска специфической информации, архивирования устаревших документов и т. д. Простейшие агенты могут быть созданы любым пользователем Notes.

Агенты могут запускаться:

- из меню;
- из других агентов;
- при нажатии кнопок;
- при выполнении действий;
- в зависимости от наступления определенного заранее события (например, при поступлении новой почты, создании документов);
- по расписанию.

Агенты могут быть личными и общими. Агенты используют те же виды действий, которые применимы к представлениям и формам. Однако, если в случае представлений и форм действия выполняются только в контексте открытого представления или формы, действия агентов распространяются на все документы во всей базе данных.

Создавать агентов можно либо путем копирования и модификации ранее созданных агентов, либо путем использования конструктора агентов.

При создании нового агента нужно в области навигатора базы данных развернуть папку *Code* (*Общий код*) и щелкнуть мышкой на строчке *Agents* (*Агенты*) или выбрать из меню команду *Create/Design/Agent* (*Создание/Конструктор/Агент*). Задайте имя агенту, стараясь, чтобы оно отражало назначение агента или выполняемые действия, выберите способ его создания из выпадающего меню (рис. 49) и нажмите кнопку «ОК».

ВНИМАНИЕ! Выбранный тип агента впоследствии изменить нельзя.

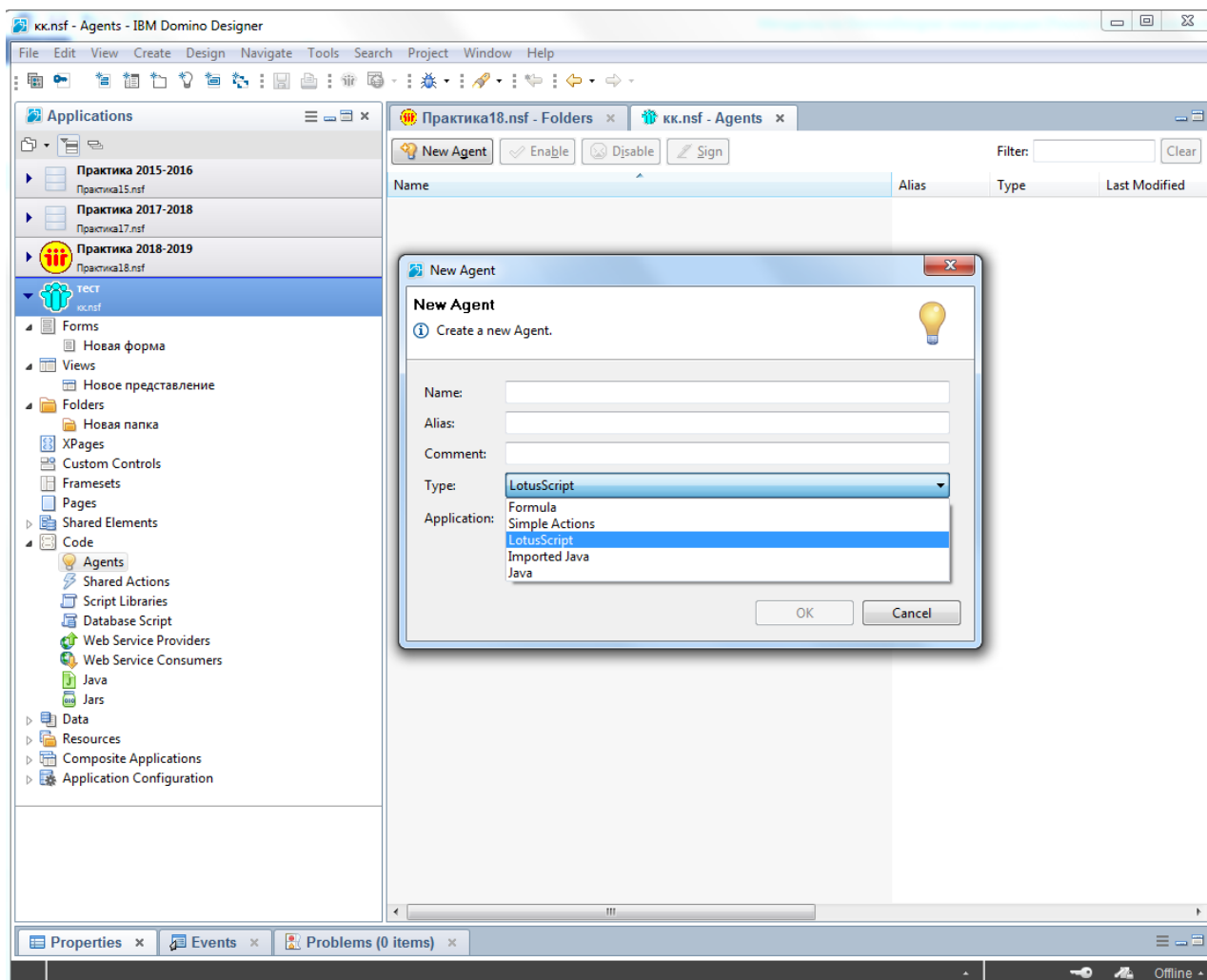


Рис. 49. Создание нового агента

Появится область проектирования агентов (рис. 50), состоящая из двух частей – окна объектов и окна действий, содержащего команды, выполняемые агентом.

Далее нужно проделать следующие действия:

1) в окне *Objects* выберите сначала *Document Selection* – агент должен знать, с какими документами из базы данных он будет работать. Отбор документов можно выполнить, нажав кнопку «Add Condition»;

2) в окне *Objects* выберите *Action* – агент должен знать, какие действия необходимо выполнить со всеми отображенными документами (кнопка «Add Action»);

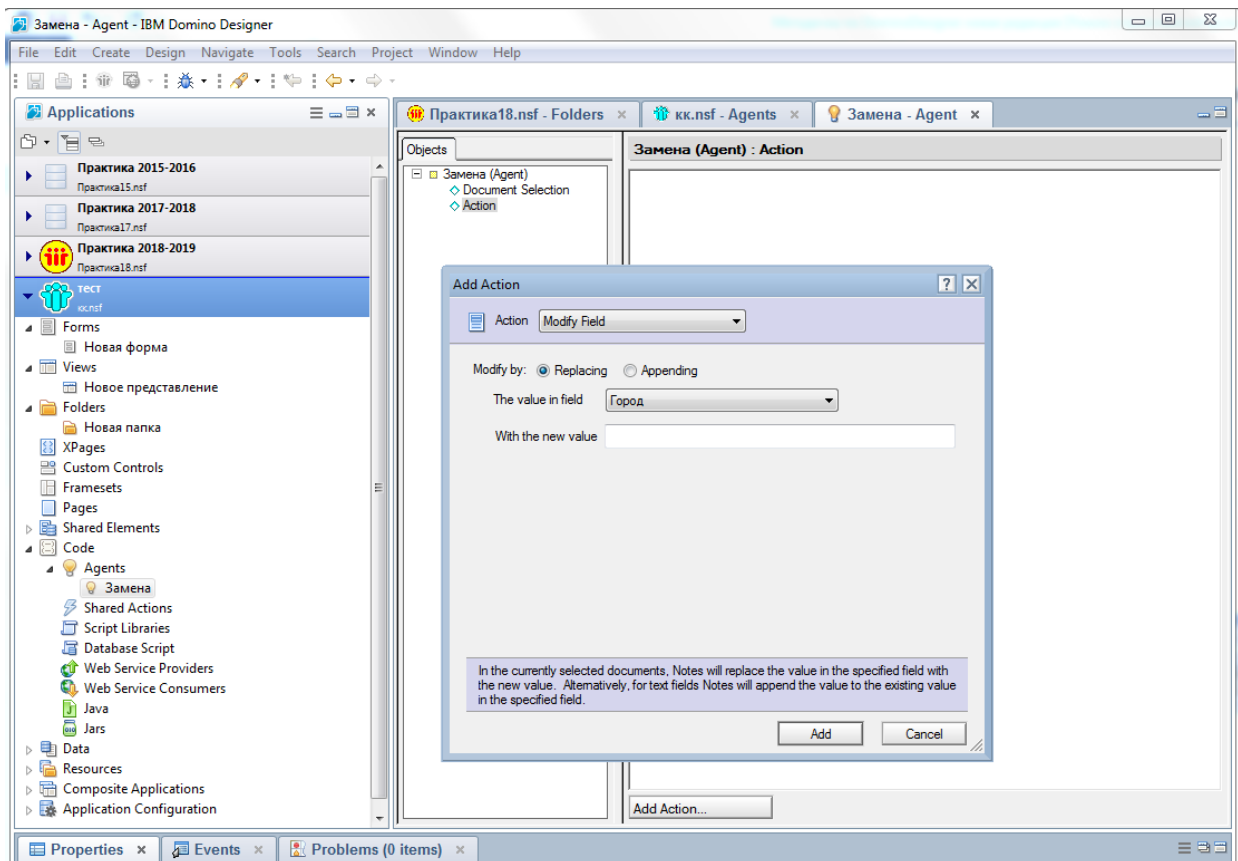


Рис. 50. Область проектирования агентов

3) выпадающий список *Action* содержит список predefined действий (рис. 51);

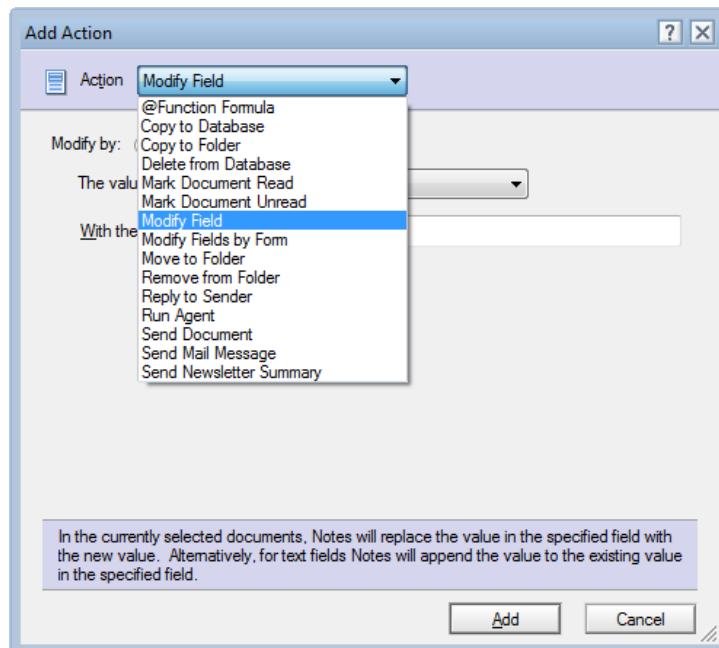


Рис. 51. Список predefined действий агента

4) сохраните и закройте агент.

В результате в области навигации базы данных в папке «Агенты» появится запись об агенте.

Запуск агента осуществляется пользователем IBM Notes следующим образом (рис. 52):

- 1) выберите представление, в котором будете запускать агента;
- 2) отметьте документы, для которых вы хотите выполнить действия;
- 3) запустите нужный агент командой из меню *Actions*.

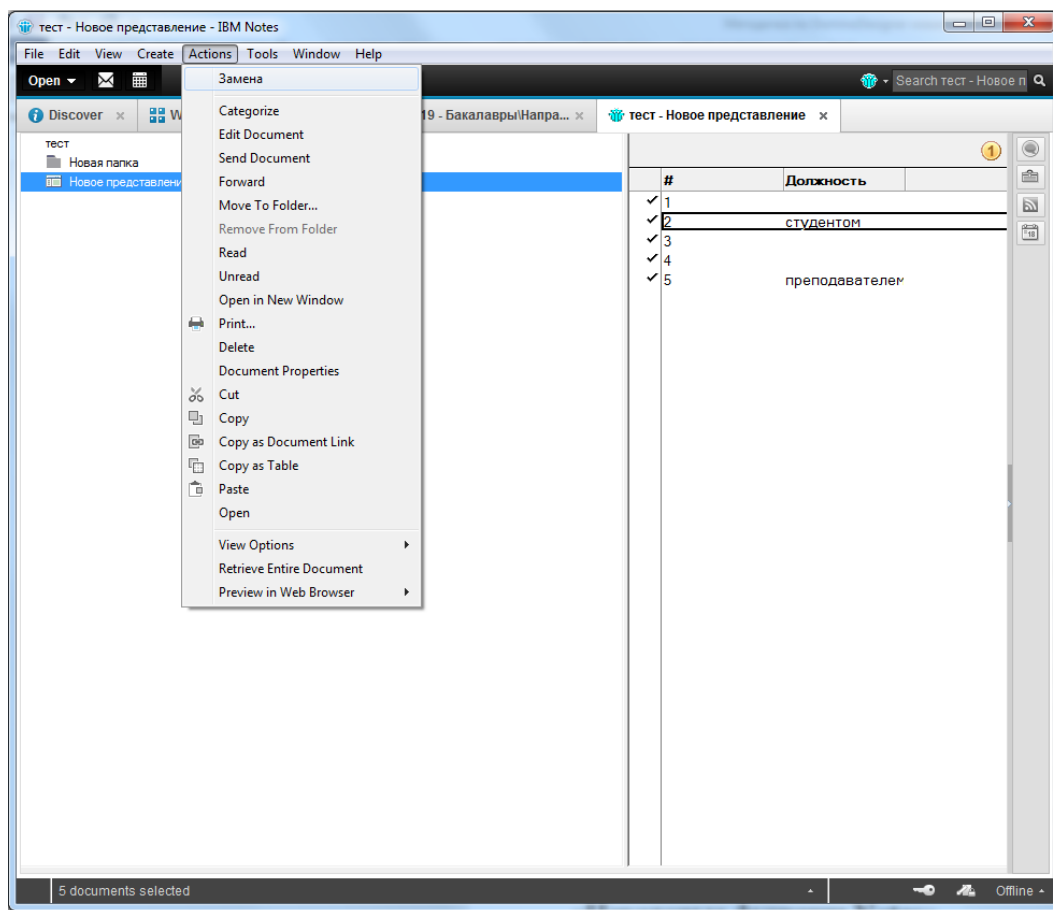


Рис. 52. Запуск агента

3.3.4. Создание кнопок (гиперобъектов) в форме

Кнопки, так же как действия и агенты, позволяют автоматически выполнять набор действий, определенный разработчиком.

Разница между кнопкой и действием заключается в способе использования — действие имеет смысл применять, когда необходимые операции должны выполняться над всем документом и запускаться независимо от того, какой фрагмент документа

находится на экране. Если же надо определить операции над полем или фрагментом формы, целесообразнее использовать кнопки.

При выполнении задания используйте пример №3 из раздела 4 «Некоторые функции Notes».

Для создания кнопки выполните следующие действия:

1. Поместите курсор в форме там, где вы планируете разместить кнопку, и выполните команду *Create/Hotspot/Button* (рис. 53).

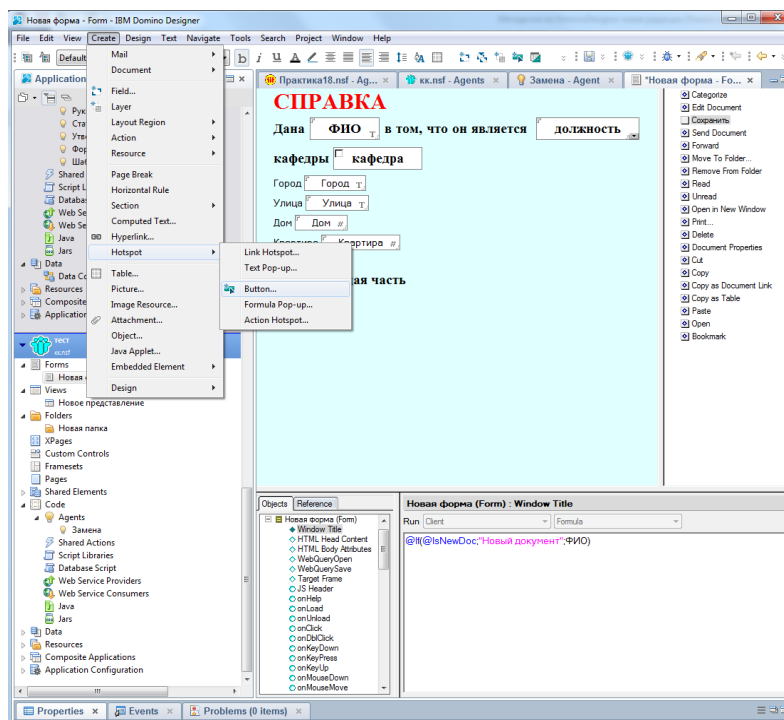


Рис. 53. Вызов команды для вставки кнопки

Откроется окно *Button (Кнопка)* со свойствами этого объекта (рис. 54).

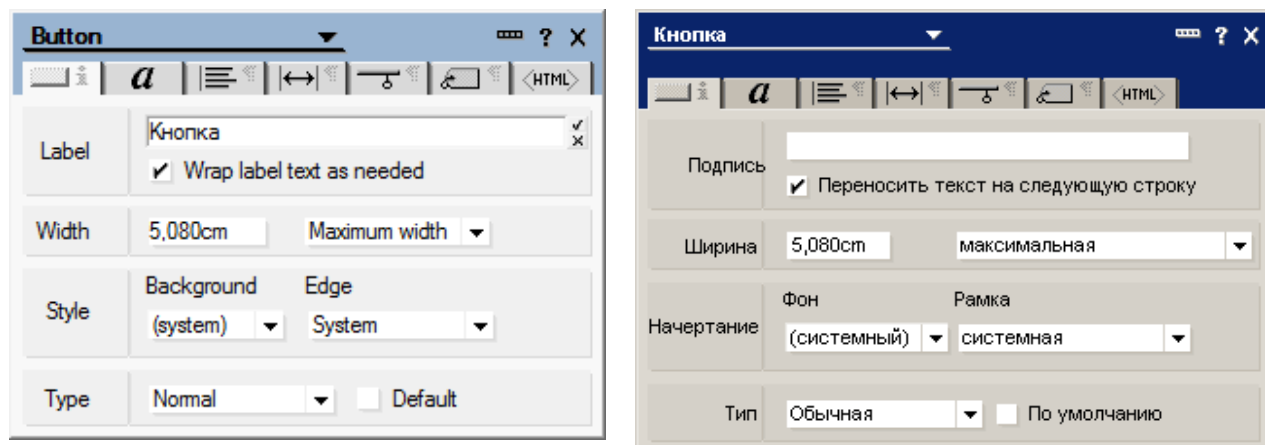


Рис. 54. Окно свойств гиперобъекта-кнопки

2. Задайте параметры кнопки – имя, атрибуты текста и т. д.
3. Определите, какие действия будут выполняться в результате нажатия на кнопку. Для этого в области конструктора для кнопки *Кнопка (Button): Click* выберите один из способов – *Простые действия* или *Формула*.
4. Дальнейшие шаги по созданию кнопки аналогичны шагам, выполняемым при создании агентов.

3.3.5. Обмен информацией с другими приложениями

При создании документа Notes может понадобиться обращение к дополнительной информации, хранящейся либо в базах Notes, либо во внешнем источнике информации.

Если нужная информация содержится в другом документе Notes, можно предоставить возможность пользователю просмотреть ее, читая подготовленный вами документ. Если информация содержится в файле, можно вложить файл в документ таким образом, чтобы читатель документа мог просмотреть содержимое файла или скопировать его.

Способы обмена информацией, поддерживаемые Notes:

- копирование данных в документ Notes;
- вложение файла в документ Notes;
- импорт файла в документ Notes;
- экспорт документа Notes в файл;
- внедрение данных в документ Notes и связывание документа Notes с данными из других приложений.

Выбор того или иного способа определяется как возможностями, так и преследуемыми целями.

Вложение файла

Технология вложения файла используется тогда, когда:

- вы не хотите, чтобы данные становились частью документа Notes;
- файл данных сжат или является системным;
- пользователи могут предпочесть использовать файл с приложением-источником, а не с Notes;
- приложение-источник не поддерживает OLE-технологию.

Чтобы вложить один или несколько файлов, нужно выполнить следующие действия:

1. Установить курсор в любое место формы в режиме проектирования и выбрать *Создание/Вложение*. На экране появится окно диалога «*Создание вложения*».

2. В окне отобразится список файлов. Выбрать нужный файл (если несколько, использовать клавишу *Ctrl*).

3. Notes вставит в документ соответствующие пиктограммы по одному на каждый присоединяемый файл.

Теперь при работе с каждым документом, созданным по этой форме, пользователю IBM Notes будет доступно вложение либо в режиме просмотра, либо в режиме редактирования.

Связывание

Создавая связь в документе Notes, вы тем самым вставляете указатель на информацию во внешнем файле. Поскольку связь указывает на оригинальные данные, любые изменения в связанном файле отражаются в документе Notes.

Связывание имеет смысл в следующих случаях:

– Вы хотите централизовать хранение конкретной информации, которой могли бы пользоваться другие;

– исходный файл не будет перемещаться или удаляться (иначе связь придется переустанавливать);

– все пользователи, которым потребуется редактировать информацию, будут иметь доступ к приложению-серверу и исходному файлу на файл-сервере.

Создание связи с документом, представлением и базой данных. Вы можете создать связь с документом, представлением или базой данных Notes, работая в своей базе данных Notes, путем создания соответствующего объекта.

Для создания объекта связи документов Notes из разных баз данных выполните следующие действия:

1) откройте в IBM Notes ту базу данных, с которой вы хотите связать документ в вашей базе данных, например базу данных «Личный журнал» или «Контакты»;

2) выберите в этой базе данных документ для связи;

3) выполните команду *Правка/Копировать как связь/С документом*;

4) вернитесь в вашу базу данных в Domino Designer;

5) выберите форму, в которую вы хотите вставить указатель на связываемый документ и откройте ее для редактирования;

6) установите курсор в том месте, где должен находиться указатель;

7) выполните команду *Правка/Вставить*. В результате Notes поместит в указанном вами месте объект (значок документа той базы данных Notes, связь с которой вы создаете), дважды нажав на который, вы откроете документ;

8) закройте и сохраните форму.

Теперь используемая форма будет содержать значок связи с выбранным документом из другой базы данных, и каждый вновь создаваемый документ также будет включать эту связь. Для проверки попробуйте создать документ по этой форме. По щелчку на объекте связи с документом на экране откроется связанный документ из другой базы данных. Чтобы вернуться в исходный документ, достаточно нажать *Esc*.

Аналогичные действия проделываются, если необходимо связать документ в вашей базе данных с представлением из другой базы данных или с другой базой данных целиком. Для этого нужно выбрать представление или базу данных в рабочей области, а в п. 3 выбрать команду *Правка/Копировать как связь/С представлением* или команду *Правка/Копировать как связь/С базой данных* соответственно.

3.4. Некоторые функции Notes

1. *@If(условие1;действие1;условие2;действие2;...;условие99;действие99;иначе_действие)*

Оценивает условия: если условие истинно (TRUE), Notes выполняет действие, указанное сразу же после условия. Если же условие не истинно (FALSE), Notes переходит к проверке следующего условия и т. д. Если ни одно из условий не истинно, Notes выполняет «иначе_действие».

Можно указать до 99 условий и соответствующих им действий, и только 1 действие, которое выполняется, если не соблюдается ни одно из указанных условий. Как только находится условие, оцененное как истинное (TRUE), Notes выполняет связанное с ним действие и игнорирует оставшуюся часть выражения.

В своей простейшей форме функция имеет вид: *@If(условие; действие; иначе_действие)*.

Примеры:

`@If(CostOfGoods>=12.45;"Сверх бюджета";"Соответствует")`

Просматривает значения в поле *CostOfGoods*. Если значение больше или равно 12.45, условие истинно (TRUE), и возвращается строка **Сверх бюджета**. Если значение меньше 12.45, условие ложно (FALSE) и возвращается строка **Соответствует**.

`@If(CostOfGoods>=12.45;"Сверх бюджета";"`

В этом примере, если *CostOfGoods* меньше 12.45, возвращается нулевая строка.

`@If(CostOfGoods>12.45;"Сверх бюджета";CostOfGoods<12.45; "Соответствует"; "На пределе допустимого")`

В этом примере *@If* просматривает значения в поле *CostOfGoods*; если значение больше 12.45, возвращается строка **Сверх бюджета**; если нет, Notes переходит к следующему условию. Второе условие также рассматривает значения в поле *CostOfGoods*, и, если значение меньше 12.45, условие истинно, и возвращается строка **Соответствует**. Если же значение равно 12.45, Notes выполняет действие "иначе" и возвращает строку **На пределе допустимого**.

2. @IsNewDoc

Осуществляет проверку на новизну документа. Если документ новый, возвращается значение TRUE, если документ уже сохранен в базе данных, возвращается FALSE.

Эту функцию можно использовать в сочетании с функцией *@If*, чтобы задавать заголовок окна создаваемого документа Notes.

Пример:

`@If(@IsNewDoc;"Новый документ";ФИО)`

В этом случае при создании документа в заголовке окна будет надпись **«Новый документ»**, при сохранении этого документа Notes подставит в заголовок окна содержание поля ФИО.

3. @SetEnvironment(переменная;значение)

Устанавливает или возвращает значение переменной среды в файле NOTES.INI пользователя. *@Environment* возвращает текстовое значение; чтобы использовать возвращенную строку для арифметических действий, необходимо преобразовать ее в число с

помощью функции `@TextToNumber`. Имя переменной должно заключаться в кавычки. Если указан перечень переменных, Notes возвращает значения всех переменных в перечне.

`@Environment` удобно использовать для сохранения специфических установок пользователя. Это единственный способ передачи значений из одной макропрограммы в другую. Такая необходимость может возникнуть при заполнении двух форм документов, в каждую из которых требуется ввести одинаковую информацию.

Пример:

В базе данных «Рецензии на дипломный проект» содержится 2 формы – «Отзыв рецензента» и «Анкетные данные рецензента». При получении рецензии информация о рецензенте должна появиться в обеих формах, но вводить ее нужно только один раз. Для этого в одной форме, по которой заполняется исходный документ, создается поле **изменяемого типа** (например, «ФИО_рецензента»). В другой форме, по которой автоматически заполняется порождаемый документ, создается поле **вычисляемого типа** (например, «ФИО_рецензента_1»).

Кроме того, в **исходном документе** нужно поместить **кнопку**, при нажатии которой будет происходить автоматическое заполнение порождаемого документа.

Кнопка должна содержать следующие команды и функции:

`@SetEnvironment("n";ФИО_рецензента);` – присвоение переменной «n» значения поля ФИО_рецензента.

`@Command([FileSave]);` – команда сохранения исходного документа.

`@Command([Compose];"";"Анкетные данные рецензента")` – команда создания порождаемого документа (форма «Анкетные данные рецензента» должна быть создана заранее).

Каждая команда и функция начинается с новой строки, каждая строка отделяется друг от друга точкой с запятой. После выполнения указанных команд и функций переменная «n» содержится в файле Notes.ini.

В вычисляемом поле порождаемого документа должна находиться следующая формула:

`@Environment("n")` – для данных текстового типа.

Эта функция выбирает из файла Notes.ini переменную "n", считывает присвоенное ей значение поля ФИО_рецензента и подставляет в соответствующее поле порождаемого документа.

3.5. Требования к отчету

В результате выполнения лабораторной работы база данных должна содержать:

1. В одной из форм:

– панель из двух действий (одного системного, например, «*Изменить документ*», и одного пользовательского, например, «*Сохранить документ и закрыть окно*»);

– гиперобъект-кнопку, используемую для автоматического перехода и заполнения полей другой формы документа;

– поле вычисляемого типа, значение которого зависит от значения другого поля в той же форме;

– заголовок окна документа, формируемого по значению поля, однозначно идентифицирующего документ.

2. В одном из представлений – панель из двух действий (одного системного, например, «*Переместить в папку*», и одного пользовательского, например, «*Добавить новый документ*»);

3. Агента, выполняющего какие-либо действия над группой документов.

Контрольные вопросы

1. Что такое действие в *IBM Notes*?

2. В чем отличие панели действий для формы и панели действий для представления?

3. В чем отличие панели действий для формы от гиперобъекта-кнопки?

4. Каково предназначение агента?