

Прикладные механизмы платформы «1С:Предприятие 8.2»

Первый блок. Продвинутый курс.

Глава 1. Базовые приемы работы в конфигураторе

Копирование объектов, проектирование Бд...

Копирование через буфер обмена

- Варианты копирования
 - Одиночные объекты
 - Прикладные объекты
 - Копирование разнородных объектов: Документ -> Справочник
 - Реквизиты, Формы, Макеты
 - Копирование групп
 - Ветки прикладных объектов
 - Реквизиты, Формы, Макеты...
 - Копирование между разными информационными базами
 - Пример ошибочного копирования

Проектирование базы данных

- Встроенных case-инструментов нет
- Связи между таблицами
 - Связь один-ко-многим
 - Реализуется реквизитами в таблицах
 - Подчиненные таблицы
 - Табличные части
 - Связь много-ко-многим
 - Требуется вводить промежуточные таблицы
 - Связь один-к-одному
 - Особый вид, встречается редко

Связь один-к-одному

- Простая реализация связи
 - Объединение данных двух таблиц в одну
- Но все же такая связь иногда необходима!
 - Некоторые поля сущности часто модифицируются
 - Эти поля можно выделить в отдельную сущность
 - В исходной таблице есть большое количество индексов
- Решение связи
 - Однаковые UUID у записей разных таблиц
 - Пример

Глава 2. Обработчики событий

Классы обработчиков, работа с подписками на события...

Виды обработчиков

- Обработчики событий элементов управления формы
 - Обработчики событий форм!
 - Произвольные имена
 - Пустой обработчик
- Обработчики модулей
 - Объекта, Менеджера, Набора записей
 - Приложения, Сеанса, Внешнего соединения
 - Жесткие имена
- Обработчики действий команд
- Подписки на события
 - Возможность доопределения обработчиков событий

Подписки на события

- Использование
 - Модификация прикладного решения без изменения существующих модулей
 - Определение действий, осуществляемых сразу в нескольких объектах
- Ограничение
 - Только подписки на события объекта, менеджера значений, набора записей
 - Подписаться на события формы нельзя, пока что...

Создание подписки на событие

- Указание объекта
 - Выбор одного, группы однородных и разнородных объектов
- Выбор события
 - Отображаются события доступные для каждого выбранного объекта
- Создание обработчика
 - Общий не глобальный модуль
 - Сервер, внешнее соединение
 - Параметры метода

Работа подписки на событие

- Система использует алгоритм
 - 1. Выполняется обработчик события объекта
 - Если он был назначен
 - 2. Выполняется обработчик подписки
 - Если на шаге 1 произошел Отказ, то шаг 2 не выполняется
 - Если назначение несколько обработчиков, то они выполняются в произвольном порядке
 - Неприятные выводы
- Пример создания обработчика «Перед записью»

Подписка на событие в обычном приложении

- Работа подписки в обычном приложении
 - Файловый вариант
 - Клиент-серверный
 - Проверка конфигурации
- Мутабельные значения
 - Объекты, которые могут изменяться при изменении свойств
 - Наборы записей, объекты, таблицы значений
 - Их нельзя передавать между клиентом и сервером
 - Но есть исключения
 - Универсальные коллекции
 - Почему нет проблемы в тонком клиенте?

«Допроведение» документов

- Формирование дополнительных движений
 - 1. В обработке проведения запись наборов не происходит
 - 2. Запись НЗ в основной обработке проведения
- Корректировка движений
 - Обращение к набору записей
 - Баланс – легкая «обновляемость» системы и масштабируемость

Где же формируются проводки!?

- Где выполняется эта проверка?
- Методика поиска кода, диагностирующего ошибку

Ввод по строке

- Настройка механизма
 - Удаление реквизитов, по которым осуществляется ввод по строке
- Последовательность событий
 - Автоподбор
 - Окончание ввода текста
 - Обработка получения данных выбора
 - Обработка выбора значения
 - При изменении
- Переопределение события модуля менеджера
 - Параметры обработчика события
 - Изменение параметров, Отбор
 - Программное формирование списка выбора
 - Реализация LIKE

Глава 3. Объектные типы данных

Отличия от ссылки, специальные свойства объектов...

Объект и Ссылка

- Из базового курса известно
 - Ссылка – для чтения данных
 - Объект – для добавления/изменения/удаления
- Но в чем же принципиальные различия?
 - Ссылка – это всего лишь UUID объекта базы данных
 - Объект – полная информация об элементе, считывается в память
- Обращение к данным
 - Ссылка: выполняется запрос к базе за получением поля
 - Прочитанные значения хранятся в кеше 20 секунд
 - Объект: данные берутся из памяти

Оператор сравнения

- Сравнение ссылок
- Сравнение объектов

Дополнительные свойства объектов

- Не хранимые свойства
 - Обмен данными
 - Можно управлять регистрацией изменений
 - Дополнительные свойства
 - Хранение произвольных свойств объекта во время его жизни
 - Использование
 - Передача дополнительных данных в подписку на событие
 - Передача в модуль формы данных объекта
 - Передача из модуля формы данных объекта
 - Ограничение

Дополнительные свойства и переменные

- Отличия дополнительных свойств и экспортных переменных модуля объекта
 - Пример с подпиской на события
- Дополнительные свойства доступны всегда!
 - Не требуется модификация модуля объекта
- Экспортные переменные модуля объекта инициализируются при получении объекта
 - При создании формы
 - При интерактивной записи
 - ...

Создание собственного «класса»

- ООП в платформе «1С:Предприятие 8» нет
- Доопределение методов и свойств текущих типов данных
- Создание обработки-класса
 - Методы
 - Свойства
- Применение в типовых конфигурациях
 - Обработка «БухгалтерскиеИтоги»

Глава 4. Критерий отбора

Примеры использования объекта

Критерий отбора

- Предназначение
 - Создание правил отбора по реквизитам различных объектов
 - В том числе, по реквизитам табличных частей
 - Примеры
- Это прикладной объект
 - Имеет формы
 - Может быть отнесен к подсистемам
 - Не он не входит в глобальный командный интерфейс

Работа с критерием отбора

- На уровне БД объект представлен отдельной таблицей?
 - Структура хранения БД
 - Т.е. можно в запросах и в формах отображать данные критерия отбора
- Настройка критерия
 - Определение типа значения
 - Состав отбора
- Работа с критерием в пользовательском режиме
 - Отличие от 8.1
 - Панель навигации форм

Программный интерфейс критерия

- Типы данных для критерия отбора
- Создание обработки, отображающей данные критерия отбора
- Открытие формы критерия отбора

Глава 5. Команды

Новый объект в платформе «1С:Предприятие 8.2»

Команды

- Предназначение
 - Отображение новых элементов в командных интерфейсах
 - Глобальном
 - Управляемой формы
 - Выполнение определенных алгоритмов
- Команды – основа управляемого интерфейса
 - Стандартные команды
 - Флаг «Использовать стандартные команды» в объектах конфигурации
 - Произвольные команды
 - Общие команды
 - Глобальный командный интерфейс + интерфейс форм
 - Подчиненные команды конкретного объекта
 - Интерфейс управляемых форм + Глобальный интерфейс!
 - Команды управляемой формы

Свойства команды

- Принадлежность группе команд
 - Глобальный командный интерфейс
 - Интерфейс управляемой формы
- Модуль команды
 - Обработчик команды исполняется на клиенте
 - Возможны серверные вызовы
- Параметр команды
 - Только для команд, входящих в интерфейс управляемой формы
- Принадлежность подсистемам
 - Только для общих команд

Создание независимых команд

- Независимые команды располагаются в глобальном командном интерфейсе
- Общая команда
 - Глобальный командный интерфейс
 - Открытие формы нового документа/элемента в рабочей области
 - Открытие в независимом окне
- Подчиненная команда
 - Аналогичные действия
- Отличия общей от подчиненной команды
 - Принадлежность к подсистемам => Отображение в командных интерфейсах

Модуль команды

- Доступные директивы компиляции
 - На клиенте
 - На сервере
 - На клиенте, на сервере
- Контекста формы нет!
- Обработчик команды
 - Жесткое имя ОбработкаКоманды
 - Клиентский метод
- Взаимные вызовы методов

Директива «НаКлиентеНаСервере»

- Такие методы компилируются в двух контекстах
 - Их можно вызывать как со стороны клиента, так и со стороны сервера
 - Важно понимать!
 - С клиента вызывается клиентский метод
 - С сервера – серверный
 - Пример с использованием директивы компиляции
- Вызов методов общих модулей
 - Варианты
 - Сервер + Вызов сервера
 - Сервер + Клиент (УП)
 - Примеры
- Применимость директивы компиляции

Создание параметризованной команды

- Предназначены для отображения в командном интерфейсе форм
 - Независимые команды не могут иметь параметра
- Возможные варианты расположения
 - Панель навигации формы
 - Командная панель формы
- Тип параметра команды
 - Возможность указания составного типа
- Передача параметров в обработчик команды
- Пример
 - Открытие обработки

Обработка команды

- Параметры обработчика команды
 - ПараметрКоманды
 - Неопределено для независимой команды
 - Ссылка для параметрической команды
 - Что будет для нового объекта?
 - ПараметрыВыполненияКоманды
- Параметры метода ОткрытьФорму()
- Контекст выполнения
 - Данные формы недоступны!

Особенности параметризованных команд

- Hint! Указание параметра, не соответствующего текущему объекту
- Для групп «Панель навигации формы» или «Командная панель формы» нужно обязательно указывать параметры
 - Иначе команды не будут отображены в интерфейсе

Отображение команд

- Независимые команды
 - Отображение в соответствующих подсистемах
 - Возможность отображения в командном интерфейсе рабочего стола
- Параметризованные команды
 - Отображение в формах соответствующего объекта
 - Во всех формах, где основным реквизитом выступает указанный объект, либо таблица этого объекта
 - Возможность отключить видимость команды
 - Для всех объектов, связанных с текущим можно «вытянуть» эту команду
 - Пример
 - Анализируется только реквизиты формы, либо подчиненные основному реквизиту

Настройка прав для команды

- В роли доступен вид доступа для команды «Просмотр»
- При настройке командного интерфейса (глобального, локального)
 - Доступна настройка видимости по ролям для команды

Стандарты размещения команд

- Требуется ли параметр для работы команды?
 - Панель навигации или панель действий
 - Панель навигации формы или командная панель формы
- Изменяется информация в рабочей области?
 - Панель навигации или панель навигации формы
 - Панель действий или командная панель формы
 - Возможные действия
 - Печать
 - Изменение данных
 - Обработка

Если не следовать стандартам?

- Система не ограничивает разработчика в обработке команд
- Можно определить собственные стандарты
 - Но!
 - Теряется «узнаваемость интерфейсов»
 - Команды панели навигации кэшируются
 - Пример, повторное открытие независимого окна из панели навигации

Режим использования команды

- Команды могут выполнять действия над многими объектами
 - Настройка режима
- Параметр Команды
 - Передается массив, даже если выбран один элемент

Видимость команд

- Зависит от
 - Доступности в роли соответствующей команды
 - Функциональных опций
 - Принадлежность команд функциональным опциям
 - Для параметрических команд
 - Право на просмотр параметра (Справочник, Документ ...)

Настройка отображения команды

- Свойства команды
 - Имя, синоним, подсказка
- Отображение команды
 - Текст, картинка
 - Картинка отображается только в командной панели формы
- Горячие клавиши
 - Имеют смысл только для команд формы
- Свойство «Изменяет данные»
 - Не тоже самое, что и «Сохраняемые данные»!
 - Ставит команду в один ряд с «Записать», «Провести», ...

Особенности групп команд

- Настройка отображения для группы команд
 - Имеет смысл только для командных панелей формы
- Группа команд отобразиться не будет, если ни одна из команд текущему пользователю не доступна
- Командная панель формы
 - Команды группы «Важное» отображаются отдельными кнопками
 - Остальные команды отображаются в виде подменю

Глава 6. Построитель отчета и построитель запроса

Его применение в «1С:Предприятие 8.2»

Построитель отчета

- Предназначение
 - Быстрое построение отчетов
 - Содержит возможности визуальной настройки отчета
 - Конструктор выходной формы
- Свойства и методы построителя
 - Свойства
 - Текст
 - Параметры
 - Методы
 - Выполнить
 - Вывод данных
- Применение в платформе «1С:Предприятие 8.2»
 - Только в толстом клиенте

Построитель запроса

- В отличие от построителя отчета
 - Не имеет возможности интерактивной настройки и вывода
- Предназначение
 - Описание запроса с необязательными инструкциями
 {...}
 - Построение запроса к произвольному источнику данных
- Свойства и методы построителя запроса

Расширение конструктора запроса

- Закладка «Построитель запроса»
 - Настраиваются необязательные инструкции
 - Будут включены в результирующий запрос, если будут выполнены настройки
- В конструкторе запроса с обработкой результата расширение не доступно
- Настройки построителя
 - Параметры виртуальной таблицы
 - Выходные поля
 - Условия
 - Сортировка
 - Порядок

Примеры работы с построителем

- Описание текста запроса с необязательными полями
 - Условие в параметрах виртуальных таблиц
 - Нужно описывать только поля, на которые могут накладываться условия
 - Условие в предложении ГДЕ
- Метод ПолучитьТекст()

Обращение к различным источникам

- Свойство «ИсточникДанных»
 - Таблица значений, табличный документ, результат запроса, табличная часть, набор записей регистра
- Пример обращения к таблице значений
 - Задача. На входе есть таблица значений.
 - Колонки – Контрагент, Товар, Количество.
 - На выходе нужно получить дерево, где в качестве измерений будет выступать Контрагент.

Ограничения построителя запросов

- С помощью построителя запросов можно описывать динамические запросы
 - Но только в части
 - Выходных полей
 - Условий ГДЕ и параметров ВТ
 - Полей сортировки
 - Итоговых полей
- Язык построителя запросов не поддерживает временные таблицы!
 - Но пакетные запросы выполнять можно.

Работа с таблицей значений

- Задача.
 - На входе есть таблица значений. Необходимо на ее основе создать новую ТЗ, по некоторому фильтру. Далее новую таблицу нужно свернуть по некоторым колонкам.
- Варианты решения
 - Решение «в лоб»
 - Метод найти строки + ручное создание таблицы
 - Построитель запроса
 - Метод Копировать таблицы значений

Глава 7. Запросы

Новые механизмы, нетривиальные задачи.

Предложение Having

- Предложение Где применяется к исходным данным
 - До выполнения группировки
- Пример
 - Найти поставщиков, от которых было 2 и более закупок
- Предложение Имеющие
 - Применяется для результата запроса
 - После выполнения группировок
 - Имеет смысл только для суммируемых полей
 - Описание в конструкторе запроса

Оператор CASE

- Стандартный синтаксис оператора
 - Case
 - When Exp1=Value1 Then Result1
 - When Exp2=Value2 Then Result2
 - ...
 - Else ResultN
 - End As Alias
- Нестандартный синтаксис
 - Case Expr1
 - When Value1 Then Result1
 - When Value2 Then Result2
 - ...
 - Else ResultN
 - End As Alias

Оператор В

- Несколько синтаксисов оператора
 - Поле В (&Массив)
 - В Иерархии
 - Поле В (&СписокЗначений)
 - В Иерархии
 - Поле В (&Значение1, &Значение2)
 - В Иерархии
 - Поле В (Выбрать ...)
 - (Поле1, Поле2) В (Выбрать Т.Поле3, Т.Поле4 Из Таблица Т)
 - Поле В (&ТаблицаЗначений)
 - (Поле1, Поле2) В (&ТаблицаЗначений)

Предложение Выбрать

- Альтернативный синтаксис
 - Выбрать Документ.ПоступлениеТоваров.Ссылка
 - Преобразование этого текста конструктором запроса
- Применение
 - Выборка всех данных из одной или нескольких таблиц

Строки неограниченной длины

- Использование в запросах строк неограниченной длины может привести к ошибке
 - В СУБД MS SQL Server 2000
- Лучше избегать использования неограниченных строк
 - Если длину нельзя ограничить, то использовать
 - Подстрока(Реквизит, 1, 200)
 - Выразить(Реквизит КАК Страна(200))
 - В этом случае строка будет дополнена пробелами

ФУНКЦИЯ LIKE

- ПОДОБНО
 - Логический оператор
 - Поиск строк по указанному шаблону
- Описание шаблона
 - %
 - _
 - [...]
 - Символ или диапазон
 - [^...]
 - Пример
 - %-[0-9]-[А-Я]_
- Задача
 - Определить строки состоящие из 3-х слов

Обработка типов данных

- Для полей составного типа данных
 - Оператор ССЫЛКА
 - Проверка является ли поле ссылкой на указанную таблицу
 - Возвращает Истину или Ложь
 - Оператор ВЫРАЗИТЬ
 - Выразить(Поле КАК Тип)
 - Тип – таблица или примитивный тип данных
- Функция ТипЗначения()
 - Возвращает Тип

Особенность Full Outer Join

- Вариант работы клиент-серверный, СУБД PostgreSQL
 - Производительность работы полного внешнего соединения не высокая
 - Лучше переписать запрос с использованием правого или левого соединений

Упорядочивание в запросах

- Явное упорядочивание следует использовать
 - В отчетах
 - В тех запросах, результаты которых будут представлены пользователю
- Если упорядочивания не использовать
 - Порядок записей неизвестен
 - На одном и том же запросе (и наборе данных) порядок может измениться
 - При настройке или изменении СУБД
- Автоупорядочивание
 - Лучше не использовать, так как порядок записей в этом случае не прозрачен

Особенности упорядочивания

- Поля сортировки не обязательно должны входить в поля выборки
- Но! При использовании предложения «Различные» все поля сортировки должны быть полями выборки

Особенности сравнения строк

- При сравнении строк во встроенном языке регистр учитывается
- Во всех операциях с БД сравнение строк происходит без учета регистра

Выгрузка результата запроса

- По умолчанию выгрузка происходит в таблицу значений
- Для запроса с итогами можно выгрузить результат в дерево значений
 - Параметр метода
 - Пример

Временные таблицы

- Область применения
 - Разработка сложных запросов
 - Оптимизация запросов
 - Выделение общих частей запроса в одну временную таблицу
 - Уменьшение количества вложенных запросов
 - Обращение в запросе к внешним источникам данных
 - Построение результата с итогами по данным ТЗ
- Хранение временных таблиц
 - Файловый вариант
 - Каталог временных файлов
 - Клиент-серверный вариант
 - База TempDB
- Временная таблица доступна только в текущем сеансе
- Время жизни ограничено

Создание временных таблиц

- Интерфейс работы с временной таблицей обеспечивает объект МенеджерВременныхТаблиц
 - Одноименное свойство запроса
- Предложение «Поместить» языка запросов
 - Настройка в конструкторе запроса
- Обращение к временной таблице из другого запроса
- Время жизни временных таблиц
 - До уничтожения менеджера временных таблиц
 - Явное уничтожение, предложение «Уничтожить»

Индексирование

- Предназначение
 - Более эффективное выполнение отборов
- Настройка индексирования в конструкторе
- Поля, которые нужно индексировать
 - Использующиеся в предложении «Где»
 - Использующиеся в условиях соединений

Обращение к внешним источникам

- В качестве параметра в запрос можно передавать
 - Таблицу значений
 - Результат запроса
 - Табличную часть
 - Этот параметр можно использовать только как источник данных для временной таблицы
 - Предложение Поместить
- 2 итерации
 - Помещение внешнего источника во врем. таблицу
 - Обращение к ней в последующих запросах

Работа с внешним источником

- Ограничения
 - Колонки таблицы значений должны быть строго типизированы
 - В запросе, создающем временную таблицу нельзя использовать join И union
 - Нельзя выполнять разыменование полей ТЗ
 - Нельзя использовать предложения
 - Сгруппировать По, Имеющие
 - Итоги

Работа с вложенными запросами

- Использование вложенных запросов
 - Исходные таблицы
 - Условия
 - Предложение ГДЕ
 - В параметрах виртуальных таблиц
 - В условиях связи таблиц
- Приемы редактирования и создания вложенных запросов

Пакет запросов

- Выполнение нескольких запросов в рамках одного «обращения» в СУБД
- Предназначение
 - Выполнение нескольких независимых запросов
 - Независимое обращение к результатам
 - Построение многошагового запроса
 - Последовательное выполнение преобразований
 - На начальных этапах создаются временные таблицы
 - На заключительных – обращение к ВТ

Использование пакета запросов

- Синтаксис пакета
- Описание пакета запросов с помощью конструктора
 - Представление в конструкторе: пакет + объединения
- Обращение к результатам пакета
 - Метод Выполнить()
 - Метод ВыполнитьПакет()
 - Коллекция результатов
- Использование временных таблиц
 - Hint! Менеджер временных таблиц создавать не обязательно
 - Если в следующих запросах не потребуется обращение к созданным таблицам

Практическая задача #1

- Получение остатков по номенклатуре, указанной в документе.
 - Использование временных таблиц + пакет запросов

Практическая задача #2

- Определение последнего документа продажи по каждому товару
 - Максимум(Ссылка) – не верный результат

Практическая задача #3

- Получить цену на товар из прайс-листа на дату каждой продажи
 - Сравнение цены из прайса и отпускной
 - Задача «Срез последних на каждую дату из другой таблицы»

Практическая задача #4

- В документе отгрузки указывается товар и качество
 - Если качество задано, то система прежде всего должна попытаться списать товар с этим качеством
 - Если его не хватает, то берется товар любого другого качества
- Вариант решения №1
 - Левое соединение документа с регистром по товару и качеству
 - Не учитывается, что в одном документе может быть один и тот же товар разного качества
 - Дополнительная обработка на клиенте
- Вариант решения №2
 - Пошаговое выполнение запроса

Практическая задача #5

- Регистр «Взаиморасчеты»
 - Измерение
 - Контрагент
 - Ресурс
 - Сумма
- Задача
 - Определить дату оплаты каждой накладной
 - Структуру регистра модифицировать нельзя
- Решение
 - Определение долга по каждому документу
 - Определения таблицы оплат
 - Вычисление, когда был «закрыт» долг по текущему документу

Практическая задача #6

- Получение любого контактного лица для контрагента
 - Справочник контактных лиц подчинен контрагентам

Практическая задача #7

- Получение данных на каждый день из указанного интервала
 - Задача. Заполнение регистра «График работы» данными по рабочим и выходным дням
- Генерация числовой последовательности
 - Формирование последовательности дат
 - Ограничение по диапазону

Практическая задача #8

- Для каждой номенклатуры нужно определить «Частоту продаж»
 - Рассчитывается по формуле (дата последней продажи – дата первой продажи) / количество сделок

Древовидные структуры

- Деревья и графы
 - Работа с деревьями в «1С:Предприятии» возможна благодаря иерархическим структурам
 - Оператор языка запросов «В иерархии»
- Классы задач
 - Построение производственного дерева «Продукция – полуфабрикаты»
 - Задача учета аналогов
- Наиболее частые операции
 - Вывод в виде дерева
 - Получение всех потомков текущего узла, независимо от количества уровней вложенности

Работа с древовидными структурами

- Задачи
 - Вывод в виде дерева
 - Получение всех потомков текущего узла
- Для решения требуется рекурсивный алгоритм
 - На больших объемах данных – занимает длительное время
- Варианты решения
 - Если ограничена глубина вложенности, то выполнения нескольких соединений
 - Хранение всех взаимосвязей между потомками
 - Если структура является деревом, то ее можно описать в иерархическом справочнике

Глава 8. Оптимизация производительности прикладных решений

Правила написания кода на встроенном языке...

Обращение к данным объектов

- Чтение реквизитов объекта
 - Для чтения значений реквизитов неправильно получать ссылку или объект
 - Пример
 - В этом случае весь объект считывается из базы данных
 - У него может быть десяток заполненных табличных частей
 - Решение
- Обращение через несколько точек
 - Нужно стараться «сократить» путь до получения значения
 - Если обращение через точку происходит несколько раз

Эффективная манипуляция данными

- Запросы в цикле
 - Неявные запросы
 - Пример
- Проверки при записи справочников/документов
 - Следует выносить в «ОбработкуПроверкиЗаполнения»
 - Это событие выполняется до начала транзакции записи/проведения
 - Позволяет обрабатывать ошибки, не выполняя блокировок
- Тип данных ЛюбаяСсылка
 - Можно столкнуться с ограничением MS SQL Server
 - В запросе может участвовать не более 256 таблиц

Хранение данных в объекте

- Хранение в объекте большого объема данных
 - Реквизит типа «Хранилище значения»
 - Картинки, файлы, видео
- При получении данных объекта (Ссылка, Объект) происходит полное чтение данных объекта
- Правильный вариант хранения
 - Вспомогательный справочник со ссылкой на текущий элемент/документ
 - В этом справочнике реквизит типа «Хранилище значения»

Прочие приемы кодирования

- Вывод в табличный документ
 - В качестве полей нужно указывать не ссылки, а представления
 - Чтобы не было лишних обращений к базе
 - Ссылку можно помещать в расшифровку
- Обращение к метаданным
 - Обращение к функции Метаданные() более быстрое, чем к свойству глобального контекста
- Метод ЗаполнитьЗначенияСвойств
 - Позволяет выполнить копирование строк любых коллекций

Неочевидные вопросы производительности

- Использование арифметических операций
 - * vs /
- Размещение строк кода
 - Код в одной строке работает быстрее
- Влияние на производительность указанных «особенностей»

Преобразование рекурсии в цикл

- Количество рекурсивных вызовов ограничено
 - Зависит от релиза платформы
- При выполнении вызовов процедур система сохраняет служебную информацию
 - Стек вызовов, контекст методов
 - Все это делает рекурсивные вызовы не оптимальными с точки зрения производительности
- Решение – преобразовать рекурсию в цикл
 - Пример
 - Обход дерева значений двумя алгоритмами

Работа с таблицей значений

- Активное использование таблиц значений – тяжелое наследие «1С:Предприятия 7.7»
 - Большинство задач можно решить с использованием запроса
- Медленные операции для таблицы значений
 - Поиск строк
 - Решение – возможность индексирования
 - Свойство Индексы таблицы значений
 - Сортировка по полям ссылочного
 - Система за представлением каждой ссылки будет обращаться в БД
 - Решение – включение представления в таблицу значений
- Однако таблица значений используется во всех типовых алгоритмах по формированию движений
 - Почему??

Разделение функциональности

- Это один из самых эффективных приемов по повышению масштабируемости систем
 - Функциональность решения можно разделить на
 - Оперативную
 - Нужно здесь и сейчас. Для принятия правильных решений
 - Примеры. Остатки товаров на складе, долги клиентов
 - Неоперативную
 - Может быть рассчитана позже, по регламенту.
 - Например. Стоимостная оценка товаров, прибыль, НДС...
 - Применение подхода в типовых конфигурациях
 - Партионный учет
 - Отложенное проведение

Оперативное проведение документов

- Система может отслеживать проведение документов актуальным временем
 - Эту информацию можно эффективно использовать для получения «быстрых» данных
 - Пример, различные проверки при проведении документов
 - Проверки, обращающиеся к внешним данным нельзя выполнять в «ОбработкеПроверкиЗаполнения»!!
 - Реализация в типовых конфигурациях
 - Виды доступа для документов
 - Неоперативное проведение

«Мега-документы»

- Концепция мега-документов
- Проблемы документов, с большим количеством строк
 - Длительная блокировка объектов при проведении
 - Сложность редактирования
 - Поиск нужных данных в документе
 - При любом изменении документ перепроводится
- Решение проблемы
 - Создание множества «легких» документов
 - Объединение их в единое пространство для редактирования с помощью обработки
 - Пример, «Ввод начальных остатков» в «1С:Бухгалтерии 8»

Итоги регистров

- Регистры накопления и бухгалтерии
- Использование итогов позволяет значительно увеличить скорость получения данных
 - Нужно убедиться, что
 - Итоги рассчитываются ежемесячно
 - Включены текущие итоги
- Режим разделения итогов
 - Позволяет параллельно осуществлять операции записи по конкурирующим ресурсам
 - Когда сворачиваются дубли записей?

Глава 9. Оптимизация производительности при работе с БД

Оптимизация запросов к БД...

Оптимизация запросов к БД

- Важность вопроса
 - Работа с БД зачастую является самой ресурсоемкой
 - Запись
 - Длительная операция
 - Программным образом на нее не повлиять
 - Наращивать аппаратную мощность
 - Чтение
 - Существует возможность оптимизации запросов
 - Любую задачу можно решить разными путями
 - Задача разработчика – выбрать оптимальный путь
 - «Серебряной» пули не существует
 - Для каждого набора данных может быть свой оптимальный алгоритм
 - Но есть ряд общих правил

Использование вложенных запросов

- Вложенные запросы могут выступать как
 - Таблицы-источники
 - В условиях ГДЕ, условиях соединений, условиях виртуальных таблиц
- Негативное влияние на производительность
 - При выполнении любого запроса СУБД строит план выполнения запроса
 - При использовании вложенных запросов оптимизатор может выбирать неверных план выполнения запроса
 - Так как он ориентируется на количество записей в таблицах
 - Количество записей во вложенном запросе определить сложно
- Решение
 - Использование временных таблиц
 - Создание таблицы потратиться некоторое время
 - Оно будет компенсировано оптимальным планом
 - В этом случае запрос будет вести себя «стабильно»

Виртуальные таблицы

- Условия на данные виртуальной таблицы необходимо накладывать в ее параметрах
 - Исключение – ресурсы
- Виртуальная таблица – это вложенный запрос
 - Не следует
 - В параметрах ВТ указывать вложенные запросы
 - Виртуальную таблицу описывать во вложенном запросе
- Для реализации наиболее стабильных запросов для виртуальных нужно создавать временные таблицы
 - Только, если кроме данных ВТ выбираются данные из других таблиц
- Оптимизация алгоритма контроля остатков

Индексирование временных таблиц

- Все поля, по которым будут выполняться условия необходимо проиндексировать
 - Условия предложения «ГДЕ»
 - Условия соединения
- В этом случае запрос будет выполняться не перебором записей таблицы, а поиском по индексу

Получение остатков на товарам

- Варианты решения задачи
 - Использование вложенного запроса
 - Передача списка товаров как массив
 - Создание временной таблицы
 - Наиболее универсальный вариант
- Передача периода
 - Момент времени
 - Оперативное проведение документа!

Разыменование полей

- Удобная и опасная возможность
- Важно понимать, что любое разыменование – дополнительное соединение с таблицей
 - Пример неправильного использования механизма
- Разыменование полей составного типа данных
 - Пример, получение даты регистратора в запросе
 - Опасность
 - Тип, «ЛюбаяСсылка», «ДокументСсылка», ...
 - Решение
 - Хранить необходимые данные в объектах
 - Регистр должен хранить все необходимые данные
 - Уменьшение количества типов с использованием оператора «Выразить»

Использование объединений

- При объединении желательно указывать ключевое слово «ВСЕ»
 - Поскольку система тратит дополнительные ресурсы на устранение дублей
 - Даже если известно, что их не будет
 - По умолчанию объединение настраивается именно таким образом
- Предложение «Без повторяющихся»

Попадание в индекс

- Если условие запроса соответствует существующим индексам
 - Запрос будет исполняться максимально быстро
 - Непопадание в индекс будет означать перебор записей таблиц – сканирование
- Правила
 - Все поля условия должны входить в индекс
 - Они должны быть первыми
 - В системе часто используются составные индексы
- Индексы влияют на скорость сортировки списков
 - Система строит ряд индексов по умолчанию
- Примеры запросов
 - К регистрам

Основные индексы системы

- Документ
 - Ссылка, Кластерный индекс
 - Дата + Ссылка, Номер + Ссылка
- Таблицы движений регистров
 - Период + Регистратор + Номер строки, Кластерный индекс
 - Регистратор + Номер строки
- Итоговые таблицы регистров
 - Период + Измерение1 + ... + ИзмерениеN
 - Период + Счет + Измерение1 + ... + ИзмерениеN

Создание собственных индексов

- Для регистров свойство «Индексировать» принимает два значения
 - Создается индекс
 - Измерение + Период + Регистратор + Номер строки
 - Измерение + Период
- Реквизиты документов/справочников
 - Индексировать
 - Реквизит + Ссылка
 - Индексировать с доп. упорядочиванием
 - Реквизит + Дата + Ссылка
- Ограничение
 - Строки неограниченной длины

Принципы выбора индексов

- На стандартные индексы системы нельзя повлиять
 - Нужно стараться их использовать по максимуму
- Когда нужно создавать собственные индексы?
 - Если по некоторому реквизиту происходит отбор данных
 - При этом производительность данных запросов является критичной
 - Исполняется в транзакции
 - Выбирается большой объем данных, длительное время выполнения
 - Индексы
 - Увеличивают объем базы данных
 - Увеличивают время записи объектов

Получение огромных выборок данных

- Получение запросом огромного массива информации
 - Может быть неадекватно отработано платформой
 - Особенно если есть тенденция к росту записей
- Решение
 - Поблочная выборка данных
 - Пример

Регламентные операции СУБД

- С течением времени производительность запросов может падать
 - Причины
 - Неверная статистическая информация о содержимом таблиц
 - Система может строить неверный план выполнения запроса
 - Некорректный кэш планов выполнения запроса
 - Процедурный кэш
 - Фрагментация индексов
 - Решение
 - Регулярно выполнять регламентные операции СУБД

Регламентные операции MS SQL

- Создание Maintenance Plan
 - Update Statistics Task
 - Может выполняться параллельно с работой пользователей
 - Создает дополнительную нагрузку на систему
 - Лучше выполнять в часы наименьшей активности
 - Очистка процедурного кэша
 - t-sql: DBCC FREEPROCCACHE
 - Может выполняться параллельно с работой пользователей
 - Reorganize Index Task
 - Не блокирует работу пользователей
 - Rebuild Index Task
 - Включает в себя дефрагментацию индексов
 - Блокирует таблицы на время перестройки индексов
 - Существенно оптимизирует работу индексов!!

Контроль за ходом операций

- Администратор должен следить за ходом выполнения регламентных операций
 - Особенно за созданием резервных копий!
- Просмотр «вручную»
- Автоматизация
 - КИП

Получение текста запроса

- Использование SQL Server Profiler
- Использование технологического журнала
 - Работает даже для файлового варианта!
 - Конфигурация logcfg.xml
 - Включение ТЖ

Трассировка SQL Server Profiler

- Включение трассировки
 - Поиск нужного запроса
- Сохранение трассировки в файл
- Наложение фильтров
 - Вид события
 - Длительность выполнения запроса

Поиск запросов с помощью ТЖ

- Настройки технологического журнала
 - Фильтры на виды событий
 - Фильтры на свойства событий
 - Фильтры на атрибуты
 - Duration
- Пример выполнения запроса с параметром
 - Запрос к справочнику
 - Запрос к ВТ Остатки регистра накопления

Планы построения запросов

- Для получения данных используется запрос
 - Это лишь описание желаемого результата
- База данных
 - Набор таблиц
 - Данные хранятся в виде страниц
 - Индексы
 - Любой запрос может быть исполнен по разному
 - Результат будет один и тот же, но время выполнения разное
 - Для исполнения запроса СУБД строит план выполнения запроса
 - Что, откуда и в какой последовательности выбирать
 - Задача разработчика
 - Помочь СУБД выбрать верный план выполнения

Правильный план выполнения

- Напрямую нет возможности указать СУБД «правильный» план выполнения
- Факторы влияющие на выбор верного плана
 - Выполненные регламентные операции
 - Использование индексов
 - Отсутствие вложенных запросов
- Просмотр плана выполнения запроса
 - Microsoft SQL Management Studio
 - Просмотр плана выполнения простейшего запроса
 - Дополнительная информация при наведении курсора

Анализ плана выполнения запроса

- Когда нужно анализировать планы выполнения?
- 99% проблем снимается, если
 - Регулярно выполнять регламентные операции
 - Следовать рекомендациям по написанию запросов
- Спускаться на «глубокий» уровень приходится крайне редко
 - «Ручной» способ является трудоемким
 - Центр управления производительностью

Диагностика «неверного» плана

- Операции, на которые стоит обратить внимание
 - Table Scan
 - Сканирование, перебор записей таблицы
 - Плохо когда, из большой таблицы выбирается немного записей
 - Возникает при отсутствии, либо неправильном индексе
 - Index Scan
 - Сканирование индекса
 - Возникает, если не все поля, использующиеся в условии входят в индекс
 - Nested Loops
 - Объединение таблиц вложенным циклом
 - Самый простой способ для СУБД
 - Возникает, если СУБД считает, что в одной из таблиц немного записей – вложенные запросы или неверные статистики
 - Не факт, что эти операции являются неоптимальными

Глава 10. Разработка управляемых форм

Типы данных, клиент-серверное взаимодействие...

Управляемые формы

- Состав формы не является статичным
 - Зависит от
 - Прав доступа на объекты
 - Значений функциональных опций
 - Видимости элементов по ролям
 - Настроек пользователя
 - Команд объекта и общих команд
- Форма описывается декларативно
- Не все задачи решаются с помощью управляемых форм!
 - Возможность использования обычных форм в управляемом приложении

Реквизиты и данные формы

- Значения, хранящиеся в реквизитах формы называют данными формы
- Предназначение реквизитов
 - Отображение информации на форме
 - Возможность редактирования
 - Хранение значений, необходимых для жизнедеятельности формы
- Возможно обращаться к реквизитам извне
 - Пример
 - Но! Такой подход не является рекомендуемым!

Основные свойства реквизитов

- Основной реквизит
- Принадлежность к функциональным опциям
- Просмотр и редактирование
- Сохраняемые данные, влияет на
 - Модифицированность формы
 - Недоступность редактирования при открытии формы режиме ReadOnly
 - Блокировку объекта при редактировании свойства
- Проверка заполнения
 - Для примитивных типов, ссылок, ТЗ, СЗ, ДЗ

Признаки реквизитов

- Тип данных
- Возможность сохранения
- Признак отображения на форме
- Использовать всегда
 - Принудительное считывание данных, даже если реквизит не отображается
 - Пример
 - Доступен не для всех типов данных

Клиент-серверное взаимодействие

- Данные формы существуют одновременно и на клиенте и на сервере
 - Происходит периодическая синхронизация данных
 - Форма также присутствует на сервере в момент создания
 - Далее она удаляется
- Объект присутствует только на сервере
 - На клиенте – отображение объекта
 - Даже на сервере объект существует не все время
 - Возможно преобразование данных формы в «настоящий» объект
- Различные контексты исполнения кода, разные типы данных

Механика открытия формы

- Открытие формы документа
 1. Получение ДокументОбъект из СУБД
 2. Преобразование ДокументОбъект в данные формы (сервер)
 3. Передача данных формы с сервера на клиент
 4. Отображение формы
- Запись документа
 - Обратный порядок

Типы данных тонкого клиента

- Примитивные типы данных
- Ссылки
 - Сильно ограниченный функционал
- Универсальные данные формы
 - Предназначены для отображения типов, недоступных на клиенте
 - Полностью повторяют структуру отображаемого объекта
 - Но не наследуют его методов
 - В редакторе форм отображаются в скобках
- Почему так сильно «ущемили» тонкий клиент?

Универсальные типы данных формы

- ДанныеФормыСтруктура
 - Для отображения объектов
- ДанныеФормыКоллекция
 - Для отображения табличных частей, таблиц значений
- ДанныеФормыСтруктураСКоллекцией
 - Отображение набора записей регистра
- ДанныеФормыДерево
 - Отображение дерева значений
- Динамический список
 - Отображение списка элементов, документов
- Запрещено присваивание значений реквизитам, содержащим универсальные типы

Преобразование данных формы

- Возможны взаимные преобразования
 - Данные формы $\leftarrow \rightarrow$ «Настоящий» объект
- Функции глобального контекста
 - ДанныеФормыВЗначение
 - ЗначениеВДанныеФормы
 - КопироватьДанныеФормы
- Функции управляемой формы
 - РеквизитФормыВЗначение
 - ЗначениеВРеквизитФормы
- Примеры
 - Получение объекта, вызов экспортного метода
 - Модификация объекта

Передача данных формы на сервер

- Ссылки, примитивные типы можно свободно передавать на сервер
- Универсальные типы нельзя передавать на сервер
 - Даже в контекстные вызовы
 - Хотя в такой передаче смысла нет
 - Hint! Ошибка происходит именно в момент окончания серверного вызова
- Причина
 - Параметры передаются по ссылке
 - Выход
 - Передача по значению
 - В этом случае между клиентом и сервером будет передаваться большой объем информации
 - Полностью структура
 - Сопутствующие данные
 - Либо использовать контекстные вызовы

Особенность передачи данных формы

- Передача данных формы на сервер не рекомендуется, но все же допустима!
- Пример
 - Использование буферной переменной
 - Модификация данных на сервере
- Когда можно использовать этот подход
 - Особенность! Штатные механизмы оптимизации не будут работать!
 - Пример
 - Для того чтобы избежать синхронизации между клиентом и сервером можно очистить содержимое параметра

Осуществление контекстных вызовов

- Механика контекстного серверного вызова
 - С клиента на сервер передается контекст
 - Происходит синхронизация формы с сервером
 - Данные формы
 - Параметры формы
 - Контейнер «Элементы»
 - Метод выполняется
 - Обратная передача с сервера на клиент
 - Вывод. Все изменения отобразятся на форме, только после окончания серверного вызова
- Происходит передача только измененных данных
- На клиент возвращаются только отображаемые данные
 - Характерно для больших списков

Глава 11. Открытие управляемых форм

События возникающие при открытии, программное
открытие...

Пошаговое открытие формы

- Шаг 1
 - Имеет смысл только для открытия существующих объектов
 - Получение данных из СУБД в объект. На сервере
 - Преобразование объекта в данные формы
 - Обработчик ПриЧтенииНаСервере
 - Доступен объект в явном виде!!
- Шаг 2
 - Обработчик ПриСозданииНаСервере
 - Отказ + стандартная обработка
 - Объект на стороне сервера уже удален
- Шаг 3
 - Передача данных формы с сервера на клиент
 - Обработчик ПриОткрытии
 - Отказ

Открытие формы

- Метод ОткрытьФорму
 - Параметры
- Метод ПолучитьФорму
 - Две парадигмы влияния на функциональность формы
 - Экспортные методы формы
 - Имеют смысл только клиентские методы
- Модальное открытие и анализ результата возврата
 - Метод Закрыть()

События открытия формы

- Последовательность обработчиков
 - Обработка заполнения
 - Возникает при интерактивном открытии
 - При чтении на сервере
 - Возникает только для существующих объектов
 - При создании на сервере
 - Параметры
 - Отказ
 - Стандартная обработка – обработка параметров, переданных извне
 - При открытии
 - Еще есть шанс отказаться от открытия

Глава 12. Параметры управляемых форм

Предназначение и практическое применение...

Параметры формы

- Применение
 - Определение внешнего вида/функциональности формы
 - Передача параметров извне
 - Метод ОткрытьФорму
- После того как форма открыта параметры удаляются
 - «Мавр сделал свое дело, мавр может уходить»
 - Исключение – ключевые параметры
- Если значения параметров требуется сохранить, то их можно запомнить в реквизитах формы

Использование параметров

- Закладка «Параметры»
 - Ключевые параметры
 - Передаваемые параметры можно не описывать
 - Но тогда нельзя их будет сделать ключевыми
- Стандартные параметры
 - Контекст управляемой формы
 - Пример для формы элемента справочника
 - Параметры управляемой формы + параметры основного реквизита формы

Параметры форм объектов

- Ключ
 - Самая быстрая проверка на пустоту
- Значения заполнения
- Значение копирования
- Основание

Параметры форм справочников

- Форма элемента справочника
 - Это группа
- Форма списка справочника
 - Параметры характерны для всех форм, где основной реквизит – динамический список
 - Текущая строка
 - Отбор
 - Множественный выбор
 - Режим выбора

Параметры форм отчетов

- Отбор
- Сформировать при открытии
- Вариант

Глава 13. Запись в управляемых формах

Последовательность обработчиков событий, передача данных между объектом и формой...

Варианты записи

- Интерактивно
- Программно
 - Неправильно!
 - Получить объект и записать
 - Данные формы не обновятся
 - Правильно
 - Использование методов расширения формы
 - Доступны на клиенте!

События при проведении документа

- 1 Этап. События проверки заполнения
 - Перед записью – форма
 - Клиентский метод
 - Обработка проверки заполнения на сервере - форма
 - Особенности параметров
 - Обработка проверки заполнения
 - Перед записью на сервере – форма

События при проведении документа

- 2 Этап. Запись объекта

Начало транзакции

- Перед записью – объект

Запись объекта в БД

- При записи
- Обработка проведения
- При записи на сервере

Окончание транзакции

События при проведении документа

- 3 Этап. Завершающие операции
 - После записи на сервере
 - После записи
- 4 Этап. Закрытие формы
 - Перед закрытием
 - При закрытии

Передача данных из формы в объект

- Дополнительные свойства
 - Нужно установить именно для того объекта который будет проводиться!
- Экспортная переменная модуле объекта
 - Установить именно для проводимого объекта!
- Параметры записи
 - Возможна передача данных в обработчиках формы
 - Передача между клиентом и сервером
- Передача данных из модуля объекта в форму
 - При записи возможно
 - При создании нового объекта возможности нет

Глава 14. Отображение и настройка списков

Динамический список, таблица формы, табличная часть....

Динамический список

- Предназначен для отображения списков любой информации из базы данных
 - Предполагается возможность добавления/редактирования элементов
- Настройки унаследованы от системы компоновки данных
- Система предполагает, что списки могут быть большими
 - Данные перекачиваются на клиента порциями
- В качестве источника данных может выступать произвольный запрос

Настройка динамического списка

- Указание источника данных
 - Основная таблица
 - Ключ основной таблицы должен быть уникален в списке
 - Произвольный запрос
- Настройки динамического списка
 - Интерактивные в конфигураторе и пользовательском режиме
- Динамическое считывание данных
 - Данныечитываются на клиента оптимальными порциями
- Динамический список для дерева групп

Использование произвольного запроса

- Определение основной таблицы
- Задача. Выводить информацию о цене номенклатуры
 - Установка параметров запроса

Ограничения запроса

- Не следует увлекаться тяжелыми запросами
 - Не возможно описывать пакетные запросы
 - Использовать временные таблицы
- Если определена основная таблица, то нельзя
 - Выполнять сортировку
 - Объединение запросов
 - Использовать группировку
 - Указывать Первые и Различные

Программная настройка списка

- Создание нового отбора
 - Как сделать элемент отбора недоступным?
 - Не забывайте про параметры управляемой формы
- Упорядочивание списка
- Настройка группировки
- Применение условного оформления

Настройка отображения табличной части

- Отбор для табличной части
 - Возможен только программным образом
 - В отличие от 8.1
- Сортировка
 - Метод «Данные формы коллекция»
- Условное оформление
 - Нет возможности интерактивной настройки в Предприятии
 - Свойство управляемой формы
 - Доступно – только на сервере
 - Пример работы

Отображение полей в таблице формы

- Создание дополнительной колонки
- Динамический список
 - Перетянуть поле в элементы формы
 - Действие допустимо для вложенных реквизитов
 - Можно создавать собственные поля динамического с помощью запроса
- Табличная часть
 - «Перетягивание» поля также допустимо
 - Создание колонки реквизита
 - Доступно редактирование такой колонки!
 - Обработка через «Текущие Данные»
 - При создании на сервере
 - При изменении

Глава 15. Элементы формы

Расширения элементов, связи параметров выбора...

Элементы формы

- Типы элементов формы
 - Поле формы
 - Кнопка формы
 - Таблица формы
 - Декорация формы
 - Группа формы
- Расширения в зависимости от вида элемента
 - Пример расширения поля формы
- Основные события поля формы

Особенности ряда обработчиков

- Во многих обработчиках элементов формы нельзя делать контекстные серверные вызовы
- Пример
 - События расширения поля ввода
- Событие «ПриАктивацииСтроки»
 - Подключение обработчика ожидания

Параметры выбора

- Можно настраивать связи между элементами формы
 - Устанавливать отборы в открываемых формах
 - Пример, Контрагент – Договор
- Лучше это делать на уровне реквизитов объекта!
- Обработка параметров выбора в модуле менеджера
 - Событие «Обработка получения данных выбора»
 - Пример

Глава 16. Команды формы

Виды команд, их использование...

Виды команд

- Виды команд
 - Собственные
 - Команды объекта + общие команды
 - Стандартные
 - Глобальные
- Характерные свойства
 - Принадлежность к функциональным опциям
 - Настройка отображения

Собственные команды

- Создание собственных команд
 - Представление в виде кнопки или гиперссылки
 - Свойства
 - Изменение сохраняемых данных
- Обработчик «Действие»
 - Параметр
- Размещение
 - Независимая кнопка
 - Кнопка командной панели
 - Системная командная панель

Стандартные команды

- Команды формы
 - Зависят от типа основного реквизита
- Команды таблиц формы
 - На каждую таблицу – отдельная группа
- Команды – Да, Нет, Отмена, ...
- Размещение
 - Независимая кнопка
 - Кнопка командной панели
 - Свойство «ТолькоВоВсехДействиях»
 - Системная командная панель

Глобальные команды

- Команды всей конфигурации
 - Независимые
 - Не важно выведены они или нет!
- Команды связанных объектов
 - Все команды, связанные с данными объекта
 - Наборы записей
 - Подчиненные справочники
 - Параметризуемые команды реквизитов формы
- Размещение
 - Независимая кнопка
 - Кнопка командной панели
 - Системная командная панель
 - Панель навигации

Глава 17. Программная модификация форм

Особенности, примеры...

Программная модификация формы

- Когда необходимо?
 - Легкая модификация типовой конфигурации
 - Отображение данных, недоступных в конфигураторе
- Возможно программно управлять составом
 - Реквизитов
 - Элементов формы
 - Команд
- Ограничения
 - Платформа ориентируется на то, что управляемые формы являются «статическими»
 - При программной модификации
 - Отключается оптимизация форм
 - Возможно замедление работы с формой
 - Нельзя создать основной реквизит формы!

Контейнеры формы

- Элементы
- Команды
- Реквизиты
- Во всех контейнерах возможно программное добавление
 - Но нужно выполнить «полный цикл», например
 - Создать реквизит формы
 - Создать элемент формы
 - Установить между ними связь

Создание и отображение реквизита

- Задача №1
 - Создать новый элемент формы, связанный с реквизитом объекта
- Задача №2
 - Создать реквизиты формы и отобразить его на форме
 - Создать обработчик события
- Задача №3
 - Создание команды и отображение в виде кнопки

Глава 18. Оптимизация работы управляемых форм

Клиент-серверное взаимодействие...

Необходимость оптимизации

- Создавая управляемую форму нужно учитывать, что она может использоваться на тонких каналах связи
 - Тонкий клиент
 - Веб-клиент
- При разработке нужно учитывать
 - Количество обращений к серверу
 - Вызов на тонком канале может составлять 1.5 секунды
 - Объем трафика
 - Платформа выполняет «стандартные» серверные вызовы
 - Пример

Имитация «боевых» условий

- Режим имитации задержек
 - Ручной тормоз
- Низкая скорость соединения
 - Передача данных пакетами
 - Сжатие данных
 - Не отображается рабочий стол и картинки разделов
 - Не формируется список выбора при вводе по строке
- Настройка показателей производительности
- Hint! Показатели производительности показывают «относительную картину»
 - Например, в одной и тоже ИБ значения для файлового и клиент-серверного варианта могут различаться
 - Поэтому рекомендуется выполнять тестирование в «боевых» условиях
 - При необходимости их нужно эмулировать

Свод правил по разработке форм

- Минимизация вызовов между клиентом и сервером
 - Получение всех необходимых данных за одно обращение к серверу
 - Другой подход к разработке – с точки зрения взаимодействия клиента и сервера
- Все тяжелые алгоритмы – на сервер
 - Простые алгоритмы нужно выполнять на клиенте!!
- Избегать неявных вызовов сервера
 - Предопределенное Значение
- Мысленно можно представить вызов сервера как последовательность действий
 - А также обратная передача на клиент
- Стараться использовать внеконтекстные серверные вызовы

Сравнение эффективности алгоритмов

- Задача
 - В форме документа при нажатии на кнопку «Заполнить» подставлять значение контрагента из константы, а также заполнять реквизит формы «Статус клиента»
- Варианты реализации
 - 2 внеконтекстных серверных вызова
 - 1 контекстный серверный вызов
 - 1 внеконтекстный серверный вызов
 - Оптимизация – выбор ссылки и статуса запросом

Использование контекстных вызовов

- Несколько вариантов применения
 - Необходимость получения «настоящего» объекта
 - Преобразование из данных формы
 - Вызов методов объекта, в т.ч. экспортных, обращение к свойствам
 - Но! Для записи не нужно получать объект!!
 - Обновить данные табличной части целиком
 - Клиент хранит лишь видимые строки
 - А все данные есть на сервере
 - Пример

Неявные вызовы сервера

- При вызове ряда методов выполняется обращение к серверу
 - Например
 - ОткрытьФорму(), ПолучитьФорму()
 - ДанныеФормыКоллекция.НайтиСтроки()
- Обращение к некоторым свойствам требует вызов сервера
 - Например, Свойство «Доступность» поля формы
- Изменение некоторых свойств генерирует вызов сервера
 - Например, свойство «Видимость» поля формы

Работа с данными списков

- При работе с динамическими списками и табличными частями следует учитывать, что на клиенте отображаются не все данные
 - Читаются видимые данные
 - Первое открытие 42 строки
 - Дочитывание – видимые данные + 2 строки
 - При перемещении по списку данные «дочитываются»
- Примеры
 - Пересчет строк в табличной части
 - Изменение суммы
 - $27 + (10\ 918+XXX) + 197\ 357$
 - $1 + 1\ 644 + 246\ 327$
 - Диалог Изменить
 - Сервисная обработка

Использование временного хранилища

- Идея
 - Если требуется передача большого количества строк между разными формами, то зачем передавать их на клиента?
- Временное хранилище
 - Время жизни
 - Поместить В Временное Хранилище
 - Получить Из Временного Хранилища

Глава 19. Сообщение пользователю

Свойства, примеры использования...

СообщениеПользователю

- Свойства
 - Текст
 - Поле
 - Точный путь до реквизита
 - Или относительный путь до реквизита объекта
 - Путь к данным
 - Указание реквизита объекта
 - Ключ данных
 - Указание формы, к которой привязано сообщение
 - Эффект гиперссылки
- Методы
 - Сообщить

Примеры вывода сообщений

- Пример №1
 - Вывод сообщения на клиенте, привязанного к реквизиту формы
- Пример №2
 - Вывод сообщения на клиенте, привязанного к реквизиту объекта

Привязка вне формы

- Как быть если сообщение генерируется вне контекста формы?
- Метод УстановитьДанные
 - Выполняет соответствие объекта и данных формы
 - Установка свойств ПутьКДанным и КлючДанных
- Варианты соответствия
 - Автоматическое
 - Система ориентируется на типы данных
 - «Ручное»
 - УстановитьСоответствиеОбъектаИРеквизитаФормы

Глава 20. Примеры работы с управляемыми формами

Решение практических задач...

Задача #1

- Вывод связанных списков
- Задача
 - В форме списка справочника «Контрагенты» выводить данные о контактных лицах текущего контрагента
- 3 варианта решения
 - Использование отборов динамического списка
 - Условие в запросе динамического списка
 - Использование команд формы

Задача #2

- Создание АРМ кладовщика
- Задача
 - Создать форму в которой отображаются
 - Список документов поступления и реализации по выбранному контрагенту
 - Возможность просмотра и редактирования текущего документа в непосредственно в форме
 - Возможность создания новых документов
 - Просмотр движений текущего документа в форме обработки

Задача #3

- Форма выбора, возврат множества данных
- Задача
 - В документе «Поступление товаров» при выборе товара выводить информацию о текущих ценах
 - При выборе подставлять не только значение номенклатуры, но и цены отображаемой в списке
- Решение
 - Вариант 1
 - Событие ПриИзменении, выполнять дополнительный запрос
 - Вариант 2
 - Оповещение родительской формы о выборе
 - Метод ОповеститьОВыборе()