# Подготовка программистов 1С (Павел Чистов)

# 1/30

1С Предприятие – Среда для разработки бизнес приложений, с предопределенными классами.

Информационная база состоит из:

* Базы данных
* Конфигурация (описание структуры)

Информационная база

БД

Конфигурация

Физически, система может исполняться в двух архитектурах:

Режим файл-сервер – информационная база хранится в файлике 1cv8.1CD, плюсы/минусы:

* Дешево
* Плохая реализация много клиентской работы

Режим клиент сервер

Клиент

1С сервер

Web

SQL

ИБ

Основная задача: получение отчетности.

Регистры

Отчеты

Документы

Регистры – объекты хранящие показатели учета. Документы туда пишут данных, а отчеты на их основе анализируют их.

# Создание базы

Настройка → отображать в виде дерева; выбрал корень дерева → добавить → создание новой группы. Выбрал папку → добавить → создание новой информационной базы → создание пустой базы без конфигурации → настроил папку → далее выбрал тонкий клиент → готово.

Для создание конфигурации нужно нажать: конфигурация → открыть конфигурацию. При этом в базу запишется копия конфигурации разработчика. Для сохранения обновления конфигурации нужно нажать: конфигурация → обновить конфигурацию базы данных.

Окно «конфигурация» состоит из прикладных объектов конфигурации.

# Класс объектов (справочник)

Справочник → добавить → Имя:Сотрудники → Представление объекта:Сотрудник → Представление списка:Список сотрудников; для добавления новых реквизитов: данные → добавить реквизит.

Для запуска нажал: отладка → начать отладку (f5).

Далее создал справочник «Контрагенты» и в закладке Иерархия, поставил галочку, после чего, добавил реквизит ОсновнойМенеджер, в типе указал СправочникСсылка.Сотрудники

# 2/30

Модуль объекта (логика при событии)

Объект Справочник Сотрудники

Макеты (вывод)

Реквизиты

Модуль формы (интерактивные события)

Формы

Создание формы

Формы → элемента →лупа → готово. Далее для формы выбрал событие ПередЗаписью и нажал на кнопку лупы для добавления кода в модуль.

Даты:

$ Сегодня = ‘20170917191545’;

$ Сегодня = ‘2017-09-17 19:15:45’;

$ Сегодня = ‘2017-09-17’ // ‘00:00:00’

$ ПустаяДата = ‘00010101’;

**&НаКлиенте**

**Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)**

**Если Объект.Работающий И Объект.ДатаРождения = ‘00010101’ Тогда**

**Сообщить(“Не заполненена дата рождения”);**

**Отказ = Истина; //Отмена проводки перед записью**

**КонецЕсли;**

**КонецПроцедуры**

Проверка модуля (синтаксический контроль) Ctr+F7

Пользователю отобразить информацию при старте программы:

Конфигурация → открыть модуль управляемого приложения; события, ожидаемые к описанию в этом модуле можно посмотреть нажав на кнопку proc (лупа).

Предупреждение! В версии 8.3 модальные окна отключены. Для исправления: правой кнопкой по Конфигурация → Свойства, Режим использования модальности → Использовать.

# 3/30

Справочники – объекты, которые являются записями аналитического учета. Реализация в БД:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID(Ссылка) | Код | Наименование | Родитель | ЭтоГруппа | → |
| 000АА | 0001 | Покупатели |  | Истина | → |
| 000AB | 0002 | Шилов И.И. | 000AAA | Ложь | → |

Группы справочника не являются элементами аналитики, используется только для группировки объектов аналитического учета.

|  |  |  |  |
| --- | --- | --- | --- |
| → | ПометкаУдаления | Предопределенный | Владелец |
| → | Ложь | Ложь | ДругойСправочник |
| → | Ложь | Ложь | ДругойСправочник |

Создание предопределенного элемента:

При создании справочника → прочее → предопределенные. Для задания Владельца справочника → владельцы.

Создание нового справочника

Создал справочник Номенклатура (Товар,Услуга); включил иерархию; добавил реквизиты: Услуга (булево), ЕдИмз (СправочникСсылка.ЕдиницыИзмерения), ЦенаПродажи (Число, 15, 2). Для значения штук в ЕдИзм по умолчанию, использовал свойство Представление → значение заполнения.

Константы существуют для хранения редко изменяющейся информации, например – название организации.

# 4/30

Для того, чтобы при создании услуги в справочнике Номенклатура, в группе Услуги, автоматически проставлялась галочка, что это услуга. Сперва для реквизита Услуга в свойствах Использование, нужно установить Для группы и элемента.

*\*Null от Неопределенно отличается тем, что в ячейке с Null в принципе нельзя записать значение*

Номенклатура → Прочее → Модуль объекта → proc → ОбработкаЗаполнения. Один из аргументов – ДанныеЗаполнения – это структура, хранящая родителя создаваемого элемента.

Структура

|  |  |
| --- | --- |
| Ключ | Значение |
| “Родитель” | Сопр. Услуги |

В свойствах Услуга → Представление → Заполнять из данных заполнения.

Код модуля справочника Номенклатура:

**Процедура ОбработкаЗаполнения(**

**ДанныеЗаполнения,**

**ТекстЗаполнения,**

**СтандартнаяОбработка**

**)**

**Если ДанныеЗаполнения = Неопределено Тогда**

**Возврат**

**КонецЕсли;**

**Если ДанныеЗаполнения.Свойство(“Родитель”) Тогда**

**//Услуга = ДанныеЗаполнения.Родитель.Услуга;**

**ДанныеЗаполнения.Вставить("Услуга",**

**ДанныеЗаполнения.Родитель.Услуга);**

**КонецЕсли;**

**//СтандартнаяОбработка = Ложь;**

**КонецПроцедуры**

Для проверки заполнения (флажков и т.п.) Пишем обработку в модуле объекта (на стороне сервера). *Форму желательно нагружать меньше всего.* В модуле выбрал proc ОбработкаПроверкиЗаполнения(Отказ, ПроверРеквез). Отказ – признак успешности транзакции.

Сперва нужно установить флажок ЕдИзм → Представление → Проверка заполнения. Далее код:

**Процедура ОбработкаПроверкиЗаполнения(Отказ, ПроверяемыеРеквизиты)**

**Если Не ЭтоГруппа И Услуга Тогда**

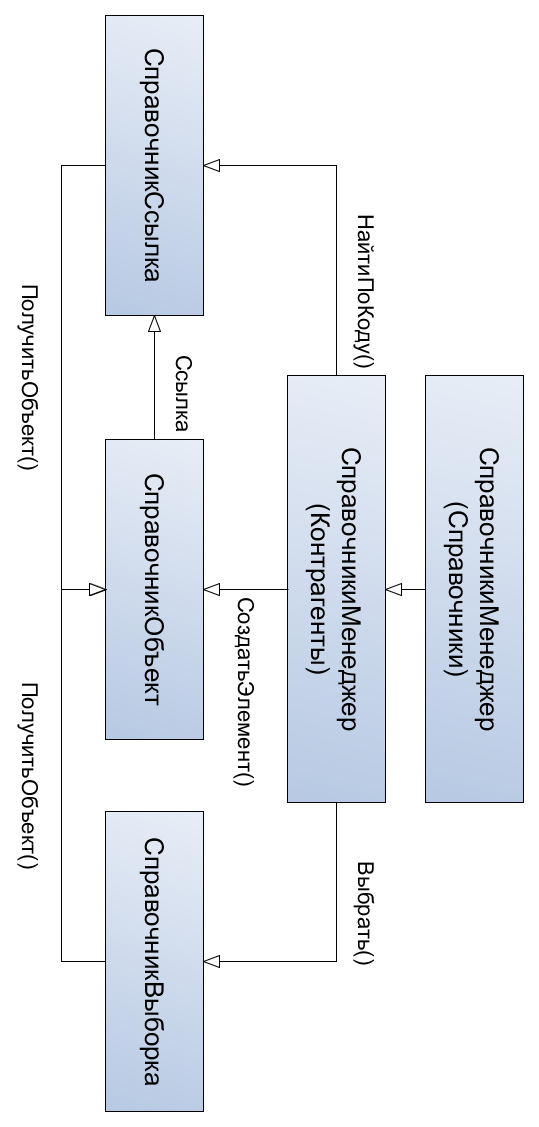
**ПроверяемыеРеквизиты.Удалить(ПроверяемыеРеквизиты.Найти(“ЕдИмз”));**

**КонецЕсли;**

**КонецПроцедуры**

Для написания функции, предназначенной для выполнения на сервере используется репозиторий общих модулей с преопределенными процедурами: Общие **→** общие модули.

Для возврата списка создал модуль ОбщиеМеханизмы.

****

**Функция ПолучитьСписокИменниников() Экспорт**

**ТД\_День = День(ТекущаяДата());**

**ТД\_Месяц = Месяц(ТекущаяДата());**

**Массив = Новый Массив();**

**Выборка = Справочник.Сотрудники.Выбрать();**

**Пока Выборка.Следующий() Цикл**

**ДР\_День = День(Выборка.ДатаРождения);**

**ДР\_Месяц = День(Выброка.ДатаРождения);**

**Если ДР\_Месяц = ТД\_Месяц И ДР\_День = ТД\_День Тогда**

**Массив.Добавить(Выборка.Наименование);**

**КонецЕсли;**

**КонецЦикла;**

**Возврат Массив;**

**КонецФункции**

В коде общего модуля

**Процедура ПриНачалеРаботыСистемы()**

**СписокИменниников = ОбщиеМеханизмы.ПолучитьСписокИменниников();**

**Для каждого ЭлементаМассива из СписокИменниников Цикл**

**Сообщить(“Сегодня ДР у ” + ЭлементМассива);**

**КонецЦикла;**

**КонецПроцедуры**

# 5 / 30

Для того чтобы не возникало ошибок, при просмотре элементов и групп в запросе, можно использовать следующую команду:

**Если Выборка.ЭтоГруппа Тогда**

**Продолжить;**

**КонецЕсли;**

При просмотре элементов в группах, для того что бы не отображать непомеченных:

Свойства справочник → формы → форма элемента → далее → готово.

Выбираю поле ОсновнойМенеджер → События:ПриИзменении → лупа. Далее на форме мы получаем строку (представление объекта) и ссылку на него.

В модуле формы:

**&НаКлиенте**

**Процедура ОсновнойМенеджерПриИзменении(Элемент)**

**Результат = ПроверитьРаботает(Объект.ОсновнойМенеджер)**

**Если Не Результат Тогда**

**Сообщить(«Менеджер не работает»);**

**КонецЕсли;**

**КонецПроцедуры**

**&НаСервереБезКонтекста**

**Функция ПроверитьРаботает(Менеджер)**

**Возврат Менеджер.Работающий;**

**КонецФункции**

Контекст работы функций модуля

**&НаСервере**

В таком случае, все данные формы передаются обратно на сервер, в ожидании того, что было сделано какое-то изменение.

**&НаСервереБезКонтекста**

В данном случае нельзя обратиться к свойствам глобального Объекта. Для доступа к нужным данным, объект нужно передать в качестве аргумента.

Создание отчета

Создал отчет ПрайсЛист → Создал форму отчета → создал реквизит ТабДок:ТабличныйДокумент, (положение заголовка – нет). Далее добавил команду СформироватьПрайс → действие → лупа …

Создал Макет с типом Табличный документ. Далее добавил текст. Выделил текст под переменные и в свойствах → Макет → Макет → Заполнение:Параметр. Для заголовка выбрал шаблон (смесь текста и параметра). Для обозначения элементов: Выделил (по полю) → имена → назначить имя. Далее перешел в модуль команды и написал код:

**&НаКлиенте**

**Процедура СформироватьПрайс(Команда)**

**ЗаполнитьТабДок(ТабДок);**

**КонецПроцедуры**

**&НаСервереБезКонтекста**

**Процедура ЗаполнитьТабДок(ТабДок)**

**ТабДок.Очистить();**

**Макет = Отчеты.ПрайсЛист.ПолучитьМакет("Макет");**

**ОблШапка = Макет.ПолучитьОбласть("Шапка");**

**ОблЭлемент = Макет.ПолучитьОбласть("Элемент");**

**ОблШапка.Параметры.ДатаОтчета = Формат(ТекущаяДата(), "ДЛФ=DD");**

**ТабДок.Вывести(ОблШапка);**

**Выборка = Справочники.Номенклатура.ВыбратьИерархически();**

**Пока Выборка.Следующий() Цикл**

**ОблЭлемент.Параметры.Заполнить(Выборка);**

**ТабДок.Вывести(ОблЭлемент);**

**КонецЦикла;**

**КонецПроцедуры**

Для добавления форматной строки в аргумент функции Формат: правой кнопкой мыши → Конструктор форматной строки.

# 6 / 30

На форме можно вывести чекбокс о выводе только услуг. Для это в модуле добавил проверку:

**Если Не Выборка.Услуга И ПечататьТолькоТовары Тогда**

**Продолжить;**

**КонецЕсли;**

Для выделения жирным на макете групп, сделал именованные области на макете, и добавил код в модуль:

**ОблГруппа = Макет.ПолучитьОбласть("Группа");**

**Если Выборка.ЭтоГруппа Тогда**

**Обл = ОблГруппа;**

**Иначе**

**Обл = ОблЭлемент;**

**КонецЕсли;**

## Регистры сведений

Нужны для того, чтобы сохранять историю данных и затем запрашивать срез информации за определенный период. Существуют четыре типа:

* Ресурс (что храним)
* Измерение (в разрезе чего храним)
* Изменение значения ресурса

|  |  |  |
| --- | --- | --- |
| Дата | Валюта | Курс |

Регистры сведений -- создать (КурсыВалют), Периодичность: В пределах дня, Режим записи: (Независимый), Основной отбор по периоду (включает период в индекс, что делает срез по времени быстрее).

Реквизиты – можно использовать как комментарии к регистру.

Добавить измерение (Валюта) -- Тип: (СправочникСсылка.Валюты), CheckBox: Ведущее (при удалении из справочника, все удаляется из регистра), Основной отбор, Запрет незаполненных. Создал Ресурс: Курс (число, 4 знака точности). Далее, для получения среза в общем модуле «ОбщиеМеханизмы» (Свойство: Вызов на сервере) написал код:

**Функция ПолучитьКурсВалюты(Валюта, Знач Дата = Неопределено) Экспорт**

**Дата = ?(Дата=Неопределено, ТекущаяДата(), Дата);**

**Структура = Новый Структура;**

**Структура.Вставить("Валюта", Валюта);**

**Данные = РегистрыСведений.КурсыВалют.ПолучитьПоследнее(**

**Дата, Структура);**

**Возврат Данные.Курс;**

**КонецФункции**

Для проверки, добавил реквизит (ВалютаВзаиморасчета: СсылкаВалюта) в справочник Контрагенты. Открыл форму справочника и из Объекта перетащил ВалютуВзаиморасчета на форму + создал реквизит Курс (Вид: Поле надписи) с типом Число. Создал события при создании формы и при изменении ФормыВалюты:

**&НаСервере**

**Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)**

**ОбновитьКурс(Объект.ВалютаВзаиморасчета, Курс);**

**КонецПроцедуры**

**&НаКлиенте**

**Процедура ВалютаВзаиморасчетаПриИзменении(Элемент)**

**ОбновитьКурс(Объект.ВалютаВзаиморасчета, Курс);**

**КонецПроцедуры**

**&НаКлиентеНаСервереБезКонтекста**

**Процедура ОбновитьКурс(Валюта, Курс)**

**Если Валюта.Пустая() Тогда**

**Курс = 0;**

**Иначе**

**Курс = ОбщиеМеханизмы.ПолучитьКурсВалюты(Валюта);**

**КонецЕсли**

**КонецПроцедуры**