

**002****Прикладные решения 1с: Предприятие 8.1.**

Решение заданий.

	<b>Прикладные решения 1с: Предприятие 8.1.</b>
	<p><b><u>Вопросы.</u></b></p> <p>1. Создание ИБ. 2. Подсистемы. 3. Справочники. 4.Документы. 5. Формы. 6. Регистры накопления.7.Отчет 8.Макеты. Редактирование макетов и форм. 9.Периодический регистр сведений.      10.Перечисления.11.Проведение документа по нескольким регистрам.      12.Оборотные регистры накопления.13.Отчеты.      14.Оптимизация проведения документа «Оказание услуги».      15.План видов характеристик. 16.Бухгалтерский учет.      17.План видов расчета. 18.Использование регистра расчета.      19.Поиск в базе данных. 20.Выполнение заданий по расписанию. 21.Редактирование движений в форме документа.      22.Список пользователей и их роли.      23.Рабочий стол и настройка командного интерфейса.      24.Обмен данными. 25.Функциональные опции.      26.Подборы и ввод на основании. 27.Приемы разработки форм.</p>
1	<p><b><u>Место БД.</u></b></p> <p>-Сервис-Шаблоны –Действия –Настройка шаблонов –Добавить      -Сервис- Параметры _Тексты модулей –Автозамена (ctrl+Q)      C:\Users\w7128\Documents\InfoBase - БД      C:\Users\w7128\AppData\Roaming\1C\1Cv82\tmplts</p>
2	<p><b><u>САЙТ 1С – 8.2..</u></b></p> <p><a href="http://www.1c.ru">http://www.1c.ru</a>  <a href="http://users.v8.1c.ru">http://users.v8.1c.ru</a> - сайт поддержки пользователей системы 1С:Предприятие 8  <a href="http://v8.1c.ru/AllInfo">http://v8.1c.ru/AllInfo</a>  <a href="http://v8.1c.ru">http://v8.1c.ru</a> - Сайт по технологической платформе системы 1С:Предприятие 8  <a href="http://www.1c.ru/rus/partners/service.jsp">http://www.1c.ru/rus/partners/service.jsp</a> - оформление подписки на ИТС  <a href="http://devtrainingforum.v8.1c.ru/forum/">http://devtrainingforum.v8.1c.ru/forum/</a> - Конференция для обучения программированию</p>
3	
4	<p><b>01.Постановка задачи 1с:8.1 firma_MASTERSKAJA_REMONTA_VMA</b></p> <p>На примере прикладного решения показана структура объектов конфигурации, их назначение и использование.</p> <p>Здесь представлен пример разработки прикладного решения для небольшой фирмы, оказывающей бытовые услуги.</p> <p>Здесь представлены разнообразные услуги, оказываемые клиентам, снабжение фирмы необходимыми материалами и их расход при оказании услуг.</p> <p>Работа предприятия позволяет рассмотреть учет персонала и заработной платы сотрудников.</p> <p>Разнообразие видов деятельности позволяет показать возможности формирования различных отчетов и итоговых данных на основе имеющейся информации.</p>
5	<p><b>02. Решаемые вопросы. Часть 2 - firma_MASTERSKAJA_REMONTA_VMA</b></p> <p>Проведение – изменение Документа ОказаниеУслуги.      Проведение документа по нескольким регистрам.      Оборотные регистры накопления.      Отчеты.      Отчет. Элементы пользовательских настроек.      Отчет за все дни.</p>
6	

	<p>Отчет по выручке - диаграмма      Отчет – получение актуальных занчений из периодического регистра сведений.      Отчет - Вывод данных в таблицу.      План видов характеристик.      Главное меню – Все функции.      Приходная накладная с характеристиками номенклатуры.      Отчет с характеристиками номенклатуры.      План видов характеристик. Виды Субконто. План счетов. Регистр бухгалтерии.      Отчет для бухгалтерии предприятия.      План видов расчета, регистр расчета.      Использование регистра Расчета.      Отчет по перерасчетам.      Вытеснение по периоду действия.</p>
7	<p><b>03. Решаемые вопросы. Часть 3 - firma_MASTERSKAJA_REMONTA_VMA</b></p> <p>Полнотекстовый поиск      Выполнение заданий по расписанию      Редактирование движений в форме документа      Список пользователей и их роли      Рабочий стол и настройка командного интерфейса      Настройка рабочего стола      Функциональные опции      Подборы и ввод на основании      Команда ввода на основании      Приемы разработки форм - оформление      Вычисляемые поля      Список выбора для поля ввода      Форма выбора для поля, содержащего ссылочный реквизит      Проверка заполнения реквизитов</p>
8	<p><b>04. Вопросы. 1с8.1. Практическое Пособие Разработчика</b></p>
9	<p>1 Часть 1. Быстрая разработка прикладных решений      2 1 Глава 3. Документ      4 3.1 Процедура МатериалыКоличество ПриИзменении      7 3.2 Процедура РассчитатьСумму()      10 3.3 Процедура МатериалыКоличество ПриИзменении()      11 Работа СДокументами.РассчитатьСумму(Строка ТабличнойЧасти)      13 3.4 Процедура Материалы Цена ПриИзменении()      16 2 Глава 7. Редактирование макетов и форм      17 7.1 Печать формы документа      20 Часть 2. Использование основных объектов конфигурации      21 3 Глава 1. Периодический регистр сведений      23 1.1 Функция РозничнаяЦена()      26 1.2 Процедура ПереченьНоменклатурыНоменклатура ПриИзменении      29 4 Глава 2. Перечисления      30 2.1 Движения документа Оказание Услуги      33 5 Глава 4. Оборотный регистр накопления      34 4.1 Добавление цикла обхода табличной части и записи движений регистра "Продажи"      37 4.2 Команды создания движений регистра "Продажи"      40 6 Глава 5. Система компоновки данных      41 5.9 Выражение для расчета параметра КонецПериода      44 5.17 Объявление переменной      47 5.18 Обработчик события формы отчета При открытии      50 5.19 Обработчик поля выбора При изменении      53 7 Глава 6. Оптимизация процедуры проведения документа Оказание Услуги      55 6.3 Заготовка процедуры проведения      58 6.1 Задание параметров запроса      61 6.6.2 Задание параметров запроса -2      64 6.7 Получение результата запроса и цикл его обхода</p>

67 6.8 Проверка на складе достаточного количества номенклатуры  
 70 6.9 Формирование движений по регистрам накопления  
 73 6.10 Запись движений регистров  
 76 8 Глава 7. План видов характеристик  
 77 7.1 Процедура СвойстваВидСвойстваПриИзменении  
 80 7.2 Процедура ВариантыПриАктивизацииСтроки  
 81 ЭлементыФормы.Свойства.Значение.Отбор.НаборСвойств.Установить  
     (Элемент.ТекущиеДанные.Ссылка, Истина);  
 83 7.3 Процедура ПриОткрытии  
 84 ЭлементыФормы.Свойства.Значение.Отбор.НаборСвойств.Установить( , Истина)  
 86 7.4 Процедура СвойстваВидСвойстваПриИзменении  
 89 7.5 ПриИзменениеСсылки()  
 92 7.6 Вызов обработчика ПриИзменениеСсылки  
 95 7.7 Процедура СвойстваПередНачаломДобавления()  
 98 7.8 Процедура ОбработкаПроведения()  
 99 Движение.НаборСвойств = ТекСтроКаМатериалы.НаборСвойств  
 10 9 Глава 8. Бухгалтерский учёт  
 102 8.5 Движения по регистру бухгалтерии Управленческий  
 105 10 Глава 10. Использование регистра расчета  
 106 10.2 Обработчик проведения документа НачисленияСотрудникам  
 109 10.3 Заготовка процедуры РасчитатьНачисления  
 112 10.4 Изменение процедуры РасчитатьНачисления  
 115 10.5 Добавление обхода набора записей и расчета записей  
 118 10.6 Добавление текста запроса во вторую строку условия  
 121 10.7 Добавление обхода набора записей регистра и вычисления результата вторичных записей  
 124 10.8 Обработчик события нажатия кнопки  
 3 10.9 Процедура перерасчета начислений  
 6 10.10 Обработчик события Нажатие кнопки  
 9 10.11 Изменение обработчика события Нажатие кнопки  
 12 11 Глава 11. Поиск в базе данных  
 13 11.1 Определение переменной СписокПоиска  
 16 11.2 Обработчик события ПриОткрытии формы  
 19 11.3 Обработчик нажатия кнопки Найти  
 22 11.4 Процедура ВывестиРезультатПоиска()  
 25 11.5 Процедура ДоступностьКнопок()  
 28 11.6.1 Обработчик нажатия кнопки ПредыдущаяПорция  
 31 11.6.2 Обработчик нажатия кнопки СледующаяПорция  
 34 11.7 Обработчик события onClick поля HTML-документа  
 37 12 Глава 12. Выполнение заданий по расписанию  
 38 12.1 Процедура обновления индекса  
 41 12.2 Процедура слияния индексов  
 44 12.3 Обработчик ожидания  
 47 12.4 Сообщение о запуске регламентного задания  
 48 Сообщить("Запуск регламентного задания Обновление индекса " + ТекущаяДата())  
 49 15 Глава 15. Обмен данными  
 50 15.1 Функция формирования префикса номера  
 53 15.2 Обработчик события ПриУстановкеНовогоКода  
 56 15.3 Обработчик события ПриУстановкеНовогоНомера  
 59 15.4 Создание служебной переменной  
 62 15.5 Обработчик события формы ПередЗаписью  
 65 15.6 Обработчик события формы ПриЗаписи  
 68 15.7 Обработчик события формы ПередОткрытием  
 71 15.8 Создание процедуры ОбменСФилиалами  
 74 15.9 Обработчик события нажатия кнопки Выполнить  
 77 15.10 Формирование имени файла в процедуре записи данных  
 80 15.11 Создание объекта записи XML в процедуре записи данных  
 83 15.12 Создание очередного номера сообщения и запись заголовка сообщения в XML  
 86 15.13 Получение выборки из записей регистрации изменений, предназначенных данному узлу  
 89 15.14 Перебор выборки записей и сериализация их в открытый XML файл  
 92 15.15 Формирование имени файла, содержащего данные обмена  
 95 15.16 Добавление чтения найденного файла с данными обмена  
 98 15.17 Добавление чтения заголовка XML-сообщения

	<p>10 1 15.18 Добавление проверки сообщения</p> <p>104 15.19 Удаление записей регистрации изменений для узла отправителя сообщения</p> <p>107 15.20 Чтение данных из сообщения</p> <p>110 15.21 Представление данных XML в виде значения, имеющего тип 1С:Предприятия</p> <p>113 15.22 Разрешение возможных коллизий</p> <p>116 15.23 Запись полученных данных</p> <p>119 15.24 Обработчик нажатия кнопки Создать начальный образ</p> <p>2 15.25 Обработчик нажатия кнопки Записать изменения</p> <p>515.26 Обработчик нажатия кнопки Прочитать изменения</p> <p>815.27 Просмотр работы событий объекта ПланОбменаОбъект</p> <p>1115.28 Перемещение Узла2 в корень дерева</p> <p>1415.29 Отключение поддерева от дерева</p> <p>1715.30 Создание распределенной информационной базы из баз с идентичной конфигурацией</p> <p>2017 Глава 17. Создание документа ввода начальных остатков</p> <p>2117.1 Обработчик события Перед записью формы документа</p> <p>2417.2 Обработчик события Перед записью в модуле объекта Документ</p> <p>2718 Часть 3. Специальные приемы разработки</p> <p>28 Глава 1. Общие приемы</p> <p>301.1 Обработчик события нажатия кнопки Подбор</p> <p>331.2 Обработчик события Обработка выбора табличного поля Материалы</p> <p>361.3 Обработчик события нажатия кнопки Подбор</p> <p>391.4 Изменения в обработчике события нажатия кнопки Подбор</p> <p>421.5 Изменения в обработчике события Обработка выбора</p> <p>451.7 Обработчик нажатия кнопки ДокументыКлиента</p> <p>481.8 Изменения обработчика события При открытии формы документа</p> <p>5118 Глава 2. Общие формы</p> <p>532.1 Текст обработчика события При активизации строки</p> <p>562.2 Обработчик события При открытии формы</p> <p>592.3 Обработка события При изменении для кнопки Остатки материалов</p> <p>622.4 Обработка события При открытии формы документа</p> <p>652.5 Обработка события При изменении переключателя</p> <p>682.6 Обработчик события При активизации строки табличного поля</p> <p>712.7 Обработчик события При выводе строки табличного поля</p> <p>742.8 Обработчик события При открытии формы</p> <p>772.9 Изменения в обработчике события При выводе строки табличного поля списка справочника</p>
1 0	
1 1	
1 2	
1 3	
1 4	
1 5	<p><b>003-разы... ПЛАН.полный 1-28.</b></p> <ol style="list-style-type: none"> <li>1. Создание ИБ. 2. Подсистемы. – картинки, +сортировка-командный интерфейс + правая-свойства +Подсистемы-все подсистемы +ВИДИМОСТЬ ++Панель навигации ++действий ++рабочая область</li> <li>2. –Справочники-данные реквизиты –изменить</li> <li>3. –КонтекстМодуляФормы = + лококонтекст,+ рекв.формы, +св и мет Управл форма + св и мет расш форм +глоб конт +экспорт перем</li> <li>4. <b>Форма_документ ++АНАЛИЗ_КОДА</b> = +1модуль формы +2реквизиты +3УправляемаяФорма_свойство +4Расширение управляемой формы для документа_свойство +5Глоб контекст_свойство ++6 имя неглоб общ модуля ++ не проц глоб мод ++не перемен управ прил ==== лок перем модуля</li> </ol> <p><b>АНАЛИЗ_КОДА 2 = Элементы.Материалы.ТекущиеДанные =ctrl+f1 + == ctrl+f</b> + Сервис-Параметры-Тонкий клиент</p> <p><b>АНАЛИЗ_КОДА 3 = отладчик (при написании своей программы) == значки +</b> ++f5 + Элементы.Материалы.ТекущиеДанные; + выделить + Элементы + _shift+f9 (вычислить выражение) _кнопка F2 ( в отдельном окне)</p>
1 6	

++стек вызовов + шагнуть через F10 +  
 +++ ЭтаФорма \_рассчитать или ЭтотОбъект

5.

6. Регистры накопления.- измерения, ресурсы.

Регистры накопления + Документы ++ командный интерфейс \_форма  
 +++ регистратор\_документ +++РН \_форма\_команд\_интерфейс  
 +++++КОЛЛЕКЦИЯ

++ Справочник.Сотрудники ++ Справочник["Сотрудники"] ++Табличнаячасть[0]

7. Отчет. +Макет +Запрос

+Настройки-Отчет-Детальные записи (ins)

8. Макеты. Редактирование макетов и форм.

+ОказаниеУслуги +Макеты +Печать =Создание

+ОказаниеУслуги +Макеты +Печать =Редактирование

+ОказаниеУслуги +Прочее+МодульМенеджера =код

ОбластьИтог= Макет.ПолучитьОбласть("Всего");

СуммаИтог=0;

СуммаИтог= СуммаИтог+ВыборкаПереченьНоменклатуры.Сумма;

ОбластьИтог.Параметры.ВсегоПодокументу=СуммаИтог;

ТабДок.Вывести(ОбластьИтог);

9.Периодический регистр сведений. Чтение-запись на сервере.

+

Функция РозничнаяЦена(АктуальнаяДата,ЭлементНоменклатуры)Экспорт

Отбор=Новый Структура("Номенклатура",ЭлементНоменклатуры);

ЗначенияРесурсов=РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,Отбор);

Возврат ЗначенияРесурсов.Цена;

КонецФункции

+

Процедура ПереченьНоменклатурыНоменклатураПриИзменении(Элемент)

// Вставить содержимое обработчика.

//Получить текущую строку

СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;

//Установить цену

СтрокаТабличнойЧасти.Цена=РаботаССправочниками.РозничнаяЦена(Объект.Дата,СтрокаТабличнойЧасти.Номенклатура);

//Пересчитать сумму строки

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

10.Перечисления.

+регистрция расхода ном= материал

+ ОказаниеУслуги

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры=

Перечисления.ВидыНоменклатуры.Материал Тогда

КонецЕсли;

11.Проведение документа по нескольким регистрам. РЕГИСТР ОСТАТКОВ.

Регистр СтоимостьМатериалов

+,, ПриходнаяНакладная

Движения.СтоимостьМатериалов.Записывать=Истина;

// регистр СтоимостьМатериалов приход

Движение=Движения.СтоимостьМатериалов.Добавить();

Движение.ВидДвижения=ВидДвиженияНакопления.Приход;

Движение.Период=Дата;

Движение.Материал=ТекСтрокаМатериалы.Материал;

Движение.Стоимость=ТекСтрокаМатериалы.Сумма;

++ Команд. Интерфейс формы

+++ Оказание Услуги ...

12.Оборотные регистры накопления. Продажи.(ОБОРОТЫ) Услуги = РЕГИСТР ОБОРОТОВ.

```

+++ создание регистра и для услуг и для товаров
+++ ком. Инт. = все подсистемы_видимость_см.также
+ ОказаниеУслуги Движения вручную
+
КонецЕсли;
// РегистрПродажи
    Движение=Движения.Продажи.Добавить();
    Движение.Период=Дата;
    Движение.Номенклатура=ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Клиент=Клиент;
    Движение.Мастер=Мастер;
    Движение.Количество=ТекСтрокаПереченьНоменклатуры.Количество;
    Движение.Выручка=ТекСтрокаПереченьНоменклатуры.Сумма;
    Движение.Стоимость=ТекСтрокаПереченьНоменклатуры.Стоимость*
        ТекСтрокаПереченьНоменклатуры.Количество;

```

### 13.Отчеты.

2 способа доступа к данным БД – объектный RW и табличный R

+ Объектный = ДокументОбъект.ОказаниеУслуги (обр. событ... целостность)

+ Табличный - с пом. Запросов (больш об инф.) = реальн табл (обък +необъект) + вирт.

++Объект. Таб - поле Ссылка (на тек запись) +польз. Представл.

+++Язык запросов

+++Система комп. Данных = схема комп данн +

+++ СиКД = +СКД+КомпновщикМакета+ПроцессорКомпановки+РезультатКомпановки

### ++1 Материалы\_01

+

ВЫБРАТЬ

ОстаткиМатериаловОстаткиИОбороты.Склад,  
 ОстаткиМатериаловОстаткиИОбороты.Материал,  
 ОстаткиМатериаловОстаткиИОбороты.КоличествоНачальныйОстаток,  
 ОстаткиМатериаловОстаткиИОбороты.КоличествоПриход,  
 ОстаткиМатериаловОстаткиИОбороты.КоличествоРасход,  
 ОстаткиМатериаловОстаткиИОбороты.КоличествоКонечныйОстаток

ИЗ

РегистрНакопления.ОстаткиМатериалов.ОстаткиИОбороты КАК  
 ОстаткиМатериаловОстаткиИОбороты

### ++2 РеестрДокументовОказаниеУслуги\_02

ВЫБРАТЬ

ОказаниеУслуги.Склад,  
 ОказаниеУслуги.Мастер,  
 ОказаниеУслуги.Клиент,  
 ОказаниеУслуги.Ссылка КАК Документ

ИЗ

Документ.ОказаниеУслуги КАК ОказаниеУслуги

УПОРЯДОЧИТЬ ПО

Документ

+++++1 +Отчет\_Добавить +СКД +запрос+Таблица\_Поля ++Псевдонимы++Порядок  
 ++Настройки\_НоваяГруппировка\_Поля +++Подсистемы

### ++3 РЕЙТИНГ УСЛУГ из 2 табл

ВЫБРАТЬ

СпрНоменклатура.Ссылка КАК Услуга,  
 ПродажиОбороты.ВыручкаОборот КАК Выручка

ИЗ

Справочник.Номенклатура КАК СпрНоменклатура  
 ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты  
 ПО ПродажиОбороты.Номенклатура = СпрНоменклатура.Ссылка

ГДЕ

СпрНоменклатура.ЭтоГруппа = &ЛОЖЬ  
И СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры

УПОРЯДОЧИТЬ ПО

Выручка УБЫВ

++ КонецПериода(&ДатаОкончания,"День")

+++++3.2.1 +Отчет\_Добавить +СКД +запрос+Таблица\_Поля ++Связи

СпрНоменклатура.ЭтоГруппа = ЛОЖЬ

СпрНоменклатура.ВидНоменклатуры = ВидНоменклатуры

++Псевдонимы++Порядок ++ресурсы +параметры

++Настройки\_НоваяГруппировка\_Поля

КонецПериода(&ДатаОкончания,"День")

++ состав даты –тип Дата

ВидНоменклатуры.Услуга

++Выбранные поля + Другие настройки

+++Подсистемы

+++++3.2 +Отчет оформление

+Настройки –Условное оформление –Цвет + Выбранные поля

++Кнопка вверху – Свойства элемента польз. Настроек +Отбор ++

+++++ 4 Выручка мастеров +Отчет\_Добавить +СКД +запрос+ Таблицы ++Параметры\_вирт\_табл (Период- День)

ВЫБРАТЬ

ПродажиОбороты.Мастер,

ПродажиОбороты.Период,

ПродажиОбороты.Клиент,

ПродажиОбороты.ВыручкаОборот КАК Выручка

ИЗ

РегистрНакопления.Продажи.Обороты(, , День, ) КАК ПродажиОбороты

+ Ресурсы +Параметра КонецПериода(&ДатаОкончания,"День")

++ 4.2 Настройки – Детальные записи --Поле группировки –Период

++ 4.3 Диаграмма изм-я +СКД +Вариант отчета + ОбъемВыручки +Диаграмма(стру отчета)

+Точки =мастер +Друг\_настройки =Полосы измерительной диаграммы

+++++ 5 Актуальные значения из периодических регистров сведений.

Перечень услуг +Запрос (Номенкл+ценыСрезПоследних) +&ДатаОтчета

+СВЯЗИ+УСЛОВИЯ+Псевдонимы

+

ВЫБРАТЬ

СпрНоменклатура.Родитель КАК ГруппаУслуг,

СпрНоменклатура.Ссылка КАК Услуга,

ЦеныСрезПоследних.Цена

ИЗ

Справочник.Номенклатура КАК СпрНоменклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних(&ДатаОтчета, ) КАК

ЦеныСрезПоследних

ПО ЦеныСрезПоследних.Номенклатура = СпрНоменклатура.Ссылка

ГДЕ

СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры

++РЕСУРСЫ ++Параметры (+ВидНоменклатуры =услуга)

++Настройки (группировки =групп\_усл\_иерархия+дет записи) +выбр поля

++Другие настройки +Дата отчета\_Быстрый доступ

+++++6 +Отчет\_Добавить +СКД +запрос+ РейтингКлиентов\_об

+

ВЫБРАТЬ

ПродажиОбороты.Клиент,

Из ПродажиОбороты.ВыручкаОборот КАК Выручка,  
 ПродажиОбороты.СтоимостьОборот КАК Стоимость  
 РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

+Выч поля = Доход Выручка-Стоимость  
 ++Ресурсы ++Настройки -диаграмма -точки -Клиент +Выбран поля\_Доход  
 +Вид диаг + назва +++Подсистемы

**+++++7 +Отчет универсальный**

+

**ВЫБРАТЬ**

ПродажиОбороты.Номенклатура,  
 ПродажиОбороты.Клиент,  
 ПродажиОбороты.Мастер,  
 ПродажиОбороты.КоличествоОборот,  
 ПродажиОбороты.ВыручкаОборот,  
 ПродажиОбороты.СтоимостьОборот

Из РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты  
 ++Запрос ++Ресурсы \_поля  
 ++Настройки-Добавить -новая таблица  
 ++Таблица -польз настройки

**14. Оптимизация проведения документа «Оказание услуги».**

Минусы = через точку, опред.стои-ти, контроль остатко ОП.

+ СправочникСсылка.Номенклатура - обращение  
 ++ Необъектн. Данные – в виде записей и полей  
 + ДокументОбъект.ОказаниеУслуги при проведении  
 -- - СправочникВыборка.Клиенты

+++ Док - если инфо из шапки = исп. Конструктора дв.  
 - если инфо табл.+ итоги регистров = исп. Запросы

-- Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры

**+ ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка**

+

//{{КОНСТРУКТОР\_ЗАПРОСА\_С\_ОБРАБОТКОЙ\_РЕЗУЛЬТАТА  
 // Данный фрагмент построен конструктором.  
 // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

Запрос = Новый Запрос;  
 Запрос.Текст =  
 "ВЫБРАТЬ  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК  
 ВидНоменклатуры,  
 | СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК  
 КоличествоВДокументе,  
 | СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,  
 | МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость  
 | Из  
 | | Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК  
 ОказаниеУслугиПереченьНоменклатуры  
 | ГДЕ  
 | | ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка  
 | |  
 | | СГРУППИРОВАТЬ ПО  
 | | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

```

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";
| Запрос.УстановитьПараметр("Ссылка", Ссылка);
| Результат = Запрос.Выполнить();
| ВыборкаДетальныеЗаписи = Результат.Выбрать();
| Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
|   // Вставить обработку выборки ВыборкаДетальныеЗаписи
| КонецЦикла;
//} }КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

```

### **Задача 01.+++ Создание запроса =ОПТИМИЗАЦИЯ ПРОВЕДЕНИЯ ДОКУМЕНТА.**

#### **Постановка задачи 01.**

- 001.Запрос1. Формирование временной таблицы из номенклатуры документа.**
- 002.Запрос2. Стоимость и остатки из регистров накопления. +Блокировка данных.**
- 003. Запись движения регистров. Движения.Записать();**
- 004.Запрос3. Проверка отрицательных остатков.**

+Войти ниже цикла - пр.кнопка \_КонструкторЗапросаСОбработкойРезультата

++Выбрать табл часть –ОказаниеУслуги +выбрать поля

++УСЛОВИЯ \_ССЫЛКА ++ групп.поле \_сумммир.поля (максимум –цена)

++ псевдонимы

++ ТекСтрокаПереченьНоменклатуры зам. На ВыборкаДетальныеЗаписи

### **Задача 02.+++автоматический расчет стоимости =**

запрос= + Табл.часть –оказание услуги + регистр стоим матер +регистр остатки материал.

+1 –удал \_РЕКВИЗИТА \_стоимость из + реквиз + и формы док.

+2 временная таблица \_НоменклатураДокумента (из Дополнительно)

+3 вставка след запроса после Результат = Запрос.Выполнить();

Запрос2 =Новый Запрос;

Запрос2.МенеджерВременныхТаблиц=МенеджерВТ;

Запрос2.Текст="";

+создать Запрос –пр.кн.м. +Таблица +Создание врем.Табл. + 4 поля

+3.2 +1-ый РегистрНакопления.СтоимостьМатериалов.Остатки

+ для производительности = (параметры вирт. Таблицы)

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)

Запрос2.Текст="ВЫБРАТЬ

```

| НоменклатураДокумента.Номенклатура,
| НоменклатураДокумента.ВидНоменклатуры,
| НоменклатураДокумента.КоличествоВДокументе,
| НоменклатураДокумента.СуммаВДокументе,
| СтоимостьМатериаловОстатки.СтоимостьОстаток
| ИЗ
| НоменклатураДокумента КАК НоменклатураДокумента
|   ЛЕВОЕ СОЕДИНЕНИЕ

```

РегистрНакопления.СтоимостьМатериалов.Остатки(

```

|   ,
|   |   Материал В
|   |   (ВЫБРАТЬ
|   |   |   НоменклатураДокумента.Номенклатура
|   |   ИЗ
|   |   |   НоменклатураДокумента)) КАК

```

СтоимостьМатериаловОстатки

```

|   |   ПО НоменклатураДокумента.Номенклатура =

```

СтоимостьМатериаловОстатки.Материал";

3.3 + ОстаткиМатериалов.Остатки рег. ...

3.4 +стои или колич =0 ЕСТЬNULL()

ЕСТЬNULL (СтоимостьМатериаловОстатки.СтоимостьОстаток,0)

**3/5/5 Добавить к полям функцию**

ЕСТЬNULL (СтоимостьМатериаловОстатки.СтоимостьОстаток,0)

**3.5.6 в конце**

Результат =Запрос2.Выполнить();

**3.6 Надо изменить способ получ. стоимости = стои/колич**

Если ВыборкаДетальныеЗаписи.Количество=0 Тогда

СтоимостьМатериала=0;

Иначе СтоимостьМатериала

=ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;

КонецЕсли;

3.6.2

Замена ВыборкаДетальныеЗаписи.Стоимость на СтоимостьМатериала

3.7 При перепроведении документа

// Запишем пустые наборы записей - чтобы читать остатки без учета данных в документе

Движения.СтоимостьМатериалов.Записать();

Движения.ОстаткиМатериалов.Записать();

// перед

Результат =Запрос2.Выполнить();

**3.8 Т3=Результат.Выгрузить(); для проверки явной**

**3.9 Контроль остатков**

Движения.Записать();

Если Режим =РежимПровеенияДокумента.Оперативный Тогда

// Проверить отрицательные остатки

КонецЕсли;

КонецПроцедуры

**3.9.2**

**Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента) И Склад=&Склад**

**ОстаткиМатериаловОстатки.КоличествоОстаток < 0**

**+ к запросу**

Запрос3.УстановитьПараметр("Склад",Склад);

Результат=Запрос3.Выполнить();

ВыборкаДетальныеЗаписи=Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий()Цикл

Сообщение=Новый СообщениеПользователю();

Сообщение.Текст="Не хватает" + Стока(-ВыборкаДетальныеЗаписи.КоличествоОстаток) +  
"единиц материала"""+ВыборкаДетальныеЗаписи.Материал+""";

Сообщение.Сообщить();

Отказ=Истина;

КонецЦикла;

**3.10 Блокировка записей**

// Установим необходимые блокировки данных в регистрах Стои Материалов и ОстаткиМатериалов

Движения.СтоимостьМатериалов.БлокироватьДляИзменения=Истина;

Движения.ОстаткиМатериалов.БлокироватьДляИзменения=Истина;

**15... План видов характеристик.**

**++Объекты – 1. План видов характеристик - СвойстваНоменклатуры**

2. Спр- **Варианты номен.** 3.спр -**Доп.Свойс.Ном.**  
 4. Регистр сведений -**Знач.Свойс.Ном.**

++ Номенклатура+ +ВариантыНоменклатуры\_СПР+ДополнительныеСвойстваНоменклатуры\_СПР  
 +СвойстваНоменклатуры\_ПВХ\_ЗначенияСвойствНоменклатуры\_РСв.

++ изменение формы для владельца

+

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Вставить содержимое обработчика.

Если Параметры.Отбор.Свойство("Владелец") Тогда  
 Элементы.Код.Видимость=Ложь;

КонецЕсли;

КонецПроцедуры

++ в рег свед на форме списка

+

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Вставить содержимое обработчика.

Если Параметры.Отбор.Свойство("НаборСвойств") Тогда  
 Элементы.НаборСвойств.Видимость=Ложь;

КонецЕсли;

КонецПроцедуры

++ **Движение.НаборСвойств=ТекСтрокаМатериалы.НаборСвойств;**

++ в запрос | ОказаниеУслугиПереченьНоменклатуры.НаборСвойств,  
 + |НоменклатураДокумента.НаборСвойств,

## 16. Бухгалтерский учет

+Спр +ПлХар +ПлСч +РеБу

+

Движения.Управленческий.Записывать=Истина;

// Регистр Управленческий

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.Товары;

Движение.СчетКт=ПланыСчетов.Основной.РасчетыСПоставщиками;

Движение.Период=Дата;

Движение.Сумма=ТекСтрокаМатериалы.Сумма;

Движение.КоличествоДт=ТекСтрокаМатериалы.Количество;

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубконто.Материалы]

=ТекСтрокаМатериалы.Материал;

+

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубконто.Материалы]

Движение.СубконтоДт.Материалы то же

## ++ ОказаниеУслуги

+ Движения.Управленческий.Записывать=Истина;

// Регистр Управленческий

// д62 (ДебиторскаяЗадолженность) -К 90 (капитал) РозничнаяСумма

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.ДебиторскаяЗадолженность;

Движение.СчетКт=ПланыСчетов.Основной.Капитал;

Движение.Период=Дата;

Движение.Сумма=ВыборкаДетальныеЗаписи.СуммаВДокументе;

Движение.СубконтоДт.Клиенты=Клиент;

// д90 (капитал) - к 41 (товары) - себестоимость

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.Капитал;

Движение.СчетКт=ПланыСчетов.Основной.Товары;

Движение.Период=Дата;  
 Движение.Сумма=СтоимостьМатериала\*ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
 Движение.КоличествоКт=ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
 Движение.СубконтоКт.Материалы=ВыборкаДетальныеЗаписи.Номенклатура;

#### + Отчет ОСВ

+

#### ВЫБРАТЬ

Основной.Код КАК Код,  
 Основной.Наименование,  
 УправленическийОстаткиИОбороты.СуммаНачальныйОстатокДт КАК СуммаНачДт,  
 УправленическийОстаткиИОбороты.СуммаНачальныйОстатокКт КАК СуммаНачКт,  
 УправленическийОстаткиИОбороты.СуммаОборотДт,  
 УправленическийОстаткиИОбороты.СуммаОборотКт,  
 УправленическийОстаткиИОбороты.СуммаКонечныйОстатокДт КАК СуммаКонДт,  
 УправленическийОстаткиИОбороты.СуммаКонечныйОстатокКт КАК СуммаКонКт

ИЗ

ПланСчетов.Основной КАК Основной

ЛЕВОЕ СОЕДИНЕНИЕ РегистрБухгалтерии.Управленический.ОстаткиИОбороты КАК

УправленическийОстаткиИОбороты

ПО (УправленическийОстаткиИОбороты.Счет = Основной.Ссылка)

#### УПОРЯДОЧИТЬ ПО

Код

### 17. План видов расчета, регистр расчета.

+ ВидРасчета +БазовыйПериод

+ План видов расчета (св. Использует период действия) (Зависимость от базы)

++ Зависимость от базы - по периоду действия -по периоду регистрации

+ ВидРасчета - базовый, вытесняющий, ведущий

++ Для премии – базов =оклад, ведущие - оклад и невыход

+++Рег.Сведений ++Спр.ВидыГрафиковРаботы

### 18. Использование регистра расчета

КонецДня(ТекСтрокаНачисления.ДатаОкончания)

+

Процедура ОбработкаПроведения(Отказ, Режим)

//{{ \_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

// Данный фрагмент построен конструктором.

// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

Движение.Начисления.Записывать = Истина;

Для Каждого ТекСтрокаНачисления Из Начисления Цикл

// регистр Начисления

Движение = Движение.Начисления.Добавить();

Движение.Стorno = Ложь;

Движение.ВидРасчета = ТекСтрокаНачисления.ВидРасчета;

Движение.ПериодДействияНачало = ТекСтрокаНачисления.ДатаНачала;

Движение.ПериодДействияКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);

Движение.ПериодРегистрации = ТекСтрокаНачисления.ДатаНачала;

Движение.БазовыйПериодНачало = ТекСтрокаНачисления.ДатаНачала;

Движение.БазовыйПериодКонец = КонецДня(ТекСтрокаНачисления.ДатаОкончания);

Движение.Сотрудник = ТекСтрокаНачисления.Сотрудник;

Движение.ГрафикРаботы = ТекСтрокаНачисления.ГрафикРаботы;

Движение.ИсходныеДанные = ТекСтрокаНачисления.Начислено;

КонецЦикла;

//}} \_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

КонецПроцедуры

---

#### - 19. Полнотекстовый поиск.

**-20. Выполнение заданий по расписанию.****++ имя метода +расписание****21. Редактирование движений в форме документа.****+ввод нач. остатков****+****&НаКлиенте**

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

// Вставить содержимое обработчика.

    Для Каждого ЗаписьРегистра ИЗ Объект.Движения.ОстаткиМатериалов Цикл  
        ЗаписьРегистра.Период=Объект.Дата;

КонецЦикла;

КонецПроцедуры

**+****22. Список пользователей и их роли.****+**

ВЫБРАТЬ 1 ИЗ Документ.НачисленияСотрудникам.Начисления ГДЕ

Документ.НачисленияСотрудникам.Начисления.Видрасчета=ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНач  
исления.Премия) И Документ.НачисленияСотрудникам.Начисления.Ссылка=#Параметр(1).Ссылка**++**

ВЫБРАТЬ 1 ИЗ Документ.НачисленияСотрудникам.Начисления ГДЕ

Документ.НачисленияСотрудникам.Начисления.Видрасчета=ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНач  
исления.Премия) И Документ.НачисленияСотрудникам.Начисления.Ссылка=#Параметр(1).Ссылка**++**

ДокНачисления ГДЕ НЕ 1 В (#ЕстьПремия("ДокНачисления"))

**24. Обмен данными****+константа для обмена**

Функция ПолучитьПрефиксНомера()Экспорт

Возврат Константы.ПрефиксНумерации.Получить();

КонецФункции

**+ спр**

Процедура ПриУстановкеНовогоКода (СтандартнаяОбработка, Префикс)

Префикс=Обмен.ПолучитьПрефиксНомера();

КонецПроцедуры

**+ ПланОбмена****++02 создание плана обмена****&НаСервере**

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Вставить содержимое обработчика.

Если Объект.Ссылка=ПланыОбмена.Филиалы.ЭтотУзел() Тогда

Элементы.Главный.Доступность=Ложь;

КонецЕсли

КонецПроцедуры

**+**

Процедура ЗарегистрироватьИзменения(Команда)

// Вставить содержимое обработчика.

РегистрацияИзмененийНаСервере(Элементы.Список.ТекущаяСтрока);

КонецПроцедуры

**+****&НаСервереБезКонтекста**

Процедура РегистрацияИзмененийНаСервере(Узел)

ПланыОбмена.ЗарегистрироватьИзменения(Узел);

КонецПроцедуры

```

+
&НаСервереБезКонтекста
Функция ПредопределенныйУзел(Узел)
    Возврат Узел=ПланыОбмена.Филиалы.ЭтотУзел();
КонецФункции
+
&НаКлиенте
Процедура СписокПриАктивизацииСтроки(Элемент)
    // Вставить содержимое обработчика.
    Если ПредопределенныйУзел(Элемент.ТекущаяСтрока)Тогда
        Элементы.ЗарегистрироватьИзменения.Доступность=Ложь;
    Иначе
        Элементы.ЗарегистрироватьИзменения.Доступность=Истина;
    КонецЕсли;

КонецПроцедуры
++3 создание процедур обмена
+
&НаКлиенте
Процедура ВыполнитьОбмен(Команда)
    // Вставить содержимое обработчика.
    ОбменСФилиалами();
КонецПроцедуры
+
&НаСервереБезКонтекста
Процедура ОбменСФилиалами()Экспорт
    ВыборкаУзлов=ПланыОбмена.Филиалы.Выбрать();
    Пока ВыборкаУзлов.Следующий() Цикл
        Если ВыборкаУзлов.Ссылка=>ПланыОбмена.Филиалы.ЭтотУзел() Тогда
            УзелОбъект=ВыборкаУзлов.ПолучитьОбъект();
            УзелОбъект.ПрочитатьСообщениеСИзменениями();
            УзелОбъект.ЗаписатьСообщениеСИзменениями();
        КонецЕсли;
    КонецЦикла;

КонецПроцедуры

25.Функциональные опции.

26. Подборы и ввод на основании.

+
&НаКлиенте
    Процедура Подбор(Команда)
        // Вставить содержимое обработчика.
        ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора",,Элементы.Материалы);

    КонецПроцедуры
+
&НаКлиенте
    Процедура МатериалыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)
        // Вставить содержимое обработчика.
        Элементы.Материалы.ДобавитьСтрочку();
        Элементы.Материалы.ТекущиеДанные.Материал=ВыбранноеЗначение;

    КонецПроцедуры

++замена
&НаКлиенте
    Процедура Подбор(Команда)
        // Вставить содержимое обработчика.
        ПараметрыФормы=Новый Структура("ЗакрыватьПриВыборе",Ложь);

```

```

ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора",ПараметрыФормы,Элементы.Материалы
);
КонецПроцедуры
++ или
++ ПараметрыФормы=Новый Структура("Множественныйвыбор",Истина);

&НаКлиенте
Процедура МатериалыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)
// Вставить содержимое обработчика.
Для Каждого ВыбранныйЭлемент из ВыбранноеЗначение Цикл
НоваяСтрока=Объект.Материалы.Добавить();
НоваяСтрока.Материал=ВыбранныйЭлемент;
КонецЦикла;
КонецПроцедуры

++ или
ПараметрыФормы=Новый Структура("ЗакрыватьПриВыборе,Множественныйвыбор",Ложь,Истина);

+++Ввод на основании
+
Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
//{{ __КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Клиенты") Тогда
// Заполнение шапки
Клиент = ДанныеЗаполнения.Ссылка;
ОбъектОснование = ДанныеЗаполнения.Ссылка;
КонецЕсли;
//}} __КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры
+
27.Приемы разработки форм.
+ для Справочник.Объект.Клиенты - ДанныеФормы Структура
+ ЗначениеВДанныеФормы() ДанныеФормыВЗначение()
+ ЗначениеВРеквизитФормы()

++Связанные списки
++Оформление строк в форме списка
+
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
// Вставить содержимое обработчика.
СписокСправочника=Элементы.Список;
СписокСправочника.Отображение=ОтображениеТаблицы.Список;
СписокСправочника.ВертикальныеЛинии=Ложь;
СписокСправочника.ГоризонтальныеЛинии=Ложь;
КонецПроцедуры

++Вычисляемые колонки в списке

++Список выбора для поля ввода

++Форма выбора для поля, содержащего ссылочный реквизит

++Проверка заполнения реквизитов

-- ++Использование параметризованных команд
+
&НаКлиенте

```

	<pre> Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)     // Вставить содержимое обработчика.     // ПараметрыФормы = Новый Структура("", );     // ОткрытьФорму("Документ.ОказаниеУслуги.ФормаСписка", ПараметрыФормы, ПараметрыВыполненияКоманды.Источник, ПараметрыВыполненияКоманды.Уникальность, ПараметрыВыполненияКоманды.Окно);     ПараметрыФормы = Новый Структура("Отбор,КлючНазначенияИспользования,СформироватьПриОткрытии", Новый Структура("Склад", ПараметрКоманды), "ОстаткиПоСкладу", Истина); ОткрытьФорму("Отчет.Материалы_о1.Форма", ПараметрыФормы, ПараметрыВыполненияКоманды.Источник, ПараметрыВыполненияКоманды.Уникальность, ПараметрыВыполненияКоманды.Окно);  КонецПроцедуры </pre>
1 7	<p style="text-align: center;"><b><u>004-gazv... ВАЖНО-обращение 1-28.</u></b></p>
1 8	<p><b>001... СправочникМенеджер.Номенклатура ССЫЛКА ОБЪЕКТ ВЫБОРКА</b></p> <p><b>002... Сумма = Количество * Цена +ФормаДокумента</b></p> <p><b>+</b></p> <p><b>&amp;НаКлиенте</b></p> <p><b>Процедура МатериалыКоличествоПриИзменении(Элемент)</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество *</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти.Цена;</b></p> <p><b>КонецПроцедуры</b></p> <p><b>++</b></p> <p><b>Процедура РассчитатьСумму(СтрокаТабличнойЧасти) Экспорт</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество *</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти.Цена;</b></p> <p><b>КонецПроцедуры</b></p> <p><b>++</b></p> <p><b>&amp;НаКлиенте</b></p> <p><b>Процедура МатериалыКоличествоПриИзменении(Элемент)</b></p> <p style="color: red;"><b>СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;</b></p> <p style="color: red;"><b>РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);</b></p> <p><b>КонецПроцедуры</b></p> <p><b>++</b></p> <p><b>СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;</b></p> <p style="color: red;"><b>РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);</b></p> <p><b>003м. Форма</b></p> <p><b>+</b></p> <p><b>&amp;НаКлиенте</b></p> <p><b>Процедура ФормаСписка (Справочник)</b></p> <p style="color: red;"><b>m="СправочникиКлиентыФормаСписка";</b></p> <p style="color: red;"><b>ФормаСписка=ПолучитьФорму("Справочники.Клиенты.ФормаСписка");</b></p> <p style="color: red;"><b>Сообщить(m);</b></p> <p style="color: red;"><b>УстановитьДействие("Получить","Событие");</b></p> <p><b>КонецПроцедуры</b></p> <p style="color: red;"><b>// обработчики назначаемые для форм ?</b></p> <p><b>1 004. Команды</b></p> <p><b>+</b></p>

**&НаКлиенте****Процедура Команда2(Команда)**

// Вставить содержимое обработчика.

Сообщить(ТекущаяДата());

История = ИсторияработыПользователя.Получить();

**КонецПроцедуры****004 .2 -****+****&НаКлиенте****Процедура Команда1\_сообщение()**

ТекстовоеСообщение();

**КонецПроцедуры****++****Процедура ТекстовоеСообщение()Экспорт**

Сообщить ("Текстовое сообщение1");

**КонецПроцедуры****006...****+****Процедура ОбработкаПроведения(Отказ, Режим)**

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

// Данный фрагмент построен конструктором.

    // При повторном использовании конструктора, внесенные вручную изменения будут  
утеряны!!!

Движения.ОстаткиМатериалов.Записывать = Истина;

Для Каждого ТекСтрокаМатериалы Из Материалы Цикл

// регистр ОстаткиМатериалов Приход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаМатериалы.Материал;

Движение.Склад = Склад;

Движение.Количество = ТекСтрокаМатериалы.Количество;

КонецЦикла;

//}}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

**КонецПроцедуры****008... Макеты. Редактирование макетов и форм.****Процедура Печать(ТабДок, Ссылка) Экспорт**

//{{ \_\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Документы.ОказаниеУслуги.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

|    ОказаниеУслуги.Дата,

|    ОказаниеУслуги.Клиент,

|    ОказаниеУслуги.Мастер,

|    ОказаниеУслуги.Номер,

|    ОказаниеУслуги.Склад,

|    ОказаниеУслуги.ПереченьНоменклатуры.(

|    НомерСтроки,

|    Номенклатура,

|    Количество,

|    Цена,

|    Сумма

|

|    ИЗ

|    Документ.ОказаниеУслуги КАК ОказаниеУслуги

|    ГДЕ

|    ОказаниеУслуги.Ссылка В (&amp;Ссылка)";

```

Запрос.Параметры.Вставить("Ссылка", Ссылка);
Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
Шапка = Макет.ПолучитьОбласть("Шапка");
ОбластьПереченьНоменклатурыШапка = Макет.ПолучитьОбласть("ПереченьНоменклатурыШапка");
ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("ПереченьНоменклатуры");
ОбластьИтог= Макет.ПолучитьОбласть("Всего");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;
Пока Выборка.Следующий() Цикл
    Если ВставлятьРазделительСтраниц Тогда
        ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
    КонецЕсли;

    ТабДок.Вывести(ОбластьЗаголовок);

    Шапка.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Шапка, Выборка.Уровень());

    ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);
    ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();

СуммаИтог=0;
Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл

    ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);
    ТабДок.Вывести(ОбластьПереченьНоменклатуры,
ВыборкаПереченьНоменклатуры.Уровень());
СуммаИтог= СуммаИтог+ВыборкаПереченьНоменклатуры.Сумма;
    КонецЦикла;
ОбластьИтог.Параметры.ВсегоПодДокументу=СуммаИтог;
ТабДок.Вывести(ОбластьИтог);
    Подвал.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Подвал);

    ВставлятьРазделительСтраниц = Истина;
    КонецЦикла;
//}}
КонецПроцедуры

```

#### 14+.

+

```

Процедура ОбработкаПроведения(Отказ, Режим)
//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут
утеряны!!!
Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать=Истина;
Движения.Продажи.Записывать=Истина;

//Создать менеджер временных таблиц
МенеджерВТ=Новый МенеджерВременныхТаблиц;
Запрос=Новый Запрос;
// укажем какой менеджер временных таблиц использует запрос
Запрос.МенеджерВременныхТаблиц=МенеджерВТ;
Запрос.Текст=

```

**"ВЫБРАТЬ**  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК  
**ВидНоменклатуры,**  
 | СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК  
**КоличествоВДокументе,**  
 | СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе  
**ПОМЕСТИТЬ Номенклатура**документа  
**ИЗ**  
 | Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК  
**ОказаниеУслугиПереченьНоменклатуры**  
**ГДЕ**  
 | ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка  
  
**СГРУППИРОВАТЬ ПО**  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";  
  
 Запрос.УстановитьПараметр("Ссылка", Ссылка);  
  
 Результат = Запрос.Выполнить();  
 Запрос2 =Новый Запрос;  
 Запрос2.МенеджерВременныхТаблиц=МенеджерВТ;  
 Запрос2.Текст="ВЫБРАТЬ  
 | Номенклатура<sup>документа</sup>.Номенклатура,  
 | Номенклатура<sup>документа</sup>.ВидНоменклатуры,  
 | Номенклатура<sup>документа</sup>.Количество<sup>вдокументе</sup>,  
 | Номенклатура<sup>документа</sup>.СуммаВДокументе,  
 | ЕСТЬNULL(СтоимостьМатериаловОстатки.СтоимостьОстаток, 0)  
**КАК Стоимость,**  
 | ЕСТЬNULL(ОстаткиМатериаловОстатки.Количество<sup>остаток</sup>, 0) КАК  
**Количество**  
**ИЗ**  
 | Номенклатура<sup>документа</sup> КАК Номенклатура<sup>документа</sup>  
 |      ЛЕВОЕ СОЕДИНЕНИЕ  
**РегистрНакопления.СтоимостьМатериалов.Остатки(**  
 |      , Материал В  
 |      (ВЫБРАТЬ  
  
 Номенклатура<sup>документа</sup>.Номенклатура  
**ИЗ**  
 |      Номенклатура<sup>документа</sup>)) КАК  
**СтоимостьМатериаловОстатки**  
 |      ПО Номенклатура<sup>документа</sup>.Номенклатура =  
**СтоимостьМатериаловОстатки.Материал**  
 |      ВНУТРЕННЕЕ СОЕДИНЕНИЕ  
**РегистрНакопления.ОстаткиМатериалов.Остатки(**  
 |      , Материал В  
 |      (ВЫБРАТЬ  
  
 Номенклатура<sup>документа</sup>.Номенклатура  
**ИЗ**  
 |      Номенклатура<sup>документа</sup>)) КАК  
**ОстаткиМатериаловОстатки**  
 |      ПО Номенклатура<sup>документа</sup>.Номенклатура =  
**ОстаткиМатериаловОстатки.Материал";**  
  
 // Установим необходимые блокировки данных в регистрах Стои Материалов и  
**ОстаткиМатериалов**  
**Движения.СтоимостьМатериалов.БлокироватьДляИзменения=Истина;**

**Движения.ОстаткиМатериалов.БлокироватьДляИзменения=Истина;**

```

// Запишем пустые наборы записей - чтобы читать остатки без учета данных в
документе
Движения.СтоимостьМатериалов.Записать();
Движения.ОстаткиМатериалов.Записать();

Результат =Запрос2.Выполнить();

//Т3=Результат.Выгрузить();

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
// Вставить обработку выборки ВыборкаДетальныеЗаписи

Если ВыборкаДетальныеЗаписи.Количество=0 Тогда
    СтоимостьМатериала=0;
    Иначе СтоимостьМатериала
=ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;
КонецЕсли;

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры=
    Перечисления.ВидыНоменклатуры.Материал Тогда

// регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход
Движение=Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения=ВидДвиженияНакопления.Расход;
Движение.Период=Дата;
Движение.Материал=ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Стоимость=ВыборкаДетальныеЗаписи.КоличествоВДокументе*
    СтоимостьМатериала;

КонецЕсли;

//////КонецЦикла;

//}}КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

// РегистрПродажи
Движение=Движения.Продажи.Добавить();
Движение.Период=Дата;
Движение.Номенклатура=ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Клиент=Клиент;
Движение.Мастер=Мастер;
Движение.Количество=ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.Выручка=ВыборкаДетальныеЗаписи.СуммаВДокументе;

```

**Движение.Стоимость=СтоимостьМатериала\***  
**ВыборкаДетальныеЗаписи.КоличествоВДокументе;**

**КонецЦикла;**  
**//}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ**

**Движения.Записать();**  
**Если Режим =РежимПроведенияДокумента.Оперативный Тогда**

**Запрос3=Новый Запрос;**

**Запрос3.МенеджерВременныхТаблиц=МенеджерВТ;**

**Запрос3.Текст="ВЫБРАТЬ**

**ОстаткиМатериаловОстатки.Материал,**  
**ОстаткиМатериаловОстатки.КоличествоОстаток**

**ИЗ**

**РегистрНакопления.ОстаткиМатериалов.Остатки(**

**,**  
**Материал В**

**(ВЫБРАТЬ**

**НоменклатураДокумента.Номенклатура**

**ИЗ**

**НоменклатураДокумента)**

**И Склад = &Склад) КАК ОстаткиМатериаловОстатки**

**ГДЕ**

**ОстаткиМатериаловОстатки.КоличествоОстаток < &0";**

**Запрос3.УстановитьПараметр("Склад",Склад);**

**Результат=Запрос3.Выполнить();**

**ВыборкаДетальныеЗаписи=Результат.Выбрать();**

**Пока ВыборкаДетальныеЗаписи.Следующий()Цикл**

**Сообщение=Новый СообщениеПользователю();**

**Сообщение.Текст="Не хватает"+ Стока(-**

**ВыборкаДетальныеЗаписи.КоличествоОстаток) +**

**"единиц материала"""+ВыборкаДетальныеЗаписи.Материал+"";**

**Сообщение.Сообщить();**

**Отказ=Истина;**

**КонецЦикла;**

**// Проверить отрицательные остатки**

**КонецЕсли;**

**КонецПроцедуры**

### **Кратко - 003-разв... ПЛАН.полный 1-28.**

01. Создание ИБ. 2. Подсистемы. – картинки, +сортировка-командный интерфейс + правая-свойства +Подсистемы-все подсистемы +ВИДИМОСТЬ ++Панель навигации ++действий ++рабочая область
02. –Справочники-данные реквизиты –изменить
03. –КонтекстМодуляФормы = + лококонтекст,+ рекв.формы, +св и мет Управл форма + св и мет расш форм +глоб конт +экспорт переем
04. **Форма\_документ ++АНАЛИЗ\_КОДА** = +1модуль формы +2реквизиты +3УправляемаяФорма\_свойство +4Расширение управляемой формы для документа\_свойство +5Глоб контекст\_свойство
05. ++б имя неглоб общ модуля ++ не проц глоб мод ++не перемен управ прил  
 === лок перемен модуля  
**АНАЛИЗ\_КОДА 2 = Элементы.Материалы.ТекущиеДанные =ctrl+f1 + == ctrl+f**  
 + Сервис-Параметры-Тонкий клиент  
**АНАЛИЗ\_КОДА 3 = отладчик (при написании своей программы) == значки +**  
**+f5 + Элементы.Материалы.ТекущиеДанные; + выделить + Элементы**  
**+ \_shift+f9 (вычислить выражение) \_кнопка F2 ( в отдельном окне)**  
**+стек вызовов + шагнуть через F10 + +++ ЭтаФорма \_рассчитать или ЭтотОбъект**
06. Регистры накопления.- измерения, ресурсы.

Регистры накопления + Документы ++ командный интерфейс \_форма  
 +++ регистратор\_документ +++РН \_форма\_команд\_интерфейс +++++КОЛЛЕКЦИЯ  
 ++ Справочник.Сотрудники ++ Справочник["Сотрудники"] ++Табличнаячасть[0]

06. Отчет. +Макет +Запрос  
 +Настройки-Отчет-Детальные записи (ins)

07. Макеты. Редактирование макетов и форм.  
 +ОказаниеУслуги +Макеты +Печать =Создание  
 +ОказаниеУслуги +Макеты +Печать =Редактирование  
 +ОказаниеУслуги +Прочее+МодульМенеджера =код

ОбластьИтог= Макет.ПолучитьОбласть("Всего");  
 СуммаИтог=0;  
 СуммаИтог= СуммаИтог+ВыборкаПереченьНоменклатуры.Сумма;  
 ОбластьИтог.Параметры.ВсегоПодДокументу=СуммаИтог;  
 ТабДок.Вывести(ОбластьИтог);

9.Периодический регистр сведений. Чтение-запись на сервере.

+  
 Функция РозничнаяЦена(АктуальнаяДата,ЭлементНоменклатуры)Экспорт  
 Отбор=Новый Структура("Номенклатура",ЭлементНоменклатуры);

ЗначенияРесурсов=РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,Отбор);  
 Возврат ЗначенияРесурсов.Цена;  
 КонецФункции

+  
 Процедура ПереченьНоменклатурыНоменклатураПриИзменении(Элемент)  
 // Вставить содержимое обработчика.  
 //Получить текущую строку  
 СтрокаТабличнойЧасти=Элементы.ПереченьНоменклатуры.ТекущиеДанные;  
 //Установить цену  
 СтрокаТабличнойЧасти.Цена=РаботаСоСправочниками.РозничнаяЦена(Объект.Дата,СтрокаТабличн  
 ойЧасти.Номенклатура);  
 //Пересчитать сумму строки  
 РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);  
 КонецПроцедуры

10.Перечисления.

+регистрция расхода ном= материал  
 + ОказаниеУслуги

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры=  
 Перечисления.ВидыНоменклатуры.Материал Тогда  
 КонецЕсли;

11.Проведение документа по нескольким регистрам. РЕГИСТР ОСТАТКОВ.  
 Регистр СтоимостьМатериалов

+,, ПриходнаяНакладная  
 Движения.СтоимостьМатериалов.Записывать=Истина;  
 // регистр СтоимостьМатериалов приход  
 Движение=Движения.СтоимостьМатериалов.Добавить();  
 Движение.ВидДвижения=ВидДвиженияНакопления.Приход;  
 Движение.Период=Дата;  
 Движение.Материал=ТекСтрокаМатериалы.Материал;  
 Движение.Стоимость=ТекСтрокаМатериалы.Сумма;  
 ++ Команд. Интерфейс формы  
 +++ Оказание Услуги ...

12.Оборотные регистры накопления. Продажи.(ОБОРОТЫ) Услуги = РЕГИСТР ОБОРОТОВ.

+++ создание регистра и для услуг и для товаров  
 +++ ком. Инт. = все подсистемы\_видимость\_см.также  
 + ОказаниеУслуги Движения вручную  
 +  
 КонецЕсли;

```

// РегистрПродажи
Движение=Движения.Продажи.Добавить();
Движение.Период=Дата;
Движение.Номенклатура=ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент=Клиент;
Движение.Мастер=Мастер;
Движение.Количество=ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка=ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость=ТекСтрокаПереченьНоменклатуры.Стоимость*
ТекСтрокаПереченьНоменклатуры.Количество;

```

### 13.Отчеты.

2 способа доступа к данным БД – объектный RW и табличный R

+ Объектный = ДокументОбъект.ОказаниеУслуги (обр. событ... целостность)  
+ Табличный - с пом. Запросов (больш об инф.) = реальн табл (обък +необъект) + вирт.  
++Объект. Таб - поле Ссылка (на тек запись) +польз. Представл.

+++Язык запросов

+++Система комп. Данных = схема комп данн +

+++ СиКД = +СКД+КомпоновщикМакета+ПроцессорКомпановки+РезультатКомпановки

#### ++1 Материалы\_01

+

ВЫБРАТЬ

ОстаткиМатериаловОстаткиИОбороты.Склад,  
ОстаткиМатериаловОстаткиИОбороты.Материал,  
ОстаткиМатериаловОстаткиИОбороты.КоличествоНачальныйОстаток,  
ОстаткиМатериаловОстаткиИОбороты.КоличествоПриход,  
ОстаткиМатериаловОстаткиИОбороты.КоличествоРасход,  
ОстаткиМатериаловОстаткиИОбороты.КоличествоКонечныйОстаток

ИЗ

РегистрНакопления.ОстаткиМатериалов.ОстаткиИОбороты КАК  
ОстаткиМатериаловОстаткиИОбороты

#### ++2 РеестрДокументовОказаниеУслуги\_02

ВЫБРАТЬ

ОказаниеУслуги.Склад,  
ОказаниеУслуги.Мастер,  
ОказаниеУслуги.Клиент,  
ОказаниеУслуги.Ссылка КАК Документ

ИЗ

Документ.ОказаниеУслуги КАК ОказаниеУслуги

УПОРЯДОЧИТЬ ПО

Документ

+++++1 +Отчет\_Добавить +СКД +запрос+Таблица\_Поля ++Псевдонимы++Порядок  
++Настройки\_НоваяГруппировка\_Поля +++Подсистемы

#### ++3 РЕЙТИНГ УСЛУГ из 2 табл

ВЫБРАТЬ

СпрНоменклатура.Ссылка КАК Услуга,  
ПродажиОбороты.ВыручкаОборот КАК Выручка

ИЗ

Справочник.Номенклатура КАК СпрНоменклатура  
ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты  
ПО ПродажиОбороты.Номенклатура = СпрНоменклатура.Ссылка

ГДЕ

СпрНоменклатура.ЭтоГруппа = &ЛОЖЬ  
И СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры

УПОРЯДОЧИТЬ ПО

Выручка УБЫВ

++ КонецПериода(&ДатаОкончания,"День")

+++++3.2.1 +Отчет\_Добавить +СКД +запрос+Таблица\_Поля ++Связи

СпрНоменклатура.ЭтоГруппа = ЛОЖЬ

СпрНоменклатура.ВидНоменклатуры = ВидНоменклатуры

++Псевдонимы++Порядок ++ресурсы +параметры

++Настройки\_НоваяГруппировка\_Поля

КонецПериода(&ДатаОкончания,"День")

++ состав даты –тип Дата

ВидНоменклатуры.Услуга

++Выбранные поля + Другие настройки

+++Подсистемы

+++++3.2 +Отчет оформление

+Настройки –Условное оформление –Цвет + Выбранные поля

++Кнопка вверху – Свойства элемента польз. Настроек +Отбор ++

+++++ 4 Выручка мастеров +Отчет\_Добавить +СКД +запрос+ Таблицы ++Параметры\_вирт\_табл (Период- День)

ВЫБРАТЬ

ПродажиОбороты.Мастер,  
ПродажиОбороты.Период,  
ПродажиОбороты.Клиент,  
ПродажиОбороты.ВыручкаОборот КАК Выручка

ИЗ

РегистрНакопления.Продажи.Обороты(, , День, ) КАК ПродажиОбороты

+ Ресурсы +Параметра КонецПериода(&ДатаОкончания,"День")

++ 4.2 Настройки – Детальные записи --Поле группировки –Период

++ 4.3 Диаграмма изм-я +СКД +Вариант отчета + ОбъемВыручки +Диаграмма(стру отчета)

+Точки =мастер +Друг\_настройки =Полосы измерительной диаграммы

+++++ 5 Актуальные значения из периодических регистров сведений.

Перечень услуг +Запрос (Номенкл+ценыСрезПоследних) +&ДатаОтчета

+СВЯЗИ+УСЛОВИЯ+Псевдонимы

+

ВЫБРАТЬ

СпрНоменклатура.Родитель КАК ГруппаУслуг,  
СпрНоменклатура.Ссылка КАК Услуга,  
ЦеныСрезПоследних.Цена

ИЗ

Справочник.Номенклатура КАК СпрНоменклатура

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних(&ДатаОтчета, ) КАК

ЦеныСрезПоследних

ПО ЦеныСрезПоследних.Номенклатура = СпрНоменклатура.Ссылка

ГДЕ

СпрНоменклатура.ВидНоменклатуры = &ВидНоменклатуры

++РЕСУРСЫ ++Параметры (+ВидНоменклатуры =услуга)

++Настройки (группировки =групп\_ усл\_иерархия+дет записи) +выбр поля

++Другие настройки +Дата отчета\_Быстрый доступ

+++++6 +Отчет\_Добавить +СКД +запрос+ РейтингКлиентов\_об

+

ВЫБРАТЬ

ПродажиОбороты.Клиент,  
ПродажиОбороты.ВыручкаОборот КАК Выручка,  
ПродажиОбороты.СтоимостьОборот КАК Стоимость

ИЗ

РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

+Выч поля = Доход Выручка-Стоимость  
 ++Ресурсы ++Настройки -диаграмма -точки -Клиент +Выбран поля\_Доход  
 +Вид диаг + назва +++Подсистемы

#### +++++7 +Отчет универсальный

+

ВЫБРАТЬ

ПродажиОбороты.Номенклатура,  
 ПродажиОбороты.Клиент,  
 ПродажиОбороты.Мастер,  
 ПродажиОбороты.КоличествоОборот,  
 ПродажиОбороты.ВыручкаОборот,  
 ПродажиОбороты.СтоимостьОборот

ИЗ

РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

++Запрос ++Ресурсы \_поля

++Настройки-Добавить -новая таблица

++Таблица -польз настройки

#### 14. Оптимизация проведения документа «Оказание услуги».

Минусы = через точку, опред.стои-ти, контроль остатко ОП.

+ СправочникСсылка.Номенклатура - обращение

++ Необъектн. Данные – в виде записей и полей

+ ДокументОбъект.ОказаниеУслуги при проведении

-- - СправочникВыборка.Клиенты

+++ Док - если инфо из шапки = исп. Конструктора дв.

- если инфо табл.+ итоги регистров = исп. Запросы

-- Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры

+ ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

+

//{ {КОНСТРУКТОР\_ЗАПРОСА\_С\_ОБРАБОТКОЙ\_РЕЗУЛЬТАТА

// Данный фрагмент построен конструктором.

// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
 | ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК  
 ВидНоменклатуры,  
 | СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК  
 КоличествоВДокументе,

| СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,  
 | МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость

|ИЗ

| Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК

ОказаниеУслугиПереченьНоменклатуры

|ГДЕ

| ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

|

|СГРУППИРОВАТЬ ПО

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура,

| ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

Запрос.УстановитьПараметр("Ссылка", Ссылка);

Результат = Запрос.Выполнить();

```

ВыборкаДетальныеЗаписи = Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
    // Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;

//} }КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА

```

**Задача 01.+++ Создание запроса =ОПТИМИЗАЦИЯ ПРОВЕДЕНИЯ ДОКУМЕНТА.**

**Постановка задачи 01.**

- 001.Запрос1. Формирование временной таблицы из номенклатуры документа.
- 002.Запрос2. Стоимость и остатки из регистров накопления. +Блокировка данных.
- 003. Запись движения регистра. Движения.Записать();
- 004.Запрос3. Проверка отрицательных остатков.

+Войти ниже цикла - пр.кнопка \_КонструкторЗапросаСОбработкойРезультата

++Выбрать табл часть –ОказаниеУслуги +выбрать поля

++УСЛОВИЯ \_ССЫЛКА ++ групп.поле \_сумммир.поля (максимум –цена)

++ псевдонимы

++ ТекСтрокаПереченьНоменклатуры зам. На ВыборкаДетальныеЗаписи

**Задача 02.+++автоматический расчет стоимости =**

запрос= + Табл.часть –оказание услуги + регист стоим матер +регист остатки материал.

+1 –удал \_РЕКВИЗИТА \_стоимость из + реквиз + и формы док.

+2 временная таблица \_НоменклатураДокумента (из Дополнительно)

+3 вставка след запроса после Результат = Запрос.Выполнить();

Запрос2 =Новый Запрос;

Запрос2.МенеджерВременныхТаблиц=МенеджерВТ;

Запрос2.Текст="";

+создать Запрос –пр.кн.м. +Таблица +Создание врем.Табл. + 4 поля

+3.2 +1-ый РегистрНакопления.СтоимостьМатериалов.Остатки

+ для производительности = (параметры вирт. Таблицы)

Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента)

Запрос2.Текст="ВЫБРАТЬ

```

| НоменклатураДокумента.Номенклатура,
| НоменклатураДокумента.ВидНоменклатуры,
| НоменклатураДокумента.КоличествоВДокументе,
| НоменклатураДокумента.СуммаВДокументе,
| СтоимостьМатериаловОстатки.СтоимостьОстаток
| ИЗ
| НоменклатураДокумента КАК НоменклатураДокумента
|     ЛЕВОЕ СОЕДИНЕНИЕ

```

РегистрНакопления.СтоимостьМатериалов.Остатки(

```

| ,
|     Материал В
|     (ВЫБРАТЬ
|         НоменклатураДокумента.Номенклатура
|     ИЗ
|         НоменклатураДокумента)) КАК
| СтоимостьМатериаловОстатки
|     ПО НоменклатураДокумента.Номенклатура =
| СтоимостьМатериаловОстатки.Материал";

```

3.3 + ОстаткиМатериалов.Остатки рег. ...

3.4 +стои или колич=0 ЕСТЬNULL()

ЕСТЬNULL (СтоимостьМатериаловОстатки.СтоимостьОстаток,0)

**3/5/5 Добавить к полям функцию**

ЕСТЬNULL (СтоимостьМатериаловОстатки.СтоимостьОстаток,0)

**3.5.6 в конце**

Результат =Запрос2.Выполнить();

**3.6 Надо изменить способ получ стойности = стой/колич**

Если ВыборкаДетальныеЗаписи.Количество=0 Тогда

СтоимостьМатериала=0;

Иначе СтоимостьМатериала

=ВыборкаДетальныеЗаписи.Стоимость/ВыборкаДетальныеЗаписи.Количество;

КонецЕсли;

**3.6.2**

Замена ВыборкаДетальныеЗаписи.Стоимость на СтоимостьМатериала

**3.7 При перепроведении документа**

// Запишем пустые наборы записей - чтобы читать остатки без учета данных в документе

Движения.СтоимостьМатериалов.Записать();

Движения.ОстаткиМатериалов.Записать();

// перед

Результат =Запрос2.Выполнить();

**3.8 Т3=Результат.Выгрузить(); для проверки явной****3.9 Контроль остатков**

Движения.Записать();

Если Режим =РежимПровеенияДокумента.Оперативный Тогда

// Проверить отрицательные остатки

КонецЕсли;

КонецПроцедуры

**3.9.2****Материал В (ВЫБРАТЬ НоменклатураДокумента.Номенклатура ИЗ НоменклатураДокумента) И Склад=&Склад****ОстаткиМатериаловОстатки.КоличествоОстаток < 0****+ к запросу**

Запрос3.УстановитьПараметр("Склад",Склад);

Результат=Запрос3.Выполнить();

ВыборкаДетальныеЗаписи=Результат.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий()Цикл

Сообщение=Новый СообщениеПользователю();

Сообщение.Текст="Не хватает" + Стока(-ВыборкаДетальныеЗаписи.КоличествоОстаток) +  
"единиц материала"""+ВыборкаДетальныеЗаписи.Материал+""";

Сообщение.Сообщить();

Отказ=Истина;

КонецЦикла;

**3.10 Блокировка записей**

// Установим необходимые блокировки данных в регистрах Стои Материалов и ОстаткиМатериалов

Движения.СтоимостьМатериалов.БлокироватьДляИзменения=Истина;

Движения.ОстаткиМатериалов.БлокироватьДляИзменения=Истина;

**15... План видов характеристик.****++Объекты – 1. План видов характеристик - СвойстваНоменклатуры**

2. Спр- Варианты номен. 3.спр -Доп.Свойс.Ном.

4. Регистр сведений -Знач.Свойс.Ном.

++ Номенклатура+ +ВариантыНоменклатуры\_СПР+ДополнительныеСвойстваНоменклатуры\_СПР  
+СвойстваНоменклатуры\_ПВХ\_ЗначенияСвойствНоменклатуры\_РСв.

```

++ изменение формы для владельца
+
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    // Вставить содержимое обработчика.
    Если Параметры.Отбор.Свойство("Владелец") Тогда
        Элементы.Код.Видимость=Ложь;
    КонецЕсли;
КонецПроцедуры

++ в рег свед на форме списка
+
&НаСервере
Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)
    // Вставить содержимое обработчика.
    Если Параметры.Отбор.Свойство("НаборСвойств") Тогда
        Элементы.НаборСвойств.Видимость=Ложь;
    КонецЕсли;
КонецПроцедуры

++ Движение.НаборСвойств=ТекСтрокаМатериалы.НаборСвойств;
++ в запрос  | ОказаниеУслугиПереченьНоменклатуры.НаборСвойств,
+ |НоменклатураДокумента.НаборСвойств,

```

**16. Бухгалтерский учет**

+Спр +ПлХар +ПлСч +РеБу

+

Движения.Управленческий.Записывать=Истина;

// Регистр Управленческий

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.Товары;

Движение.СчетКт=ПланыСчетов.Основной.РасчетыСПоставщиками;

Движение.Период=Дата;

Движение.Сумма=ТекСтрокаМатериалы.Сумма;

Движение.КоличествоДт=ТекСтрокаМатериалы.Количество;

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубкonto.Материалы]

=ТекСтрокаМатериалы.Материал;

+

Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубкonto.Материалы]

Движение.СубконтоДт.Материалы

то же

++ ОказаниеУслуги

+ Движения.Управленческий.Записывать=Истина;

// Регистр Управленческий

// д62 (ДебиторскаяЗадолженность) -К 90 (капитал) РозничнаяСумма

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.ДебиторскаяЗадолженность;

Движение.СчетКт=ПланыСчетов.Основной.Капитал;

Движение.Период=Дата;

Движение.Сумма=ВыборкаДетальныеЗаписи.СуммаВДокументе;

Движение.СубконтоДт.Клиенты=Клиент;

// д90 (капитал) - к 41 (товары) - себестоимость

Движение=Движения.Управленческий.Добавить();

Движение.СчетДт=ПланыСчетов.Основной.Капитал;

Движение.СчетКт=ПланыСчетов.Основной.Товары;

Движение.Период=Дата;

Движение.Сумма=СтоимостьМатериала\*ВыборкаДетальныеЗаписи.КоличествоВДокументе;

Движение.КоличествоКт=ВыборкаДетальныеЗаписи.КоличествоВДокументе;

Движение.СубконтоКт.Материалы=ВыборкаДетальныеЗаписи.Номенклатура;

+ Отчет ОСВ

+  
ВЫБРАТЬ

Основной.Код КАК Код,  
Основной.Наименование,  
УправлеченческийОстаткиИОбороты.СуммаНачальныйОстатокДт КАК СуммаНачДт,  
УправлеченческийОстаткиИОбороты.СуммаНачальныйОстатокКт КАК СуммаНачКт,  
УправлеченческийОстаткиИОбороты.СуммаОборотДт,  
УправлеченческийОстаткиИОбороты.СуммаОборотКт,  
УправлеченческийОстаткиИОбороты.СуммаКонечныйОстатокДт КАК СуммаКонДт,  
УправлеченческийОстаткиИОбороты.СуммаКонечныйОстатокКт КАК СуммаКонКт

ИЗ

ПланСчетов.Основной КАК Основной  
ЛЕВОЕ СОЕДИНЕНИЕ РегистрБухгалтерии.Управлеченческий.ОстаткиИОбороты КАК  
УправлеченческийОстаткиИОбороты  
ПО (УправлеченческийОстаткиИОбороты.Счет = Основной.Ссылка)

## УПОРЯДОЧИТЬ ПО

Код

**17. План видов расчета, регистр расчета.**

+ ВидРасчета +БазовыйПериод  
+ План видов расчета (св. Использует период действия) (Зависимость от базы)  
++ Зависимость от базы - по периоду действия -по периоду регистрации  
+ ВидРасчета - базовый, вытесняющий, ведущий  
++ Для премии – базов =оклад, ведущие - оклад и невыход  
+++Рег.Сведений ++Спр.ВидыГрафиковРаботы

**18. Использование регистра расчета**

КонецДня(ТекСтроКаНачисления.ДатаОкончания)

+

Процедура ОбработкаПроведения(Отказ, Режим)

```
//{{ __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
//  // Данный фрагмент построен конструктором.
//  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
//  Движения.Начисления.Записывать = Истина;
//  Для Каждого ТекСтроКаНачисления Из Начисления Цикл
//    // регистр Начисления
//    Движение = Движения.Начисления.Добавить();
//    Движение.Сторно = Ложь;
//    Движение.ВидРасчета = ТекСтроКаНачисления.ВидРасчета;
//    Движение.ПериодДействияНачало = ТекСтроКаНачисления.ДатаНачала;
//    Движение.ПериодДействияКонец = КонецДня(ТекСтроКаНачисления.ДатаОкончания);
//    Движение.ПериодРегистрации = ТекСтроКаНачисления.ДатаНачала;
//    Движение.БазовыйПериодНачало = ТекСтроКаНачисления.ДатаНачала;
//    Движение.БазовыйПериодКонец = КонецДня(ТекСтроКаНачисления.ДатаОкончания);
//    Движение.Сотрудник = ТекСтроКаНачисления.Сотрудник;
//    Движение.ГрафикРаботы = ТекСтроКаНачисления.ГрафикРаботы;
//    Движение.ИсходныеДанные = ТекСтроКаНачисления.Начислено;
//  КонецЦикла;
// }} __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
```

КонецПроцедуры

**- 19. Полнотекстовый поиск.****-20. Выполнение заданий по расписанию.**

++ имя метода +расписание

**21. Редактирование движений в форме документа.**

**+ввод нач. остатков**

+  
&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

// Вставить содержимое обработчика.

Для Каждого ЗаписьРегистра ИЗ Объект.Движения.ОстаткиМатериалов Цикл  
ЗаписьРегистра.Период=Объект.Дата;

КонецЦикла;

КонецПроцедуры

+

**22. Список пользователей и их роли.**

+

ВЫБРАТЬ 1 ИЗ Документ.НачисленияСотрудникам.Начисления ГДЕ

Документ.НачисленияСотрудникам.Начисления.Видрасчета=ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНач  
исления.Премия) И Документ.НачисленияСотрудникам.Начисления.Ссылка=#Параметр(1).Ссылка

++

ВЫБРАТЬ 1 ИЗ Документ.НачисленияСотрудникам.Начисления ГДЕ

Документ.НачисленияСотрудникам.Начисления.Видрасчета=ЗНАЧЕНИЕ(ПланВидовРасчета.ОсновныеНач  
исления.Премия) И Документ.НачисленияСотрудникам.Начисления.Ссылка=#Параметр(1).Ссылка

++

ДокНачисления ГДЕ НЕ 1 В (#ЕстьПремия("ДокНачисления"))

**24. Обмен данными**

+константа для обмена

Функция ПолучитьПрефиксНомера()Экспорт

Возврат Константы.ПрефиксНумерации.Получить();

КонецФункции

+ спр

Процедура ПриУстановкеНовогоКода (СтандартнаяОбработка, Префикс)

Префикс=Обмен.ПолучитьПрефиксНомера();

КонецПроцедуры

**+ ПланОбмена****++02 создание плана обмена**

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Вставить содержимое обработчика.

Если Объект.Ссылка=ПланыОбмена.Филиалы.ЭтотУзел() Тогда

Элементы.Главный.Доступность=Ложь;

КонецЕсли

КонецПроцедуры

+

Процедура ЗарегистрироватьИзменения(Команда)

// Вставить содержимое обработчика.

РегистрацияИзмененийНаСервере(Элементы.Список.ТекущаяСтрока);

КонецПроцедуры

+

&НаСервереБезКонтекста

Процедура РегистрацияИзмененийНаСервере(Узел)

ПланыОбмена.ЗарегистрироватьИзменения(Узел);

КонецПроцедуры

+

&НаСервереБезКонтекста

Функция ПредопределенныйУзел(Узел)

Возврат Узел=ПланыОбмена.Филиалы.ЭтотУзел();

КонецФункции

```

+
&НаКлиенте
Процедура СписокПриАктивизацииСтроки(Элемент)
    // Вставить содержимое обработчика.
    Если ПредопределенныйУзел(Элемент.ТекущаяСтрока)Тогда
        Элементы.ЗарегистрироватьИзменения.Доступность=Ложь;
    Иначе
        Элементы.ЗарегистрироватьИзменения.Доступность=Истина;
    КонецЕсли;

КонецПроцедуры
++3 создание процедур обмена
+
&НаКлиенте
Процедура ВыполнитьОбмен(Команда)
    // Вставить содержимое обработчика.
    ОбменСФилиалами();
КонецПроцедуры
+
&НаСервереБезКонтекста
Процедура ОбменСФилиалами()Экспорт
    ВыборкаУзлов=ПланыОбмена.Филиалы.Выбрать();
    Пока ВыборкаУзлов.Следующий() Цикл
        Если ВыборкаУзлов.Ссылка=>ПланыОбмена.Филиалы.ЭтотУзел() Тогда
            УзелОбъект=ВыборкаУзлов.ПолучитьОбъект();
            УзелОбъект.ПрочитатьСообщениеСИзменениями();
            УзелОбъект.ЗаписатьСообщениеСИзменениями();
        КонецЕсли;
    КонецЦикла;

КонецПроцедуры

25.Функциональные опции.

26. Подборы и ввод на основании.

+
&НаКлиенте
    Процедура Подбор(Команда)
        // Вставить содержимое обработчика.
        ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора",,Элементы.Материалы);

    КонецПроцедуры
+
&НаКлиенте
    Процедура МатериалыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)
        // Вставить содержимое обработчика.
        Элементы.Материалы.ДобавитьСтрочку();
        Элементы.Материалы.ТекущиеДанные.Материал=ВыбранноеЗначение;

    КонецПроцедуры

++замена
&НаКлиенте
    Процедура Подбор(Команда)
        // Вставить содержимое обработчика.
        ПараметрыФормы=Новый Структура("ЗакрыватьПриВыборе",Ложь);

        ОткрытьФорму("Справочник.Номенклатура.ФормаВыбора",ПараметрыФормы,Элементы.Материалы
    );
    КонецПроцедуры

```

++ или  
 ++ ПараметрыФормы=Новый Структура("Множественныйвыбор",Истина);  
**&НаКлиенте**  
 Процедура МатериалыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)  
 // Вставить содержимое обработчика.  
 Для Каждого ВыбранныйЭлемент из ВыбранноеЗначение Цикл  
 НоваяСтрока=Объект.Материалы.Добавить();  
 НоваяСтрока.Материал=ВыбранныйЭлемент;  
 КонецЦикла;  
 КонецПроцедуры

++ или  
 ПараметрыФормы=Новый Структура("ЗакрыватьПриВыборе,Множественныйвыбор",Ложь,Истина);

#### **+++Ввод на основании**

+
 Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)  
 //{{ \_\_КОНСТРУКТОР\_ВВОД\_НА\_ОСНОВАНИИ  
 // Данный фрагмент построен конструктором.  
 // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!  
 Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Клиенты") Тогда  
 // Заполнение шапки  
 Клиент = ДанныеЗаполнения.Ссылка;  
 ОбъектОснование = ДанныеЗаполнения.Ссылка;  
 КонецЕсли;  
 //}} \_\_КОНСТРУКТОР\_ВВОД\_НА\_ОСНОВАНИИ

КонецПроцедуры

+

#### **27.Приемы разработки форм.**

+
 для Справочник.Объект.Клиенты - ДанныеФормы Структура  
 + ЗначениеВДанныеФормы() ДанныеФормыВЗначение()  
 + ЗначениеВРеквизитФормы()

#### **++Связанные списки**

#### **++ Оформление строк в форме списка**

+

**&НаСервере**

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Вставить содержимое обработчика.  
 СписокСправочника=Элементы.Список;  
 СписокСправочника.Отображение=ОтображениеТаблицы.Список;  
 СписокСправочника.ВертикальныеЛинии=Ложь;  
 СписокСправочника.ГоризонтальныеЛинии=Ложь;

КонецПроцедуры

#### **++Вычисляемые колонки в списке**

#### **++Список выбора для поля ввода**

#### **++Форма выбора для поля, содержащего ссылочный реквизит**

#### **++Проверка заполнения реквизитов**

#### **-- ++Использование параметризованных команд**

+

**&НаКлиенте**

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

// Вставить содержимое обработчика.  
 // ПараметрыФормы = Новый Структура("", );  
 // ОткрытьФорму("Документ.ОказаниеУслуги.ФормаСписка", ПараметрыФормы,  
 ПараметрыВыполненияКоманды.Источник, ПараметрыВыполненияКоманды.Уникальность,

	<p>ПараметрыВыполненияКоманды.Окно);  ПараметрыФормы = Новый  Структура("Отбор,КлючНазначенияИспользования,СформироватьПриОткрытии", Новый  Структура("Склад", ПараметрКоманды), "ОстаткиПоСкладу", Истина);  ОткрытьФорму("Отчет.Материалы_о1.Форма", ПараметрыФормы,  ПараметрыВыполненияКоманды.Источник, ПараметрыВыполненияКоманды.Уникальность,  ПараметрыВыполненияКоманды.Окно);    КонецПроцедуры</p>
2 0	<p style="text-align: center;"><b><u>005-КОД 1-28.</u></b></p> <p>{1,  {19,  {"Практическое пособие разработчика",1,0,"","", ""},  {5,  {"Занятие 4. Документы",1,0,"","", ""},  {0,  {"4.1 Процедура МатериалыКоличествоПриИзменении()",0,0,"4.1","СтрокаТабличнойЧасти =  Элементы.Материалы.ТекущиеДанные;  СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество *  СтрокаТабличнойЧасти.Цена;  "}  },  {0,  {"4.2 Процедура РассчитатьСумму()",0,0,"4.2","Процедура  РассчитатьСумму(СтрокаТабличнойЧасти) Экспорт    СтрокаТабличнойЧасти.Сумма = СтрокаТабличнойЧасти.Количество *  СтрокаТабличнойЧасти.Цена;    КонецПроцедуры  "}  },  {0,  {"4.3 Процедура  МатериалыКоличествоПриИзменении()",0,0,"4.3","РаботаСДокументами.РассчитатьСумму(СтрокаТ  абличнойЧасти);"}  },  {0,  {"4.4 Процедура МатериалыЦенаПриИзменении()",0,0,"4.4","СтрокаТабличнойЧасти =  Элементы.Материалы.ТекущиеДанные;  РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);  "}  },  {0,  {"4.5 Процедура ПереченьНоменклатурыКоличествоПриИзменении() и  ПереченьНоменклатурыЦенаПриИзменении()",0,0,"","", "СтрокаТабличнойЧасти = Элементы.  ПереченьНоменклатуры.ТекущиеДанные;  РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);  "}  },  {1,  {"Занятие 8. Редактирование макетов и форм",1,0,"","", ""},  {0,  {"8.1 Печать формы документа в модуле менеджера",0,0,"8.1",""  Процедура Печать(ТабДок, Ссылка) Экспорт                    //{{_КОНСТРУКТОР_ПЕЧАТИ(Печать                    Макет = Документы.ОказаниеУслуги.ПолучитьМакет("Печать"));</p>

```

Запрос = Новый Запрос;
Запрос.Текст =
"""ВЫБРАТЬ
|   ОказаниеУслуги.Дата,
|   ОказаниеУслуги.Клиент,
|   ОказаниеУслуги.Мастер,
|   ОказаниеУслуги.Номер,
|   ОказаниеУслуги.Склад,
|   ОказаниеУслуги.ПереченьНоменклатуры.(
|       НомерСтроки,
|       Номенклатура,
|       Количество,
|       Цена,
|       Сумма
|   )
ИЗ
|   Документ.ОказаниеУслуги КАК ОказаниеУслуги
ГДЕ
|   ОказаниеУслуги.Ссылка В (&Ссылка)""";
Запрос.Параметры.Вставить("""Ссылка""", Ссылка);
Выборка = Запрос.Выполнить().Выбрать();

```

**ОбластьЗаголовок = Макет.ПолучитьОбласть("""Заголовок""");**

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

**ОбластьПереченьНоменклатурыШапка =**

**Макет.ПолучитьОбласть("""ПереченьНоменклатурыШапка""");**

**ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("""ПереченьНоменклатуры""");**

**ОбластьИтог = Макет.ПолучитьОбласть("""Всего""");**

**ТабДок.Очистить();**

**ВставлятьРазделительСтраниц = Ложь;**

**Пока Выборка.Следующий() Цикл**

**Если ВставлятьРазделительСтраниц Тогда**

**ТабДок.ВывестиГоризонтальныйРазделительСтраниц();**

**КонецЕсли;**

**ТабДок.Вывести(ОбластьЗаголовок);**

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

**ТабДок.Вывести(Шапка, Выборка.Уровень());**

**ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);**

**ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();**

**СуммаИтог = 0;**

**Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл**

**ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);**

**ТабДок.Вывести(ОбластьПереченьНоменклатуры,**

**ВыборкаПереченьНоменклатуры.Уровень());**

**СуммаИтог = СуммаИтог + ВыборкаПереченьНоменклатуры.Сумма;**

**КонецЦикла;**

**ОбластьИтог.Параметры.ВсегоПодДокументу = СуммаИтог;**

**ТабДок.Вывести(ОбластьИтог);**

**ВставлятьРазделительСтраниц = Истина;**

**КонецЦикла;**

**//}}**

**КонецПроцедуры**

**"}  
{**

```

    },
    {2,
    {"Занятие 9. Периодический регистр сведений",1,0,"","",""},

    {0,
    {"9.1 Функция РозничнаяЦена()",0,0,"9.1","Функция РозничнаяЦена(АктуальнаяДата,
    ЭлементНоменклатуры) Экспорт

    // Создать вспомогательный объект Отбор
    Отбор = Новый Структура("Номенклатура", ЭлементНоменклатуры);

    // Получить актуальные значения ресурсов регистра
    ЗначенияРесурсов = РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата, Отбор);
    Возврат ЗначенияРесурсов.Цена;

КонецФункции
    "}
    },
    {0,
    {"9.2 Процедура ПереченьНоменклатурыНоменклатураПриИзменении()",0,0,"9.2","// Получить
текущую строку табличной части
СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;

// Установить цену
СтрокаТабличнойЧасти.Цена = РаботаСoSправочниками.РозничнаяЦена(Объект.Дата,
СтрокаТабличнойЧасти.Номенклатура);

// Пересчитать сумму строки
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
    "}
    },
    {1,
    {"Занятие 10. Перечисления",1,0,"","",""},

    {0,
    {"10.2 Движения документа ОказаниеУслуги",0,0,"10.2","Процедура ОбработкаПроведения(Отказ,
    Режим)

    //{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
    // Данный фрагмент построен конструктором.
    // При повторном использовании конструктора, внесенные вручную изменения будут
утеряны!!!
    Движения.ОстаткиМатериалов.Записывать = Истина;

    Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда
        // регистр ОстаткиМатериалов Расход
        Движение = Движения.ОстаткиМатериалов.Добавить();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Склад = Склад;
        Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
        КонецЕсли;
    КонецЦикла;
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ

КонецПроцедуры
    "}
    },
    {4,
}

```

```

{"Занятие 11. Проведение документа по нескольким регистрам",1,0,"","", ""},  

{0,  

 {"11.1 Движения документа ПриходнаяНакладная по регистру """Стоимость  

материалов""",0,0,"11.1", "// регистр Стоимость Материалов Приход  

Движение = Движения.СтоимостьМатериалов.Добавить();  

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;  

Движение.Период = Дата;  

Движение.Материал = ТекСтрокаМатериалы.Материал;  

Движение.Стоимость = ТекСтрокаМатериалы.Сумма;  

"}  

},  

{0,  

 {"11.2 Движения документа ПриходнаяНакладная",0,0,"11.2", "Процедура  

ОбработкаПроведения(Отказ, Режим)  

  

Движения.ОстаткиМатериалов.Записывать = Истина;  

Движения.СтоимостьМатериалов.Записывать = Истина;  

  

Для Каждого ТекСтрокаМатериалы Из Материалы Цикл  

  

// регистр ОстаткиМатериалов Приход  

Движение = Движения.ОстаткиМатериалов.Добавить();  

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;  

Движение.Период = Дата;  

Движение.Материал = ТекСтрокаМатериалы.Материал;  

Движение.Склад = Склад;  

Движение.Количество = ТекСтрокаМатериалы.Количество;  

  

// регистр Стоимость Материалов Приход  

Движение = Движения.СтоимостьМатериалов.Добавить();  

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;  

Движение.Период = Дата;  

Движение.Материал = ТекСтрокаМатериалы.Материал;  

Движение.Стоимость = ТекСтрокаМатериалы.Сумма;  

КонецЦикла;  

  

КонецПроцедуры  

"}  

},  

{0,  

 {"11.3 Движения документа ОказаниеУслуги по регистру """Стоимость материалов""",0,0,"11.3", "//  

регистр СтоимостьМатериалов Расход  

Движение = Движения.СтоимостьМатериалов.Добавить();  

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  

Движение.Период = Дата;  

Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;  

Движение.Стоимость =  

ТекСтрокаПереченьНоменклатуры.Количество*ТекСтрокаПереченьНоменклатуры.Стоимость;  

"}  

},  

{0,  

 {"11.4 Движения документа ОказаниеУслуги",0,0,"11.4", "Процедура ОбработкаПроведения(Отказ,  

Режим)  

  

Движения.ОстаткиМатериалов.Записывать = Истина;  

Движения.СтоимостьМатериалов.Записывать = Истина;  

  

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл  

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =  

Перечисления.ВидыНоменклатуры.Материал Тогда  

  

// регистр ОстаткиМатериалов Расход

```

```

Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость =

ТекСтрокаПереченьНоменклатуры.Количество*ТекСтрокаПереченьНоменклатуры.Стоимость;
КонецЕсли;
КонецЦикла;

КонецПроцедуры
"}
},
{2,
{"Занятие 12. Оборотный регистр накопления",1,0,"","", ""},
{0,
{"12.1 Движения документа ОказаниеУслуги по регистру """Продажи""",0,0,"12.1","// Регистр Продажи
Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость *
ТекСтрокаПереченьНоменклатуры.Количество;
"}
},
{0,
{"12.2 Движения документа ОказаниеУслуги",0,0,"12.2","Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;
Движение.СтоимостьМатериалов.Записывать = Истина;
Движение.Продажи.Записывать = Истина;

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

// регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

// регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;

```







```
    | НоменклатураДокумента.КоличествоВДокументе,
    | НоменклатураДокумента.СуммаВДокументе,
...
"}  
}  
},  
{5,  
{"Занятие 16. Бухгалтерский учет",1,0,"","", ""},  
{0,  
{"16.1 Движения документа ПриходнаяНакладная по регистру ""Управленческий""",0,0,"16.1","/}  
Регистр Управленческий  
Движение = Движения.Управленческий.Добавить();  
Движение.СчетДт = ПланыСчетов.Основной.Товары;  
Движение.СчетКт = ПланыСчетов.Основной.РасчетыСПоставщиками;  
Движение.Период = Дата;  
Движение.Сумма = ТекСтрокаМатериалы.Сумма;  
Движение.КоличествоДт = ТекСтрокаМатериалы.Количество;  
Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубкonto.Материалы] =  
ТекСтрокаМатериалы.Материал;  
"}  
},  
{0,  
{"16.2 Движения документа ПриходнаяНакладная",0,0,"16.2","Процедура  
ОбработкаПроведения(Отказ, Режим)
```

**Движения.СтоимостьМатериалов.Записывать = Истина;**  
**Движения.Управленческий.Записывать = Истина;**

Для Каждого ТекСтрока Материалы из Материалы Цикл

```
// регистр ОстаткиМатериалов Приход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Материал = ТекСтроМатериалы.Материал;
Движение.НаборСвойств = ТекСтроМатериалы.НаборСвойств;
Движение.Склад = Склад;
Движение.Количество = ТекСтроМатериалы.Количество;
```

```
// регистр Стоимость Материалов Приход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаМатериалы.Материал;
Движение.Стоимость = ТекСтрокаМатериалы.Сумма;
```

```
// Регистр Управленческий
Движение = Движения.Управленческий.Добавить();
Движение.СчетДт = ПланыСчетов.Основной.Товары;
Движение.СчетКт = ПланыСчетов.Основной.РасчетыСПоставщиками;
Движение.Период = Дата;
Движение.Сумма = ТекСтрокаМатериалы.Сумма;
Движение.КоличествоДт = ТекСтрокаМатериалы.Количество;
Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубкonto.Материалы] =
    атериалы.Материал;
Цикла;
```

КонецПроцедуры

110

1

19  
{0.}

16.3 Движения документа ОказаниеУслуги по регистру ""Управленческий""",0,0,"16.3","// Регистр

```

Управленческий
// Первая проводка: Д 62(ДебиторскаяЗадолженность) – К 90 (Капитал) Розничная сумма
Движение = Движения.Управленческий.Добавить();
Движение.СчетДт = ПланыСчетов.Основной.ДебиторскаяЗадолженность;
Движение.СчетКт = ПланыСчетов.Основной.Капитал;
Движение.Период = Дата;
Движение.Сумма = ВыборкаДетальныеЗаписи.СуммаВДокументе;
Движение.СубконтоДт[ПланыВидовХарактеристик.ВидыСубкonto.Клиенты] = Клиент;
// Вторая проводка: Д 90 (Капитал) – К 41 (Товары) – себестоимость
Движение = Движения.Управленческий.Добавить();
Движение.СчетДт = ПланыСчетов.Основной.Капитал;
Движение.СчетКт = ПланыСчетов.Основной.Товары;
Движение.Период = Дата;
Движение.Сумма = СтоимостьМатериала * ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.КоличествоКт = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
Движение.СубконтоКт[ПланыВидовХарактеристик.ВидыСубкonto.Материалы] =
ВыборкаДетальныеЗаписи.Номенклатура;
"}
},
{0,
{"16.4 Запись движений регистра бухгалтерии",0,0,"16.4","Движения.Управленческий.Записывать =
Истина; "}
},
{0,
{"16.5 Выражение для расчета параметров НачалоПериода и
КонецПериода",0,0,"16.5","&Период.ДатаНачала
&Период.ДатаОкончания
"}
},
{11,
{"Занятие 18. Использование регистра расчета",1,0,"","",""},
{0,
{"18.2 Обработчик проведения документа НачисленияСотрудникам",0,0,"18.2","// Записываем
движения регистров
Движения.Начисления.Записать();"

// Получим список всех сотрудников, содержащихся в документе
Запрос = Новый Запрос(
    """ВЫБРАТЬ РАЗЛИЧНЫЕ
    | НачисленияСотрудникамНачисления.Сотрудник
    | ИЗ
    | Документ.НачисленияСотрудникам.Начисления
    | КАК НачисленияСотрудникамНачисления
    |
    | ГДЕ
    | НачисленияСотрудникамНачисления.Ссылка = &ТекущийДокумент""");
Запрос.УстановитьПараметр("ТекущийДокумент", Ссылка);

// Сформируем список сотрудников
ТаблЗнач = Запрос.Выполнить().Выгрузить();
МассивСотрудников = ТаблЗнач.ВыгрузитьКолонку("Сотрудник");

//Вызов процедуры РассчитатьНачисления из общего модуля
ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисления,
ПланыВидовРасчета.ОсновныеНачисления.Оклад, МассивСотрудников);
Движения.Начисления.Записать( , Истина);

ПроведениеРасчетов.РассчитатьНачисления(Движения.Начисления,
ПланыВидовРасчета.ОсновныеНачисления.Премия, МассивСотрудников);
Движения.Начисления.Записать( , Истина);
"}
}

```

```

},  

{0,  

{"18.3 Заготовка процедуры РассчитатьНачисления",0,0,"18.3","Процедура  

РассчитатьНачисления(НаборЗаписейРегистра, ТребуемыйВидРасчета, СписокСотрудников)  

Экспорт

    Регистратор=НаборЗаписейРегистра.Отбор.Регистратор.Значение;

    // Рассчитать первичные записи
    Если ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Оклад Тогда

        // Рассчитать вторичные записи
        ИначЕсли ТребуемыйВидРасчета = ПланыВидовРасчета.ОсновныеНачисления.Премия Тогда
        КонецЕсли;

    КонецПроцедуры
"}  

},  

{0,  

{"18.4 Изменение процедуры РасчитатьНачисления",0,0,"18.4","Запрос = Новый Запрос;  

Запрос.Текст =
    """ВЫБРАТЬ
    | НачисленияДанныеГрафика.ЗначениеПериодДействия КАК Норма,
    | НачисленияДанныеГрафика.ЗначениеФактическийПериодДействия КАК Факт,
    | НачисленияДанныеГрафика.НомерСтроки КАК НомерСтроки
    ИЗ
    | РегистрРасчета.Начисления.ДанныеГрафика(Регистратор = &Регистратор И
    | ВидРасчета = &ВидРасчета И Сотрудник В (&СписокСотрудников))
    | КАК НачисленияДанныеГрафика""";  

  

Запрос.УстановитьПараметр("""Регистратор""", Регистратор);
Запрос.УстановитьПараметр("""ВидРасчета""", ТребуемыйВидРасчета);
Запрос.УстановитьПараметр("""СписокСотрудников""", СписокСотрудников);  

  

ВыборкаРезультата = Запрос.Выполнить().Выбрать();
"}  

},  

{0,  

{"18.5 Добавление обхода набора записей и расчета первичных записей",0,0,"18.5","Для Каждого
ЗаписьРегистра Из НаборЗаписейРегистра Цикл
    СтруктураНомер = Новый Структура("""НомерСтроки""");
    СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
    ВыборкаРезультата.Сбросить();
    Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
        Если ВыборкаРезультата.Норма = 0 Тогда
            Сообщение = Новый СообщениеПользователю;
            Сообщение.Текст = """Вид расчета: Оклад – Нет рабочих дней в заданном
периоде""";
            Сообщение.Сообщить();
            ЗаписьРегистра.Результат = 0;
        Иначе
            // Рассчитать оклад по фактическому периоду и исходным данным
            ЗаписьРегистра.Результат = (ЗаписьРегистра.ИсходныеДанные
/ВыборкаРезультата.Норма) * ВыборкаРезультата.Факт;
            Сообщение = Новый СообщениеПользователю;
            Сообщение.Текст = """Выполнен расчет"""+ ЗаписьРегистра.Регистратор + """ –
""" + ЗаписьРегистра.ВидРасчета + """ – """
            + ЗаписьРегистра.Сотрудник;
            Сообщение.Сообщить();
        КонецЕсли;
    КонецЕсли;
}

```

```

КонецЦикла;
"}
},
{0,
{"18.6 Добавление текста запроса во вторую строку условия",0,0,"18.6","Запрос = Новый Запрос;
Запрос.Текст =
    """ВЫБРАТЬ
    | НачисленияБазаНачисления.РезультатБаза КАК База,
    | НачисленияБазаНачисления.НомерСтроки КАК НомерСтроки
    |ИЗ
    | РегистрРасчета.Начисления.БазаНачисления(&ИзмеренияОсновного,
    | &ИзмеренияБазового, , Регистратор = &Регистратор И ВидРасчета = &ВидРасчета И
    | Сотрудник В (&СписокСотрудников)) КАК НачисленияБазаНачисления""";
Измер = Новый Массив(1);
Измер[0] = """Сотрудник""";

Запрос.УстановитьПараметр("""ИзмеренияОсновного""", Измер);
Запрос.УстановитьПараметр("""ИзмеренияБазового""", Измер);
Запрос.УстановитьПараметр("""Регистратор""", Регистратор);
Запрос.УстановитьПараметр("""ВидРасчета""", ТребуемыйВидРасчета);
Запрос.УстановитьПараметр("""СписокСотрудников""", СписокСотрудников);

ВыборкаРезультата = Запрос.Выполнить().Выбрать(); "}
},
{0,
{"18.7 Добавление обхода набора записей регистра и вычисления результата вторичных
записей",0,0,"18.7","Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
    СтруктурНомер = Новый Структура("""НомерСтроки""");
    СтруктурНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
    ВыборкаРезультата.Сбросить();
    Если ВыборкаРезультата.НайтиСледующий(СтруктурНомер) Тогда
        ЗаписьРегистра.Результат = ВыборкаРезультата.База * (10 / 100);
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = """Выполнен расчет"""+ЗаписьРегистра.Регистратор + """ - """ +
ЗаписьРегистра.ВидРасчета + """ - """ + ЗаписьРегистра.Сотрудник;
        Сообщение.Сообщить();
    КонецЕсли;
КонецЦикла;

"}
},
{0,
{"18.8 Обработчик команды
Перерассчитать",0,0,"18.8","ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗнач
ение("""ПланВидовРасчета.ОсновныеНачисления.Оклад"""));
ПроведениеРасчетов.ПерерассчитатьНачисления(ПредопределенноеЗначение("""ПланВидовРасчета.О
сновныеНачисления.Премия"""));

"}
},
{0,
{"18.9 Процедура перерасчета начислений",0,0,"18.9","Процедура
ПерерассчитатьНачисления(ТребуемыйВидРасчета) Экспорт

    // Здесь следует выбрать из набора записей перерасчета записи в следующей
последовательности:
    // записи документа1 для сотрудников из списка,
    // записи документа2 для сотрудников из списка, и т. д.
    Запрос = Новый Запрос(
        """ВЫБРАТЬ

```

```

| НачисленияПерерасчет.ОбъектПерерасчета,
| НачисленияПерерасчет.Сотрудник
| ИЗ
| РегистрРасчета.Начисления.Перерасчет КАК НачисленияПерерасчет
|
| ГДЕ
| НачисленияПерерасчет.ВидРасчета = &ТребуемыйВидРасчета
|
| ИТОГИ ПО
| НачисленияПерерасчет.ОбъектПерерасчета"");

Запрос.УстановитьПараметр("""ТребуемыйВидРасчета""", ТребуемыйВидРасчета);
СписокСотрудников = Новый СписокЗначений;

// Перебрать группировку по регистратору.
ВыборкаПоРегистратору =
Запрос.Выполнить().Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
Пока ВыборкаПоРегистратору.Следующий() Цикл
    Регистратор = ВыборкаПоРегистратору.ОбъектПерерасчета;

    // Перебрать группировку по сотрудникам для выбранного регистратора
    // и создать список сотрудников.
    ВыборкаПоСотрудникам = ВыборкаПоРегистратору.Выбрать();
    СписокСотрудников.Очистить();

    Пока ВыборкаПоСотрудникам.Следующий() Цикл
        СписокСотрудников.Добавить(ВыборкаПоСотрудникам.Сотрудник);
    КонецЦикла;

    // Получить набор записей регистра расчета для выбранного регистратора.
    НаборЗаписей = РегистрыРасчета.Начисления.СоздатьНаборЗаписей();
    НаборЗаписей.Отбор.Регистратор.Значение = Регистратор;
    НаборЗаписей.Прочитать();

    РассчитатьНачисления(НаборЗаписей, ТребуемыйВидРасчета, СписокСотрудников);
    НаборЗаписей.Записать(, Истина);

    // Очистить перерассчитанные записи в перерасчете.
    НаборЗаписейПерерасчета =
РегистрыРасчета.Начисления.Перерасчеты.Перерасчет.СоздатьНаборЗаписей();
    НаборЗаписейПерерасчета.Отбор.ОбъектПерерасчета.Значение = Регистратор;
    НаборЗаписейПерерасчета.Записать();
    КонецЦикла;

КонецПроцедуры
"}
```

},  
{0,  
{"18.10 Обработчик команды  
Сформировать",0,0,"18.10","СформироватьНаСервере(ДиаграммаГанта);"}  
},  
{0,  
{"18.11 Шаблон процедуры СформироватьНаСервере()",0,0,"18.11","&НаСервереБезКонтекста  
Процедура СформироватьНаСервере(Диаграмма)  
  
Запрос = Новый Запрос;  
Запрос.Текст = ;  
  
КонецПроцедуры  
"}

},  
{0,



```

// Процедура поиска, получение и отображение результата
Процедура Исследовать(Направление)
    Если ПустаяСтрока(ПоисковоеВыражение) Тогда
        Предупреждение("Не задана строка поиска.");
        Возврат;
    КонецЕсли;

    ИсследоватьСервер(Направление);
КонецПроцедуры
};

{
{"19.3 Процедура поиска на сервере",0,0,"19.3","&НаСервере"
Процедура ИсследоватьСервер(Направление) Экспорт
    СписокПоиска = ПолнотекстовыйПоиск.СоздатьСписок();
    СписокПоиска.СтрокаПоиска = ПоисковоеВыражение;

    Если Направление = 0 Тогда
        СписокПоиска.ПерваяЧасть();
    ИначеЕсли Направление = -1 Тогда
        СписокПоиска.ПредыдущаяЧасть(ТекущаяПозиция);
    ИначеЕсли Направление = 1 Тогда
        СписокПоиска.СледующаяЧасть(ТекущаяПозиция);
    КонецЕсли;

    РезультатыПоиска.Очистить();
    Для Каждого Результат Из СписокПоиска Цикл
        РезультатыПоиска.Добавить(Результат.Значение);
    КонецЦикла;

    РезультатПоиска =
СписокПоиска.ПолучитьОтображение(ВидОтображенияПолнотекстовогоПоиска.HTMLТекст);
    ТекущаяПозиция = СписокПоиска.НачальнаяПозиция();
    ПолноеКоличество = СписокПоиска.ПолноеКоличество();

    Если СписокПоиска.Количество() <> 0 Тогда
        СообщениеОРезультате = """"Показаны """ + Стока(ТекущаяПозиция + 1) + """ - """
+
        Стока(ТекущаяПозиция + СписокПоиска.Количество()) + """ из """ +
Стока(ПолноеКоличество);
        Элементы.СледующаяПорция.Доступность = (ПолноеКоличество - ТекущаяПозиция) >
СписокПоиска.Количество();
        Элементы.ПредыдущаяПорция.Доступность = (ТекущаяПозиция > 0);
    Иначе
        СообщениеОРезультате = """Не найдено""";
        Элементы.СледующаяПорция.Доступность = Ложь;
        Элементы.ПредыдущаяПорция.Доступность = Ложь;
    КонецЕсли;
КонецПроцедуры
};

{
{"19.4 Обработчик события ПриНажатии() поля РезультатПоиска",0,0,"19.4","ЭлементHTML =
ДанныеСобытия.Event.srcElement;
Если (ЭлементHTML.id = ""FullTextSearchListItem"") Тогда

    // Получить имя файла (номер строки списка поиска), содержащегося в гиперссылке
    НомерВСписке = Число(ЭлементHTML.nameProp);

    // Получить строку списка поиска по номеру
    ВыбраннаяСтрока = РезультатыПоиска[НомерВСписке].Значение;
}

```

```

// Открыть форму найденного объекта
ОткрытьЗначение(ВыбраннаяСтрока);
СтандартнаяОбработка = Ложь;
КонецЕсли;
"}
},
},
{5,
{"Занятие 20. Выполнение заданий по расписанию",1,0,"","", ""},
{0,
{"20.1 Процедура ОбновлениеИндекса()",0,0,"20.1","Если
ПолнотекстовыйПоиск.ПолучитьРежимПолнотекстовогоПоиска() =
РежимПолнотекстовогоПоиска.Разрешить Тогда
Если Не ПолнотекстовыйПоиск.ИндексАктуален() Тогда
ПолнотекстовыйПоиск.ОбновитьИндекс( , Истина);
КонецЕсли;
КонецЕсли;
"}
},
{
{0,
{"20.2 Процедура СлияниеИндексов()",0,0,"20.2","Если
ПолнотекстовыйПоиск.ПолучитьРежимПолнотекстовогоПоиска() =
РежимПолнотекстовогоПоиска.Разрешить Тогда
Если Не ПолнотекстовыйПоиск.ИндексАктуален() Тогда
ПолнотекстовыйПоиск.ОбновитьИндекс(Истина);
КонецЕсли;
КонецЕсли;
"}
},
{
{0,
{"20.3 Обработчик события формы обработки ПланировщикЗаданий
ПриОткрытии()",0,0,"20.3","#Если ТолстыйКлиентУправляемоеПриложение Тогда
ПодключитьОбработчикОжидания("""ОбработкаЗаданий""", 3);
#Иначе
Предупреждение("""Обработка может быть запущена только в толстом клиенте!""");
Закрыть();
#КонецЕсли
},
{
{0,
{"20.4 Обработчик ожидания",0,0,"20.4","&НаКлиенте
Процедура ОбработкаЗаданий()
#Если ТолстыйКлиентУправляемоеПриложение Тогда
ВыполнитьОбработкуЗаданий();
#КонецЕсли

КонецПроцедуры
"}
},
{
{0,
{"20.5 Сообщение о запуске регламентного задания",0,0,"20.5","Сообщение = Новый
СообщениеПользователю;
Сообщение.Текст = """Запуск регламентного задания Обновление индекса """ + ТекущаяДата();
Сообщение.Сообщить();
"}
},
{
{2,
{"Занятие 21. Создание документа ввода начальных остатков",1,0,"","", ""},
{0,
{"21.1 Обработчик события ПередЗаписью формы документа",0,0,"21.1","Для Каждого

```





```

КонецПроцедуры
"}
},
{0,
{"24.7 Функция ПредопределенныйУзел()",0,0,"24.7","&НаСервереБезКонтекста
Функция ПредопределенныйУзел(Узел)

```

Возврат Узел = ПланыОбмена.Филиалы.ЭтотУзел();

КонецФункции

```

"}
},
{0,
{"24.8 Обработчик события СписокПриАктивизацииСтроки() элемента формы
Список",0,0,"24.8","Если ПредопределенныйУзел(Элемент.ТекущаяСтрока) Тогда
    Элементы.ЗарегистрироватьИзменения.Доступность = Ложь;
Иначе
    Элементы.ЗарегистрироватьИзменения.Доступность = Истина;
КонецЕсли;
}
},
{0,
{"24.9 Обработчик команды ВыполнитьОбмен",0,0,"24.9","ОбменСФилиалами();"}
},
{0,
{"24.10 Создание процедуры ОбменСФилиалами",0,0,"24.10","&НаСервереБезКонтекста
Процедура ОбменСФилиалами() Экспорт

```

ВыборкаУзлов = ПланыОбмена.Филиалы.Выбрать();

Пока ВыборкаУзлов.Следующий() Цикл

```

// Произвести обмен данными со всеми узлами, кроме текущего (ЭтотУзел)
Если ВыборкаУзлов.Ссылка <> ПланыОбмена.Филиалы.ЭтотУзел() Тогда
    УзелОбъект = ВыборкаУзлов.ПолучитьОбъект();

    // Получить сообщение
    УзелОбъект.ПрочитатьСообщениеСИзменениями();

    // Сформировать сообщение
    УзелОбъект.ЗаписатьСообщениеСИзменениями();

```

КонецЕсли;
КонецЦикла;

```

КонецПроцедуры
}
},
{0,
{"24.11 Формирование имени файла в процедуре записи данных",0,0,"24.11","Процедура
ЗаписатьСообщениеСИзменениями() Экспорт

```

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "----- Выгрузка в узел " + Стока(ЭтотОбъект) + " -----";
Сообщение.Сообщить();
Каталог = КаталогВременныхФайлов();

// Сформировать имя временного файла
ИмяФайла = Каталог + ?(Прав(Каталог, 1) = "\\", "\\\\", "\\\\"") + "Message" + СокрЛП(ПланыОбмена.
Филиалы.ЭтотУзел().Код) + "\_" + СокрЛП(Ссылка.Код) + ".xml";

```

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = """----- Конец выгрузки -----""";
Сообщение.Сообщить();

КонецПроцедуры
"}
```

```

},
```

```

{0,
```

```

{"24.12. Создание объекта записи XML в процедуре записи данных",0,0,"24.12","" // Создать объект
записи XML
    // *** ЗаписьXML-документов
    ЗаписьXML = Новый ЗаписьXML;
    ЗаписьXML.ОткрытьФайл(ИмяФайла);
    ЗаписьXML.ЗаписатьОбъявлениеXML();
    ЗаписьXML.Закрыть();

"}
```

```

},
```

```

{0,
```

```

{"24.13 Создание очередного номера сообщения и запись заголовка сообщения в XML",0,0,"24.13",""
    // *** Инфраструктура сообщений
    ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
    ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Ссылка);
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "" Номер сообщения: "" + ЗаписьСообщения.НомерСообщения;
    Сообщение.Сообщить();
    ЗаписьСообщения.ЗакончитьЗапись();

"}
```

```

},
```

```

{0,
```

```

{"24.14 Получение выборки из записей регистрации изменений, предназначенных данному
узлу",0,0,"24.14","" // Получить выборку измененных данных
    // *** Механизм регистрации изменений
    ВыборкаИзменений =
ПланыОбмена.ВыбратьИзменения(ЗаписьСообщения.Получатель,ЗаписьСообщения.НомерСообщени
я);
"}
```

```

},
```

```

{0,
```

```

{"24.15 Перебор выборки записей и сериализация их в открытый XML-файл",0,0,"24.15","" Пока
ВыборкаИзменений.Следующий() Цикл
    // Записать данные в сообщение *** XML-сериализация
    ЗаписатьXML(ЗаписьXML, ВыборкаИзменений.Получить());
    КонецЦикла;

"}
```

```

},
```

```

{0,
```

```

{"24.16 Формирование имени файла, содержащего данные обмена",0,0,"24.16",""Процедура
ПрочитатьСообщениеСИзменениями() Экспорт

    Каталог = КаталогВременныхФайлов();

    // Сформировать имя файла
    ИмяФайла = Каталог +?(Прав(Каталог, 1) = ""\", \"\", \"\") +
    \"\"Message\"\" + СокрЛП(Ссылка.Код) + \"\"_\"\" +
СокрЛП(ПланыОбмена.Филиалы.ЭтотУзел().Код) + \"\".xml\"\";

    Файл = Новый Файл(ИмяФайла);
    Если Не Файл.Существует() Тогда
        Возврат;
    КонецЕсли;

    УдалитьФайлы(ИмяФайла);
}

```

```

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "----- Конец загрузки -----";
Сообщение.Сообщить();

КонецПроцедуры
"}
```

```

},
```

```

{0,
```

```

{"24.17.Добавление чтения найденного файла с данными обмена",0,0,"24.17","" // *** Чтение
документов XML
    // Попытаться открыть файл
    ЧтениеXML = Новый ЧтениеXML;
    Попытка
        ЧтениеXML.ОткрытьФайл(ИмяФайла);
    Исключение
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "Невозможно открыть файл обмена данными."";
        Сообщение.Сообщить();
    Возврат;
    КонецПопытки;
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "----- Загрузка из " + Стока(ЭтотОбъект) + " -----";
    Сообщение.Сообщить();
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = " - Считывается файл " + ИмяФайла;
    Сообщение.Сообщить();

    ЧтениеXML.Закрыть();
"}
```

```

},
```

```

{0,
```

```

{"24.18 Добавление чтения заголовка XML-сообщения",0,0,"24.18","" // Загрузить из найденного
файла
    // *** Инфраструктура сообщений
    ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();

    // Читать заголовок сообщения обмена данными – файла XML
    ЧтениеСообщения.НачатьЧтение(ЧтениеXML);

    ЧтениеСообщения.ЗакончитьЧтение();
"}
```

```

},
```

```

{0,
```

```

{"24.19 Добавление проверки сообщения",0,0,"24.19","" // Сообщение предназначено не для этого узла
    Если ЧтениеСообщения.Отправитель <> Ссылка Тогда ВызватьИсключение "Неверный
узел";
    КонецЕсли;
"}
```

```

},
```

```

{0,
```

```

{"24.20 Удаление записей регистрации изменений для узла отправителя",0,0,"24.20","" // Удаляем
регистрацию изменений для узла отправителя сообщения.
    // *** Служба регистрации изменений
    ПланыОбмена.УдалитьРегистрациюИзменений(ЧтениеСообщения.Отправитель,ЧтениеСообщения.НомерПринятого);
"}
```

```

},
```

```

{0,
```

```

{"24.21 Чтение данных из сообщения",0,0,"24.21","" // Читаем данные из сообщения *** XML-
сериализация
    Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл

```

```

    КонецЦикла;
}

},
{0,
{"24.22 Представление данных XML в виде значения, имеющего тип",0,0,"24.22","" // Читаем
очередное значение
    Данные = ПрочитатьXML(ЧтениеXML);
}
},
{0,
{"24.23 Разрешение возможных коллизий",0,0,"24.23","" // Не переносим изменение, полученное
в главный из неглавного, если есть регистрация изменения
    Если Не ЧтениеСообщения.Отправитель.Главный И
        ПланыОбмена.ИзменениеЗарегистрировано(ЧтениеСообщения.Отправитель,
Данные) Тогда
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "" – Изменения отклонены"";
        Сообщение.Сообщить();
    Продолжить;
    КонецЕсли;
}
},
{0,
{"24.24 Запись полученных данных",0,0,"24.24","" // Записать полученные данные
    Данные.ОбменДанными.Отправитель = ЧтениеСообщения.Отправитель;
    Данные.ОбменДанными.Загрузка = Истина;
    Данные.Записать();
}
},
{0,
{"24.25 Функция ПредопределенныйУзел()",0,0,"24.25","&НаСервереБезКонтекста
Функция ПредопределенныйУзел(Узел)

    Возврат Узел = ПланыОбмена.Отделения.ЭтотУзел();

КонецФункции
}

},
{0,
{"24.26 Процедура ПолеВводаОтделениеОбработкаВыбора()",0,0,"24.26","" Если
ПредопределенныйУзел(ВыбранноеЗначение)Тогда
        Элементы.СоздатьНачальныйОбраз.Доступность = Ложь;
    Иначе
        Элементы.СоздатьНачальныйОбраз.Доступность = Истина;
    КонецЕсли;
}
},
{0,
{"24.27 Обработчик нажатия кнопки Создать начальный образ",0,0,"24.27","" Диалог = Новый
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.ВыборКаталога);
        Диалог.Заголовок = ""Укажите каталог информационной базы:"";
    Если Диалог.Выбрать() Тогда
        СоздатьНачальныйОбразНаСервере(ПолеВводаОтделение, Диалог.Каталог);
        Предупреждение(""Создание начального образа узла завершено."");
    КонецЕсли;
}
},
{0,
{"24.28 Процедура СоздатьНачальныйОбразНаСервере",0,0,"24.28","&НаСервереБезКонтекста
Процедура СоздатьНачальныйОбразНаСервере(Узел, КаталогСоединения)
}
}

```

```

ПланыОбмена.СоздатьНачальныйОбраз(Узел, ""File ="" + КаталогСоединения);

КонецПроцедуры
"}
},
{0,
{"24.29 Обработчик нажатия кнопки Записать изменения",0,0,"24.29","" Диалог = Новый
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Сохранение);
Диалог.Заголовок = """Укажите файл обмена:""";
Если Диалог.Выбрать() Тогда
    ЗаписьИзмененияНаСервере(ПолеВводаОтделение, Диалог.ПолноеИмяФайла);
    Предупреждение("""Запись изменений завершена."""");
КонецЕсли;
"}
},
{0,
{"24.30 Процедура ЗаписьИзмененияНаСервере",0,0,"24.30","&НаСерверБезКонтекста
Процедура ЗаписьИзмененияНаСервере(Узел, ИмяФайла)

// Создать и проинициализировать объект ЗаписьXML
ЗаписьXML = Новый ЗаписьXML;
ЗаписьXML.ОткрытьФайл(ИмяФайла);

// Создать объект ЗаписьСообщенияОбмена и начать запись сообщения
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);

// Записать содержимое тела сообщения обмена данными распределенной ИБ
ПланыОбмена.ЗаписьИзменения(ЗаписьСообщения);

// Закончить запись сообщения и запись XML
ЗаписьСообщения.ЗакончитьЗапись();
ЗаписьXML.Закрыть();

КонецПроцедуры
"}
},
{0,
{"24.31 Обработчик нажатия кнопки Прочитать изменения",0,0,"24.31","" Диалог = Новый
ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);
Диалог.Заголовок = """Укажите файл обмена:""";
Если Диалог.Выбрать() Тогда
    ПрочитатьИзмененияНаСервере(Диалог.ПолноеИмяФайла);
    Предупреждение("""Чтение изменений завершено."""");
КонецЕсли;
"}
},
{0,
{"24.32 Процедура ПрочитатьИзмененияНаСервере",0,0,"24.32","&НаСерверБезКонтекста
Процедура ПрочитатьИзмененияНаСервере(ИмяФайла)

// Создать и проинициализировать объект ЧтениеXML
ЧтениеXML = Новый ЧтениеXML;
ЧтениеXML.ОткрытьФайл(ИмяФайла);

// Создать объект ЧтениеСообщенияОбмена и начать чтение сообщения
ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
ЧтениеСообщения.НачатьЧтение(ЧтениеXML);

// Прочитать содержимое тела сообщения
ПланыОбмена.ПрочитатьИзменения(ЧтениеСообщения);

```

```
// Закончить чтение сообщения и чтение XML
```

```
ЧтениеСообщения.ЗакончитьЧтение();
```

```
ЧтениеXML.Закрыть();
```

КонецПроцедуры

```
"}
```

```
},
```

```
{0,
```

```
{"24.33 Просмотр работы событий объекта ПланОбменаОбъект",0,0,"24.33","Процедура
ПриОтправкеДанныхГлавному(ЭлементДанных, ОтправкаЭлемента)
```

```
Сообщить("""ПриОтправкеДанныхГлавному """+ ЭлементДанных);
```

КонецПроцедуры

```
Процедура ПриОтправкеДанныхПодчиненному(ЭлементДанных, ОтправкаЭлемента)
```

```
Сообщить("""ПриОтправкеДанныхПодчиненному """+ ЭлементДанных);
```

КонецПроцедуры

```
Процедура ПриПолученииДанныхОтГлавного(ЭлементДанных, ПолучениеЭлемента,
ОтправкаНазад)
```

```
Сообщить("""ПриПолученииДанныхОтГлавного """+ ЭлементДанных);
```

КонецПроцедуры

```
Процедура ПриПолученииДанныхОтПодчиненного(ЭлементДанных, ПолучениеЭлемента,
ОтправкаНазад)
```

```
Сообщить("""ПриПолученииДанныхОтПодчиненного """+ ЭлементДанных);
```

КонецПроцедуры

```
"}
```

```
},
```

```
{0,
```

```
{"24.34 Перемещение Узла2 в корень дерева",0,0,"24.34","// В информационной базе Узла2.
ПланыОбменаМенеджер.УстановитьГлавныйУзел(Неопределено);
```

// В информационной базе Узла1.

```
ПланыОбменаМенеджер.УстановитьГлавныйУзел(Узел2);
```

```
"}
```

```
},
```

```
{0,
```

```
{"24.35 Отключение поддерева от дерева",0,0,"24.35","// В информационной базе Узла1.
```

```
ПланыОбменаМенеджер.УстановитьГлавныйУзел(Неопределено);
```

```
"}
```

```
},
```

```
{0,
```

```
{"24.36 Создание распределенной информационной базы из баз с идентичной
конфигурацией",0,0,"24.36","// В информационных базах Узла2, Узла3 и Узла4.
```

```
ПланыОбменаМенеджер.УстановитьГлавныйУзел(Узел1);
```

```
"}
```

```
}
```

```
},
```

```
{1,
```

```
{"Занятие 25. Функциональные опции",1,0,"",""},
```

```
{0,
```

```
{"25.1 Обработчик события ПослЗаписи формы констант",0,0,"25.1","ОбновитьИнтерфейс();"
```

```
}
```

```
},
```

```

{6,
{"Занятие 26. Общие приемы разработки",1,0,"","",""},
{0,
{"26.1 Обработчик нажатия кнопки
Подбор",0,0,"26.1","ОткрытьФорму("""Справочник.Номенклатура.ФормаВыбора""",,
Элементы.Материалы);

"}
```

},  
{0,  
{"26.2 Обработчик события ОбработкаВыбора таблицы Материалы",0,0,"26.2","НоваяСтрока =
Объект.Материалы.Добавить();  
НоваяСтрока.Материал = ВыбранноеЗначение;  
"}

},  
{0,  
{"26.3 Обработчик нажатия кнопки Подбор",0,0,"26.3","ПараметрыФормы = Новый
Структура("""ЗакрыватьПриВыборе""", Ложь);
ОткрытьФорму("""Справочник.Номенклатура.ФормаВыбора""", ПараметрыФормы,
Элементы.Материалы);

```

"}
```

},  
{0,  
{"26.4 Обработчик нажатия кнопки Подбор",0,0,"26.4","ПараметрыФормы = Новый
Структура("""МножественныйВыбор""", Истина);
ОткрытьФорму("""Справочник.Номенклатура.ФормаВыбора""", ПараметрыФормы,
Элементы.Материалы);  
"}

},  
{0,  
{"26.5 Обработчик события ОбработкаВыбора таблицы Материалы",0,0,"26.5","Для Каждого
ВыбранныйЭлемент Из ВыбранноеЗначение Цикл
НоваяСтрока = Объект.Материалы.Добавить();  
НоваяСтрока.Материал = ВыбранныйЭлемент;
КонецЦикла;  
"}

},  
{0,  
{"26.6 Обработчик нажатия кнопки Подбор",0,0,"26.6","ПараметрыФормы = Новый
Структура("""ЗакрыватьПриВыборе, МножественныйВыбор""", Ложь, Истина);
ОткрытьФорму("""Справочник.Номенклатура.ФормаВыбора""", ПараметрыФормы,
Элементы.Материалы);  
"}

},  
{4,  
{"Занятие 27. Формы",1,0,"","",""},  
{0,  
{"27.3 Обработчик события формы ПриСозданииНаСервере",0,0,"27.3","СписокСправочника =
Элементы.Список;  
// Задать режим отображения справочника
СписокСправочника.Отображение = ОтображениеТаблицы.Список;  
// Скрыть линии сетки
СписокСправочника.ВертикальныеЛинии = Ложь;
СписокСправочника.ГоризонтальныеЛинии = Ложь;  
"}

