```
{codecitation style="brush: xml;"}
```

```
Процедура КнопкаВыполнитьНажатие(Кнопка) ТипДокумента =  
ЭлементыФормы.ПолеВыбора2.Значение; НомерДок =  
ЭлементыФормы.ПолеВыбора1.Значение; Если СокрЛП(ТипДокумента) = "" Тогда  
Предупреждение("Выберите тип документа"); ИначеЕсли СокрЛП(НомерДок) = ""  
Тогда Предупреждение("Выберите номер документа"); Иначе ДокСсылка =  
СсылкаНаДокумент(ТипДокумента,НомерДок); ДокОбъект =  
ДокСсылка.ПолучитьОбъект(); ТабТоварыТЗ =  
мПодготовитьТаблицыДокументаПоТовары(ДокОбъект); ТабУслугиТЗ =  
мПодготовитьТаблицыДокументаПоУслуги(ДокОбъект); ТабТараТЗ =  
мПодготовитьТаблицыДокументаПоТаре(ДокОбъект);  
ПоказатьТаблицу(ТабТоварыТЗ); ПоказатьТаблицу(ТабУслугиТЗ);  
ПоказатьТаблицу(ТабТараТЗ); КонецЕсли; КонецПроцедуры
```

```
{/codecitation}
```

```
{codecitation style="brush: xml;"}
```

```
// Процедура формирует таблицы документа. // функция
мПодготовитьТаблицыДокументаПоТовары(ЭтотОбъект) Экспорт // Получим
необходимые данные для проведения и проверки заполнения данных по табличной
части "Товары". СтруктураПолей = Новый Структура();
СтруктураПолей.Вставить("Номенклатура" , "Номенклатура");
СтруктураПолей.Вставить("ЕдиницаИзмерения" , "ЕдиницаИзмерения");
СтруктураПолей.Вставить("Цена" , "Цена");
СтруктураПолей.Вставить("Номенклатура" , "Номенклатура");
СтруктураПолей.Вставить("Услуга" , "Номенклатура.Услуга");
СтруктураПолей.Вставить("Набор" , "Номенклатура.Набор");
СтруктураПолей.Вставить("Комплект" , "Номенклатура.Комплект");
СтруктураПолей.Вставить("Количество" , "Количество * Коэффициент
/Номенклатура.ЕдиницаХраненияОстатков.Кoeffициент");
СтруктураПолей.Вставить("КоличествоДок" , "Количество");
СтруктураПолей.Вставить("Качество" , "Качество");
СтруктураПолей.Вставить("Склад" , "Склад");
СтруктураПолей.Вставить("ВидСклада" , "Склад.ВидСклада");
СтруктураПолей.Вставить("Сумма" , "Сумма");
СтруктураПолей.Вставить("СуммаДок" , "Сумма");
СтруктураПолей.Вставить("СтавкаНДС" , "СтавкаНДС");
СтруктураПолей.Вставить("НДС" , "СуммаНДС");
СтруктураПолей.Вставить("НДСДок" , "СуммаНДС");
СтруктураПолей.Вставить("ХарактеристикаНоменклатуры" ,
"ХарактеристикаНоменклатуры"); СтруктураПолей.Вставить("СерияНоменклатуры"
, "СерияНоменклатуры"); СтруктураПолей.Вставить("ПроцентСкидкиНаценки"
, "ПроцентСкидкиНаценки");
СтруктураПолей.Вставить("ПроцентАвтоматическихСкидок" ,
"ПроцентАвтоматическихСкидок");
СтруктураПолей.Вставить("УсловиеАвтоматическойСкидки" ,
"УсловиеАвтоматическойСкидки");
СтруктураПолей.Вставить("ЗначениеУсловияАвтоматическойСкидки",
"ЗначениеУсловияАвтоматическойСкидки"); СтруктураПолей.Вставить("КлючСтроки"
, "КлючСтроки"); СтруктураПолей.Вставить("ЗаказПокупателя"
, "ЗаказПокупателя"); СтруктураПолей.Вставить("КоличествоВЕдиницахДокумента"
, "Количество"); СтруктураПолей.Вставить("ЕдиницаИзмеренияМест"
, "ЕдиницаИзмеренияМест"); СтруктураПолей.Вставить("КоличествоМест"
, "КоличествоМест"); СтруктураПолей.Вставить("Коэффициент"
, "Коэффициент"); СтруктураПолей.Вставить("СпособСписанияОстаткаТоваров"
, "СпособСписанияОстаткаТоваров"); СтруктураПолей.Вставить("СуммаНДС"
, "СуммаНДС"); СтруктураПолей.Вставить("КлючСвязи"
, "КлючСвязи"); СтруктураСложныхПолей = Новый Структура;
СтруктураСложныхПолей = Неопределено; РезультатЗапросаПоТоварам =
```

Сканируем табличную часть документа 1с

Автор: А.Волос
19.02.2014 06:32

```
мСформироватьЗапросПоТабличнойЧасти(ЭтотОбъект, "Товары",  
СтруктураПолей, СтруктураСложныхПолей); // Подготовим таблицу  
ТаблицаПоТоварам = мПодготовитьТаблицу(РезультатЗапросаПоТоварам); Возврат  
ТаблицаПоТоварам; КонецФункции // СформироватьТаблицыДокумента() //  
Процедура формирует таблицы документа. // функция  
мПодготовитьТаблицыДокументаПоУслуги(ЭтотОбъект) Экспорт // Получим  
необходимые данные для проведения и проверки заполнения данных по табличной  
части "Услуги". СтруктураПолей = Новый Структура();  
СтруктураПолей.Вставить("Номенклатура" , "Номенклатура");  
СтруктураПолей.Вставить("Содержание" , "Содержание");  
СтруктураПолей.Вставить("Услуга" , "Номенклатура.Услуга");  
СтруктураПолей.Вставить("Набор" , "Номенклатура.Набор");  
СтруктураПолей.Вставить("Комплект" , "Номенклатура.Комплект");  
СтруктураПолей.Вставить("Количество" , "Количество");  
СтруктураПолей.Вставить("Цена" , "Цена"); СтруктураПолей.Вставить("Сумма"  
 , "Сумма"); СтруктураПолей.Вставить("ПроцентСкидкиНаценки",  
"ПроцентСкидкиНаценки"); СтруктураПолей.Вставить("СтавкаНДС"  
 , "СтавкаНДС"); СтруктураПолей.Вставить("НДС"  
 , "СуммаНДС");  
СтруктураПолей.Вставить("ЗаказПокупателя" , "ЗаказПокупателя");  
СтруктураПолей.Вставить("ПроцентАвтоматическихСкидок"  
 ,  
"ПроцентАвтоматическихСкидок");  
СтруктураПолей.Вставить("УсловиеАвтоматическойСкидки"  
 ,  
"УсловиеАвтоматическойСкидки");  
СтруктураПолей.Вставить("ЗначениеУсловияАвтоматическойСкидки",  
"ЗначениеУсловияАвтоматическойСкидки"); СтруктураСложныхПолей = Новый  
Структура; СтруктураСложныхПолей = Неопределено;  
РезультатЗапросаПоУслугам = мСформироватьЗапросПоТабличнойЧасти(ЭтотОбъект,  
"Услуги", СтруктураПолей, СтруктураСложныхПолей); // Подготовим таблицу  
ТаблицаПоУслугам = мПодготовитьТаблицу(РезультатЗапросаПоУслугам); Возврат  
ТаблицаПоУслугам; КонецФункции // СформироватьТаблицыДокумента() //  
Процедура формирует таблицы документа. // функция  
мПодготовитьТаблицыДокументаПоТаре(ЭтотОбъект) Экспорт // Получим  
необходимые данные для проведения и проверки заполнения данных // по табличной  
части "Возвратная тара". СтруктураПолей = Новый Структура();  
СтруктураПолей.Вставить("Номенклатура" , "Номенклатура");  
СтруктураПолей.Вставить("Услуга" , "Номенклатура.Услуга");  
СтруктураПолей.Вставить("Набор" , "Номенклатура.Набор");  
СтруктураПолей.Вставить("Комплект" , "Номенклатура.Комплект");  
СтруктураПолей.Вставить("Количество" , "Количество");  
СтруктураПолей.Вставить("ЕдиницаИзмерения"  
 ,  
"Номенклатура.ЕдиницаХраненияОстатков"); СтруктураПолей.Вставить("Сумма"  
 , "Сумма"); СтруктураПолей.Вставить("Склад"  
 , "Склад");  
СтруктураПолей.Вставить("ВидСклада" , "Склад.ВидСклада");  
СтруктураПолей.Вставить("ВестиУчетПоХарактеристикам",  
"Номенклатура.ВестиУчетПоХарактеристикам");  
СтруктураПолей.Вставить("ЗаказПокупателя" , "ЗаказПокупателя");
```

```

РезультатЗапросаПоТаре = мСформироватьЗапросПоТабличнойЧасти(ЭтотОбъект,
"ВозвратнаяТара", СтруктураПолей); // Подготовим таблицу ТаблицаПоТаре =
мПодготовитьТаблицу(РезультатЗапросаПоТаре); Возврат ТаблицаПоТаре;
КонецФункции // СформироватьТаблицыДокумента() // По переданной структуре
полей формирует запрос по табличной части документа. // // Параметры: //
ДокументОбъект - объект проводимого документа, // ИмяТабличнойЧасти -
строка, имя табличной части, // СтруктураПолей - структура, ключ структуры
содержит псевдоним поля запроса, значение - строку запроса, //
СтруктураСложныхПолей - структура, ключ структуры содержит псевдоним поля
запроса, значение - строку запроса, // необязательный параметр, служит
для передачи конструкций типа "ВЫБОР" и т.д. // // Возвращаемое значение: //
Результат запроса. // Функция
мСформироватьЗапросПоТабличнойЧасти(ДокументОбъект, ИмяТабличнойЧасти,
СтруктураПолей, СтруктураСложныхПолей = Неопределено)
Экспорт ТекстЗапроса = ""; ДокументМетаданные = ДокументОбъект.Метаданные();
Для Каждого Реквизит Из СтруктураПолей Цикл ТекстЗапроса = ТекстЗапроса + ",
|Док." + Реквизит.Значение + " КАК " + СокрЛП(Реквизит.Ключ); КонецЦикла;
ТекстСоединение=""; Запрос = Новый Запрос; Если
ТипЗнч(СтруктураСложныхПолей) = Тип("Структура") Тогда // Добавим к запросу
конструкции. Для Каждого Элемент Из СтруктураСложныхПолей Цикл
ТекстЗапроса = ТекстЗапроса + ", |" + Элемент.Значение + " КАК " +
СокрЛП(Элемент.Ключ); КонецЦикла; Если
СтруктураСложныхПолей.Свойство("Проект") Тогда ТекстСоединение=" |ЛЕВОЕ
СОЕДИНЕНИЕ
РегистрСведений.УстановкаПроектовДляНоменклатуры.СрезПоследних(&ДатаДок,
НоменклатураПроекта В (ВЫБРАТЬ Номенклатура ИЗ Документ." +
ДокументМетаданные.Имя + "." + СокрЛП(ИмяТабличнойЧасти) + " ГДЕ Ссылка =
&ДокументСсылка)) КАК ПроектыНоменклатуры |ПО
Док.Номенклатура=ПроектыНоменклатуры.НоменклатураПроекта";
Запрос.УстановитьПараметр("ДатаДок",ДокументОбъект.Дата); КонецЕсли;
КонецЕсли; Запрос.Текст = "ВЫБРАТЬ | Док.НомерСтроки " + ТекстЗапроса + "
| ИЗ | Документ." + ДокументМетаданные.Имя + "." + СокрЛП(ИмяТабличнойЧасти)
+ " КАК Док"+ТекстСоединение+ " | ГДЕ Док.Ссылка = &ДокументСсылка"; //
Установим параметры запроса. Запрос.УстановитьПараметр("ДокументСсылка",
ДокументОбъект.Ссылка); Если ДокументОбъект[ИмяТабличнойЧасти].Количество()
= 0 Тогда Запрос.Текст = СтрЗаменить(Запрос.Текст, "ГДЕ Док.Ссылка =
&ДокументСсылка", "ГДЕ ЛОЖЬ"); КонецЕсли; Возврат Запрос.Выполнить();
КонецФункции // СформироватьЗапросПоТабличнойЧасти() // Выгружает результат
запроса в табличную часть, добавляет ей необходимые колонки для проведения. // //
Параметры: // РезультатЗапросаПоТоварам - результат запроса по табличной части
"Товары", // СтруктураШапкиДокумента - выборка по результату запроса по шапке
документа. // // Возвращаемое значение: // Сформированная таблица значений. //
Функция мПодготовитьТаблицу(РезультатЗапроса) Таблица =
РезультатЗапроса.Выгрузить(); Возврат Таблица; КонецФункции функция
ПоказатьТаблицу(Табл) Количество = Табл.Колонки.Количество(); Для каждого
Строка из Табл Цикл Для Индекс = 0 По Количество-1 Цикл

```

Сканируем табличную часть документа 1с

Автор: А.Волос
19.02.2014 06:32

```
Сообщить(СокрЛП(Индекс)+" "+СокрЛП(Табл.Колонки[Индекс].Имя)+ ": " +  
Строка[СокрЛП(Табл.Колонки[Индекс].Имя)];    КонецЦикла;    КонецЦикла;  
КонецФункции
```

{/codecitation}

Скачать файл внешней обработки 1с "Сканируем табличную часть":

[Скачать](#)