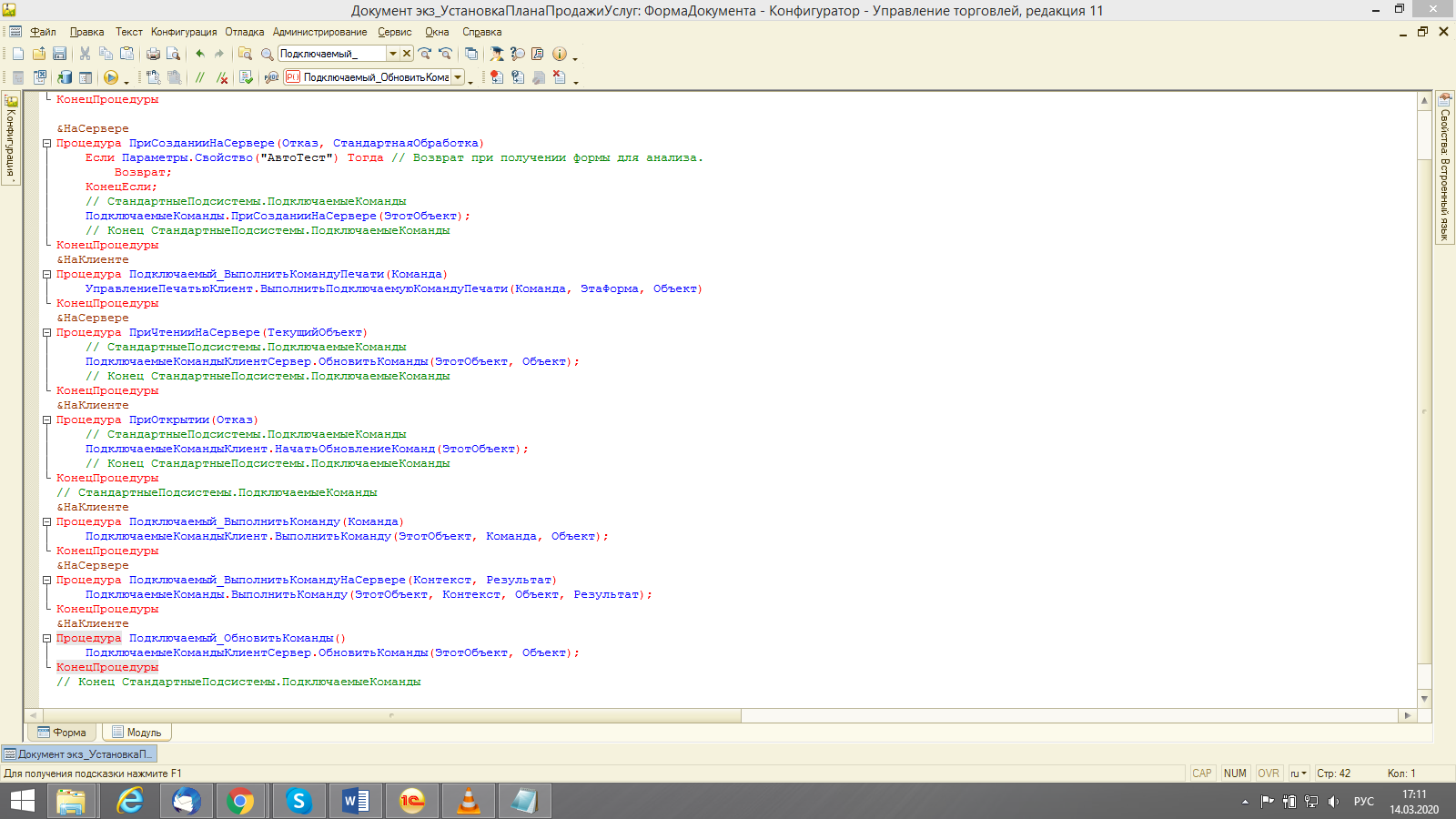
Добавляем регистр накопления экз\_ОбъемДоступныхУслуг и относим его к подсистеме Продажи – Оптовые продажи.

Добавляем измерения: Организация, Номенклатура, Месяц (дата). Добавляем ресурс: Количество (число 15.3). Добавить документ экз\_УстановкаПланаПродажиУслуг относим к Продажи – Оптовые продажи и реквизиты: Организация, Месяц, Ответственный (Пользователь), Комментарий. Добавляем табличную часть Товары: Номенклатура, Количество. Для реквизита документа Номенклатура задаем параметры выбора, чтобы можно было выбрать только услуги (Отбор.ТипНоменклатуры значение Перечисления.ТипыНоменклатуры.Услуга). Движения - регистр накопления экз\_ОбъемДоступныхУслуг. Добавляем Форму документа – Готово. Добавляем Группу Страницы: ГруппаСтраницы в эту группу добавляем Группу Страница: ГруппаОсновное (заголовок Основное) и Группу Страница: ГруппаТовары (заголовок Товары). Переносим в ГруппаТовары – Товары (для ГруппаТовары – Свойства – ПутьКДаннымЗаголовка – Объект.Товары.КоличествоСтрок). В ГруппаОсновное добавляем обычную группу без отображения: ГруппаШапка. В ГруппаШапка добавить еще обычные группы без отображения: ГруппаШапкаЛево и ГруппаШапкаПраво. В ГруппаШапкаЛево добавить обычную группу без отображения: ГруппаНомерДата и перенести в нее Номер и Дату (заголовок «от»). Перенести основные реквизиты в группы. Комментарий (многострочный режим). Добавляем ПриИзмененииМесяц: Объект.Месяц = НачалоМесяца(Объект.Месяц);

Подключаем документ с подсистемам БСП: Подменю Отчеты и Печать.

Для командной панели Формы документа отключаем Автозаполнение и копируем из реализации 3 кнопки: ФормаПровестиИЗакрыть, ФормаЗаписать, ФормаПровести (дозаполняем ИмяКоманд из Стандартные команды). Скопировать ФормаГруппаКомандыФормы, ПодменюОтчеты (проверить Источник команд пустой).

Добавить модуль формы документа (можно скопировать из Реализации по поиску СтандартныеПодсистемы):



/С ЗапуститьОбновлениеИнформационнойБазы

В общем модуле УправлениеПечатьюПереопределяемый. ПриОпределенииОбъектовСКомандамиПечати в УправлениеПечатьюУТСервер.ОпределитьОбъектыСКомандамиПечати добавляем

//Экз

СписокОбъектов.Добавить(Документы.экз\_УстановкаПланаПродажиУслуг);

//Экз.

Копируем из реализации ПодменюПечать и

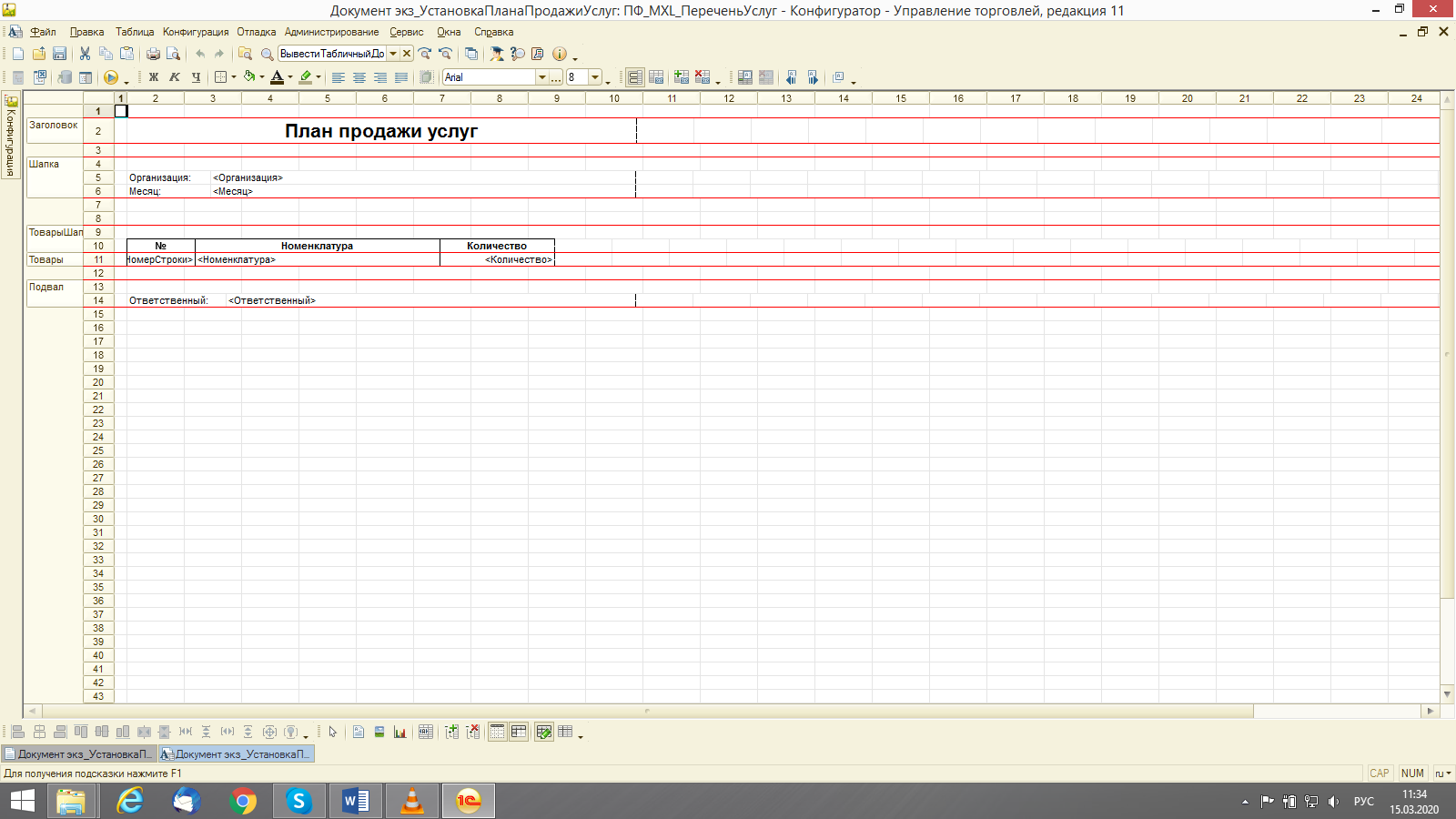
&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуПечати(Команда)

УправлениеПечатьюКлиент.ВыполнитьПодключаемуюКомандуПечати(Команда, ЭтаФорма, Объект)

КонецПроцедуры

С помощью конструктора печати создаем печатную форму (Наименование должно быть вида ПФ\_MXL\_ПереченьУслуг).



В модуле менеджера размещаем код для печати:

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или ВнешнееСоединение Тогда

#Область ПрограммныйИнтерфейс

#Область ДляВызоваИзДругихПодсистем

#Область Отчеты

Процедура ДобавитьКомандыОтчетов(КомандыОтчетов, Параметры) Экспорт

ВариантыОтчетовУТПереопределяемый.ДобавитьКомандуСтруктураПодчиненности(КомандыОтчетов);

КонецПроцедуры

#КонецОбласти

Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.Идентификатор = "ПереченьУслуг";

КомандаПечати.Представление = НСтр("ru = 'Перечень услуг'");

КомандаПечати.ПроверкаПроведенияПередПечатью = Истина;

КонецПроцедуры

Процедура Печать(МассивОбъектов, ПараметрыПечати, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

Если УправлениеПечатью.НужноПечататьМакет(КоллекцияПечатныхФорм, "ПереченьУслуг") Тогда

УправлениеПечатью.ВывестиТабличныйДокументВКоллекцию(

КоллекцияПечатныхФорм,

"ПереченьУслуг",

НСтр("ru = 'ЗПеречень услуг'"),

ПечатнаяФормаПлана(МассивОбъектов, ОбъектыПечати));

КонецЕсли

КонецПроцедуры

#КонецОбласти

Функция ПечатнаяФормаПлана(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

ТабличныйДокумент.КлючПараметровПечати = "Печать\_ПереченьУслуг";

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| экз\_УстановкаПланаПродажиУслуг.Ссылка КАК Ссылка,

| экз\_УстановкаПланаПродажиУслуг.Месяц КАК Месяц,

| экз\_УстановкаПланаПродажиУслуг.Организация КАК Организация,

| экз\_УстановкаПланаПродажиУслуг.Ответственный КАК Ответственный,

| экз\_УстановкаПланаПродажиУслуг.Товары.(

| НомерСтроки КАК НомерСтроки,

| Номенклатура КАК Номенклатура,

| Количество КАК Количество

| ) КАК Товары

|ИЗ

| Документ.экз\_УстановкаПланаПродажиУслуг КАК экз\_УстановкаПланаПродажиУслуг

|ГДЕ

| экз\_УстановкаПланаПродажиУслуг.Ссылка В(&МассивОбъектов)";

Запрос.Параметры.Вставить("МассивОбъектов", МассивОбъектов);

Шапка = Запрос.Выполнить().Выбрать();

Макет = УправлениеПечатью.МакетПечатнойФормы("Документ.экз\_УстановкаПланаПродажиУслуг.ПФ\_MXL\_ПереченьУслуг");

Пока Шапка.Следующий() Цикл

Если ТабличныйДокумент.ВысотаТаблицы > 0 Тогда

ТабличныйДокумент.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

НомерСтрокиНачало = ТабличныйДокумент.ВысотаТаблицы + 1;

ДанныеПечати = Новый Структура;

ДанныеПечати.Вставить("Организация", Шапка.Организация);

ДанныеПечати.Вставить("Месяц", Шапка.Месяц);

ТаблицаТовары = Шапка.Товары.Выгрузить();

МассивОбластейМакета = Новый Массив;

МассивОбластейМакета.Добавить("Заголовок");

МассивОбластейМакета.Добавить("Шапка");

МассивОбластейМакета.Добавить("ТоварыШапка");

МассивОбластейМакета.Добавить("Товары");

МассивОбластейМакета.Добавить("Подвал");

Для Каждого ИмяОбласти Из МассивОбластейМакета Цикл

ОбластьМакета = Макет.ПолучитьОбласть(ИмяОбласти);

Если ИмяОбласти <> "Товары" Тогда

ЗаполнитьЗначенияСвойств(ОбластьМакета.Параметры, ДанныеПечати);

ТабличныйДокумент.Вывести(ОбластьМакета);

Иначе

Для Каждого СтрокаТаблицы Из ТаблицаТовары Цикл

ОбластьМакета.Параметры.Заполнить(СтрокаТаблицы);

ТабличныйДокумент.Вывести(ОбластьМакета);

КонецЦикла;

КонецЕсли;

КонецЦикла;

УправлениеПечатью.ЗадатьОбластьПечатиДокумента(ТабличныйДокумент, НомерСтрокиНачало, ОбъектыПечати, Шапка.Ссылка);

КонецЦикла;

Возврат ТабличныйДокумент;

КонецФункции

#КонецОбласти

#КонецЕсли

Модуль объекта документа. Для образца можно взять модуль объекта документа УстановкаКвотАссортимента. Добавляем процедуры Обработчики событий.

Проведение документа и удаление проведения делаем как у УстановкаКвотАссортимента. Модуль менеджера:

Функция СуществуетПлан(Организация, НаДату, ВызывающийДокумент) Экспорт

ЕстьПлан = Ложь;

Запрос = Новый Запрос("ВЫБРАТЬ ПЕРВЫЕ 1

| экз\_ОбъемДоступныхУслуг.Регистратор КАК Регистратор

|ИЗ

| РегистрНакопления.экз\_ОбъемДоступныхУслуг КАК экз\_ОбъемДоступныхУслуг

|ГДЕ

| экз\_ОбъемДоступныхУслуг.Месяц = &Период

| И экз\_ОбъемДоступныхУслуг.Организация = &Организация");

Запрос.УстановитьПараметр("Период", НачалоМесяца(НаДату));

Запрос.УстановитьПараметр("Организация", Организация);

Выборка = Запрос.Выполнить().Выбрать();

Пока Выборка.Следующий() Цикл

Если Выборка.Регистратор <> ВызывающийДокумент Тогда

ЕстьПлан = Истина;

КонецЕсли;

КонецЦикла;

Возврат ЕстьПлан;

КонецФункции

#Область Проведение

// Инициализирует таблицы значений, содержащие данные табличных частей документа.

// Таблицы значений сохраняет в свойствах структуры "ДополнительныеСвойства".

Функция ДополнительныеИсточникиДанныхДляДвижений(ИмяРегистра) Экспорт

ИсточникиДанных = Новый Соответствие;

Возврат ИсточникиДанных;

КонецФункции

Процедура ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка)

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.УстановитьПараметр("Ссылка", ДокументСсылка);

КонецПроцедуры

Процедура ИнициализироватьДанныеДокумента(ДокументСсылка, ДополнительныеСвойства, Регистры = Неопределено) Экспорт

////////////////////////////////////////////////////////////////////////////

// Создадим запрос инициализации движений

Запрос = Новый Запрос;

ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка);

////////////////////////////////////////////////////////////////////////////

// Сформируем текст запроса

ТекстыЗапроса = Новый СписокЗначений;

ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры);

ПроведениеСерверУТ.ИнициализироватьТаблицыДляДвижений(Запрос, ТекстыЗапроса, ДополнительныеСвойства.ТаблицыДляДвижений, Истина);

КонецПроцедуры

Функция ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экз\_ОбъемДоступныхУслуг";

Если НЕ ПроведениеСерверУТ.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| экз\_УстановкаПланаПродажиУслугТовары.НомерСтроки КАК НСтр,

| экз\_УстановкаПланаПродажиУслугТовары.Ссылка.Дата КАК Период,

| экз\_УстановкаПланаПродажиУслугТовары.Номенклатура КАК Номенклатура,

| экз\_УстановкаПланаПродажиУслугТовары.Ссылка.Организация КАК Организация,

| экз\_УстановкаПланаПродажиУслугТовары.Ссылка.Месяц КАК Месяц,

| экз\_УстановкаПланаПродажиУслугТовары.Количество КАК Количество,

| ЗНАЧЕНИЕ(ВидДвиженияНакопления.Приход) КАК ВидДвижения

|ИЗ

| Документ.экз\_УстановкаПланаПродажиУслуг.Товары КАК экз\_УстановкаПланаПродажиУслугТовары

|ГДЕ

| экз\_УстановкаПланаПродажиУслугТовары.Ссылка = &Ссылка

|

|УПОРЯДОЧИТЬ ПО

| НСтр";

ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

КонецФункции

#КонецОбласти

Модуль объекта:

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или ВнешнееСоединение Тогда

#Область ОбработчикиСобытий

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)

Если ЭтоНовый() Тогда

ИнициализироватьДокумент(ДанныеЗаполнения);

КонецЕсли;

КонецПроцедуры

Процедура ПриКопировании(ОбъектКопирования)

Ответственный = Пользователи.ТекущийПользователь();

ДатаНачалаДействия = НачалоМесяца(ТекущаяДатаСеанса());

Если ЗначениеЗаполнено(ОбъектПланирования) Тогда

Если Документы.УстановкаКвотАссортимента.СуществуетПлан(ОбъектПланирования, ДатаНачалаДействия, Ссылка) Тогда

ДатаНачалаДействия = Дата(1,1,1);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

КвотаВсего = ТоварныеКатегории.Итог("Квота");

ПроведениеСерверУТ.УстановитьРежимПроведения(ЭтотОбъект, РежимЗаписи, РежимПроведения);

ДополнительныеСвойства.Вставить("ЭтоНовый", ЭтоНовый());

ДополнительныеСвойства.Вставить("РежимЗаписи", РежимЗаписи);

Если РежимЗаписи = РежимЗаписиДокумента.Проведение Тогда

ЕстьПлан = Документы.УстановкаКвотАссортимента.СуществуетПлан(ОбъектПланирования, ДатаНачалаДействия, Ссылка);

Если ЕстьПлан Тогда

Отказ = Истина;

ТекстСообщения = НСтр("ru = 'На %1 для выбранного формата уже существует документ установки квот.'")

+ Символы.ПС + НСтр("ru = 'Установка квот может производиться не чаще 1 раза в месяц.'")

+ Символы.ПС + НСтр("ru = 'Выберите другую дату или формат магазинов.'");

ТекстСообщения = СтроковыеФункцииКлиентСервер.ПодставитьПараметрыВСтроку(

ТекстСообщения, Формат(ДатаНачалаДействия,"ДЛФ=D"));

ОбщегоНазначенияКлиентСервер.СообщитьПользователю(

ТекстСообщения,

,

,

,

Истина);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Процедура ПриЗаписи(Отказ)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

ПроведениеСерверУТ.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства, РежимПроведения);

Документы.УстановкаКвотАссортимента.ИнициализироватьДанныеДокумента(Ссылка, ДополнительныеСвойства);

ПроведениеСерверУТ.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

АссортиментСервер.ОтразитьКвотыАссортимента(ДополнительныеСвойства, Движения, Отказ);

ПроведениеСерверУТ.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСерверУТ.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

Процедура ОбработкаУдаленияПроведения(Отказ)

ПроведениеСерверУТ.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства);

ПроведениеСерверУТ.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

ПроведениеСерверУТ.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСерверУТ.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

#Область ИнициализацияИЗаполнение

// Инициализирует документ

//

Процедура ИнициализироватьДокумент(ДанныеЗаполнения = Неопределено)

Ответственный = Пользователи.ТекущийПользователь();

ДатаНачалаДействия = НачалоМесяца(ТекущаяДатаСеанса());

ОбъектПланирования = ЗначениеНастроекПовтИсп.ПолучитьОбъектПланированияПоУмолчанию(ОбъектПланирования);

Если ЗначениеЗаполнено(ОбъектПланирования) Тогда

Если Документы.УстановкаКвотАссортимента.СуществуетПлан(ОбъектПланирования, ДатаНачалаДействия, Ссылка) Тогда

ДатаНачалаДействия = Дата(1,1,1);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

#КонецОбласти

#КонецОбласти

#КонецЕсли

Делаем движения по новому регистру у Реализации товаров и услуг. В процедуре Документы.РеализацияТоваровУслуг.ИнициализироватьДанныеДокумента добавит свою процедуру по аналогии с ТекстЗапросаТаблицаТоварыНаСкладах(Запрос, ТекстыЗапроса, Регистры) (2-й запрос можно удалить и убрать ненужные поля)

//Экз

ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры);

Функция ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экз\_ОбъемДоступныхУслуг";

Если НЕ ПроведениеСерверУТ.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| ЗНАЧЕНИЕ(ВидДвиженияНакопления.Расход) КАК ВидДвижения,

| &Период КАК Период,

| ТаблицаТовары.Номенклатура КАК Номенклатура,

| ТаблицаТовары.Количество КАК Количество,

| ТаблицаТовары.Ссылка.Организация КАК Организация,

| НАЧАЛОПЕРИОДА(&Период, МЕСЯЦ) КАК Месяц

|ИЗ

| Документ.РеализацияТоваровУслуг.Товары КАК ТаблицаТовары

|ГДЕ

| ТаблицаТовары.Ссылка = &Ссылка

| И ТаблицаТовары.Количество <> 0

| И ТаблицаТовары.Номенклатура.ТипНоменклатуры = ЗНАЧЕНИЕ(Перечисление.ТипыНоменклатуры.Услуга)

| И &ХозяйственнаяОперация <> ЗНАЧЕНИЕ(Перечисление.ХозяйственныеОперации.РеализацияКлиентуРеглУчет)";

ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

КонецФункции

//Экз

Добавляем процедуру ЗапасыСервер.ОтразитьОбъемДоступныхУслуг(ДополнительныеСвойства, Движения, Отказ) по аналогии с ЗапасыСервер.ОтразитьТоварыНаСкладах(ДополнительныеСвойства, Движения, Отказ).

//Экз

ЗапасыСервер.ОтразитьОбъемДоступныхУслуг(ДополнительныеСвойства, Движения, Отказ);

//Экз

//Экз

Процедура ОтразитьОбъемДоступныхУслуг(ДополнительныеСвойства, Движения, Отказ) Экспорт

Таблица= ДополнительныеСвойства.ТаблицыДляДвижений.Таблицаэкз\_ОбъемДоступныхУслуг;

Если Отказ ИЛИ Таблица.Количество() = 0 Тогда

Возврат;

КонецЕсли;

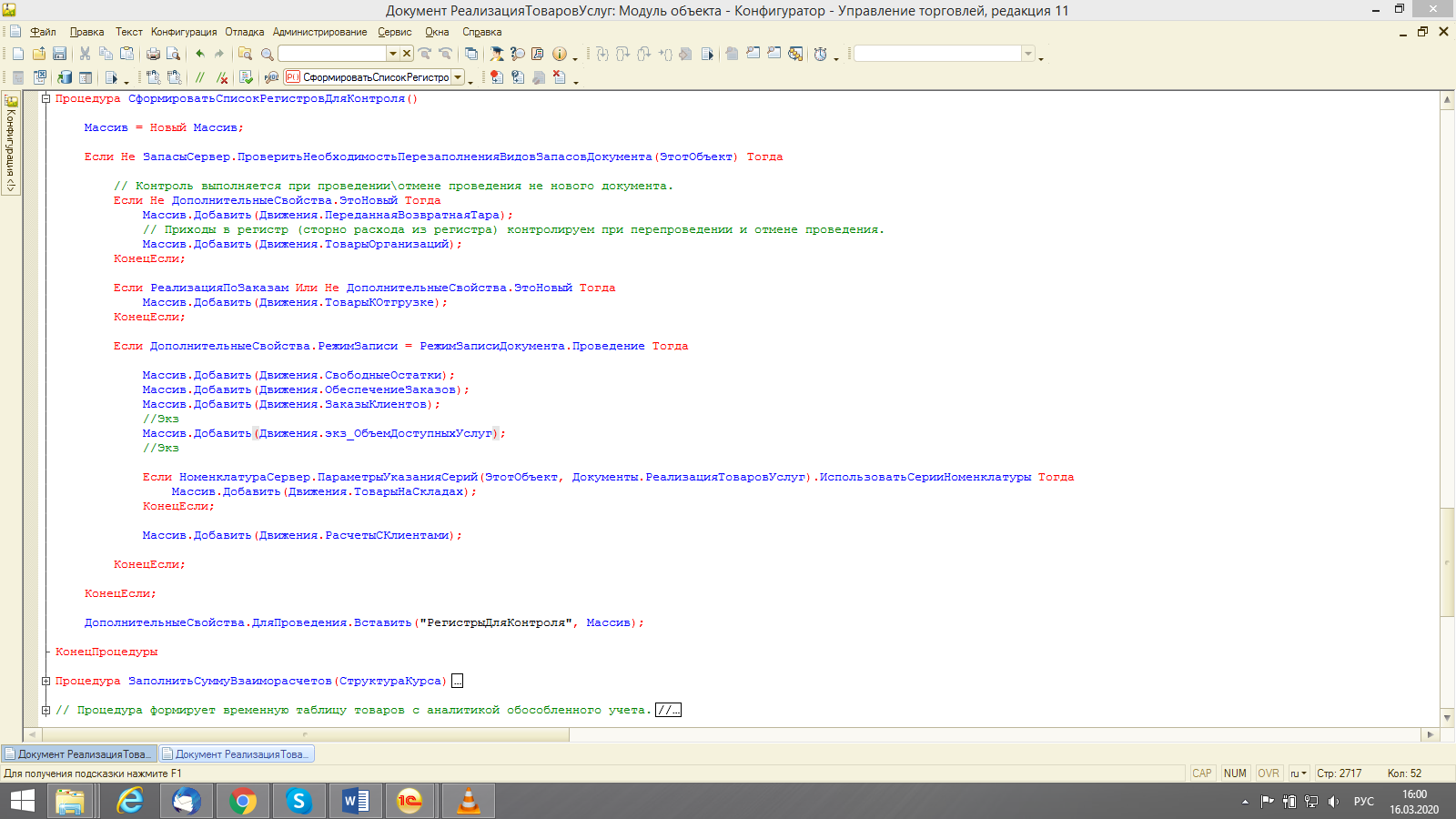
Движения.экз\_ОбъемДоступныхУслуг.Записывать = Истина;

Движения.экз\_ОбъемДоступныхУслуг.Загрузить(Таблица);

КонецПроцедуры

//Экз

Дорабатываем контроль проведения: СформироватьСписокРегистровДляКонтроля()



ПроведениеСерверУТ.ВыполнитьКонтрольРезультатовПроведения(ЭтотОбъект, Отказ)

Для решения задачи лучше использовать оборотный регистр с 2 ресурсами (план и факт), т.к. остаточный регистр может «не закрываться в ноль» (необходимо отдельное закрытие регистра).