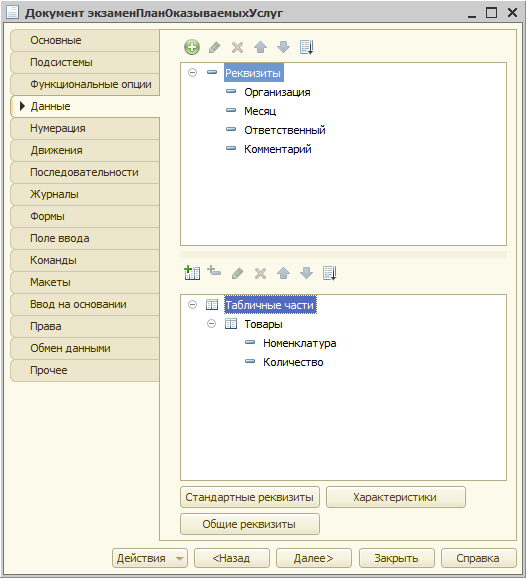
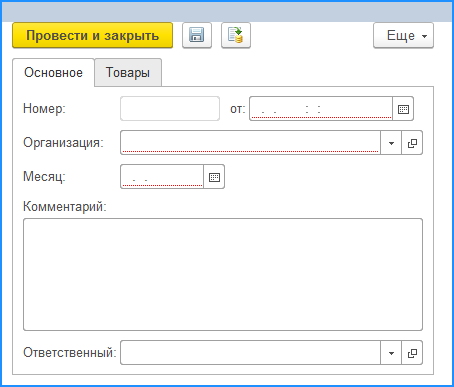
Создаем документ экзаменПланОказываемыхУслуг, включаем в подсистему Планирование, запрещаем оперативное проведение, выключаем автоматическое удаление движений при отмене проведения.



Создаем форму



Подключим документ к подсистеме БСП печать и отчеты. Модуль формы

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// СтандартныеПодсистемы.Печать

УправлениеПечатью.ПриСозданииНаСервере(ЭтаФорма, Элементы.ПодменюПечать);

// Конец СтандартныеПодсистемы.Печать

МенюОтчеты.ПриСозданииНаСервере(ЭтаФорма, Элементы.ПодменюОтчеты);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуПечати(Команда)

УправлениеПечатьюКлиент.ВыполнитьПодключаемуюКомандуПечати(Команда, ЭтаФорма, Объект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.Печать

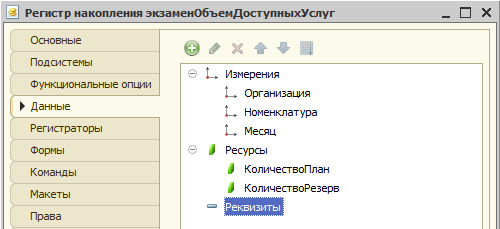
&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуОтчет(Команда)

МенюОтчетыКлиент.ВыполнитьПодключаемуюКомандуОтчет(Команда, ЭтаФорма, Объект);

КонецПроцедуры

Создадим оборотный регистр накопления экзаменОбъемДоступныхУслуг. Если использовать остаточный регистр, то нужно будет предусмотреть алгоритм сворачивания итогов регистра в 0, так как движения прихода будут каждый месяц, а расхода может и не быть.



## Алгоритм проведения документа экзаменПланОказываемыхУслуг по РН экзаменОбъемДоступныхУслуг

Модуль объекта документа

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, ТекстЗаполнения, СтандартнаяОбработка)

Организация = ЗначениеНастроекПовтИсп.ПолучитьОрганизациюПоУмолчанию();

Ответственный = Пользователи.ТекущийПользователь();

Если НЕ Документы.экзаменПланОказываемыхУслуг.СуществуетПлан(Организация, ТекущаяДатаСеанса(), Ссылка) Тогда

Месяц = НачалоМесяца(ТекущаяДатаСеанса());

КонецЕсли;

КонецПроцедуры

Процедура ПриКопировании(ОбъектКопирования)

Ответственный = Пользователи.ТекущийПользователь();

Месяц = Дата(1,1,1);

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

ПроведениеСервер.УстановитьРежимПроведения(ЭтотОбъект, РежимЗаписи, РежимПроведения);

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

ДополнительныеСвойства.Вставить("РежимЗаписи", РежимЗаписи);

Если РежимЗаписи = РежимЗаписиДокумента.Проведение Тогда

ЕстьПлан = Документы.экзаменПланОказываемыхУслуг.СуществуетПлан(Организация, Месяц, Ссылка);

Если ЕстьПлан Тогда

Отказ = Истина;

ТекстСообщения = НСтр("ru = 'На %1 для выбранной организации уже существует документ установки плана.'")

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

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

ТекстСообщения = СтроковыеФункцииКлиентСервер.ПодставитьПараметрыВСтроку(

ТекстСообщения,

Формат(Месяц,"ДФ='MMMM yyyy ""г.""'"));

ОбщегоНазначенияКлиентСервер.СообщитьПользователю(

ТекстСообщения,

,

,

,

Отказ);

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

ПроведениеСервер.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства, РежимПроведения);

Документы.экзаменПланОказываемыхУслуг.ИнициализироватьДанныеДокумента(Ссылка, ДополнительныеСвойства);

ПроведениеСервер.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

ОтразитьОбъемДоступныхУслуг();

ПроведениеСервер.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСервер.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

Процедура ОбработкаУдаленияПроведения(Отказ)

ПроведениеСервер.ИнициализироватьДополнительныеСвойстваДляПроведения(Ссылка, ДополнительныеСвойства);

ПроведениеСервер.ПодготовитьНаборыЗаписейКРегистрацииДвижений(ЭтотОбъект);

ПроведениеСервер.ЗаписатьНаборыЗаписей(ЭтотОбъект);

ПроведениеСервер.ОчиститьДополнительныеСвойстваДляПроведения(ДополнительныеСвойства);

КонецПроцедуры

Процедура ПриУстановкеНовогоНомера(СтандартнаяОбработка, Префикс)

ПрефиксацияОбъектовСобытия.УстановитьПрефиксИнформационнойБазыИОрганизации(ЭтотОбъект, СтандартнаяОбработка, Префикс);

КонецПроцедуры

Процедура ОтразитьОбъемДоступныхУслуг() Экспорт

Таблица = ДополнительныеСвойства.ТаблицыДляДвижений.ТаблицаэкзаменОбъемДоступныхУслуг;

Если Таблица.Количество() = 0 Тогда

Возврат;

КонецЕсли;

ДвиженияОбъемДоступныхУслуг = Движения.экзаменОбъемДоступныхУслуг;

ДвиженияОбъемДоступныхУслуг.Записывать = Истина;

ДвиженияОбъемДоступныхУслуг.Загрузить(Таблица);

КонецПроцедуры

Модуль менеджера документа

#Область Отчеты

Процедура ДобавитьКомандыОтчетов(КомандыОтчетов) Экспорт

ВариантыОтчетовУТПереопределяемый.ДобавитьКомандуДвиженияДокумента(КомандыОтчетов);

КонецПроцедуры

#КонецОбласти

#Область Печать

Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.Идентификатор = "СвободныеУслуги";

КомандаПечати.Представление = НСтр("ru = 'Свободные услуги'");

КомандаПечати.ПроверкаПроведенияПередПечатью = Истина;

КонецПроцедуры

#КонецОбласти

#Область Проведение

Процедура ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка)

Запрос.МенеджерВременныхТаблиц = Новый МенеджерВременныхТаблиц;

Запрос.УстановитьПараметр("Ссылка", ДокументСсылка);

Запрос.УстановитьПараметр("Период", ОбщегоНазначения.ЗначениеРеквизитаОбъекта(ДокументСсылка, "Дата"));

КонецПроцедуры

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

Запрос = Новый Запрос;

ЗаполнитьПараметрыИнициализации(Запрос, ДокументСсылка);

ТекстыЗапроса = Новый СписокЗначений;

ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры);

ПроведениеСервер.ИницализироватьТаблицыДляДвижений(Запрос, ТекстыЗапроса, ДополнительныеСвойства.ТаблицыДляДвижений, Истина);

КонецПроцедуры

Функция ТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экзаменОбъемДоступныхУслуг";

Если НЕ ПроведениеСервер.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| экзаменПланОказываемыхУслугТовары.НомерСтроки КАК НСтр,

| &Период,

| экзаменПланОказываемыхУслугТовары.Ссылка.Организация,

| экзаменПланОказываемыхУслугТовары.Ссылка.Месяц,

| экзаменПланОказываемыхУслугТовары.Номенклатура,

| экзаменПланОказываемыхУслугТовары.Количество КАК КоличествоПлан

|ИЗ

| Документ.экзаменПланОказываемыхУслуг.Товары КАК экзаменПланОказываемыхУслугТовары

|ГДЕ

| экзаменПланОказываемыхУслугТовары.Ссылка = &Ссылка

|

|УПОРЯДОЧИТЬ ПО

| НСтр";

ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

КонецФункции

#КонецОбласти

#Область Прочее

Функция СуществуетПлан(Организация, НаДату, ВызывающийДокумент) Экспорт

#КонецОбласти

ЕстьПлан = Ложь;

Запрос = Новый Запрос("ВЫБРАТЬ ПЕРВЫЕ 1

| экзаменОбъемДоступныхУслуг.Регистратор КАК Регистратор

|ИЗ

| РегистрНакопления.экзаменОбъемДоступныхУслуг КАК экзаменОбъемДоступныхУслуг

|ГДЕ

| экзаменОбъемДоступныхУслуг.Месяц = &Период

| И экзаменОбъемДоступныхУслуг.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;

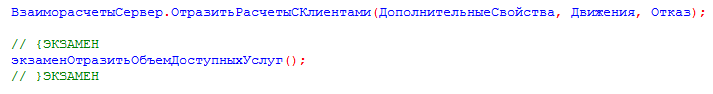
КонецЦикла;

Возврат ЕстьПлан;

КонецФункции

## Алгоритм проведения документа ЗаказКлиента по РН экзаменОбъемДоступныхУслуг

Модуль объекта, процедура ОбработкаПроведения

 #Область Экзамен

Процедура экзаменОтразитьОбъемДоступныхУслуг()

Таблица = ДополнительныеСвойства.ТаблицыДляДвижений.ТаблицаэкзаменОбъемДоступныхУслуг;

Если Таблица.Количество() = 0 Тогда

Возврат;

КонецЕсли;

НаборЗаписей = Движения.экзаменОбъемДоступныхУслуг;

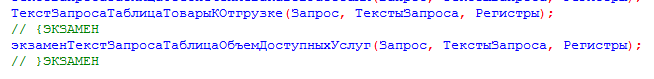
НаборЗаписей.Записывать = Истина;

НаборЗаписей.Загрузить(Таблица);

КонецПроцедуры

#КонецОбласти

Модуль менеджера, процедура ИнициализироватьДанныеДокумента

**

#Область Экзамен

Функция экзаменТекстЗапросаТаблицаОбъемДоступныхУслуг(Запрос, ТекстыЗапроса, Регистры)

ИмяРегистра = "экзаменОбъемДоступныхУслуг";

Если НЕ ПроведениеСервер.ТребуетсяТаблицаДляДвижений(ИмяРегистра, Регистры) Тогда

Возврат "";

КонецЕсли;

ТекстЗапроса =

"ВЫБРАТЬ

| ЗаказКлиентаТовары.НомерСтроки КАК НСтр,

| &ПЕРИОД КАК Период,

| НАЧАЛОПЕРИОДА(&ДатаОтгрузки, МЕСЯЦ) КАК Месяц,

| &Организация КАК Организация,

| ЗаказКлиентаТовары.Номенклатура КАК Номенклатура ,

| ЗаказКлиентаТовары.Количество КАК КоличествоРезерв

|ИЗ

| Документ.ЗаказКлиента.Товары КАК ЗаказКлиентаТовары

|ГДЕ

| ЗаказКлиентаТовары.Ссылка = &Ссылка

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

|

|УПОРЯДОЧИТЬ ПО

| НСтр";

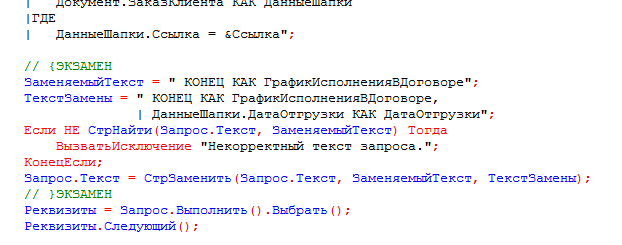
ТекстыЗапроса.Добавить(ТекстЗапроса, ИмяРегистра);

Возврат ТекстЗапроса;

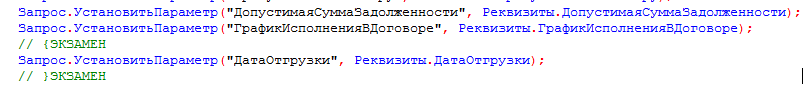
КонецФункции

#КонецОбласти

Процедура ЗаполнитьПараметрыИнициализации

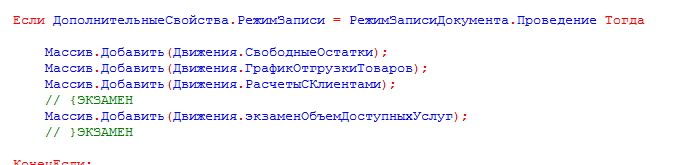


Далее в этой же процедуре



Контроль объема доступных услуг

В процедуре СформироватьСписокРегистровДляКонтроля



ОМ ПроведениеСервер, процедура ВыполнитьКонтрольРезультатовПроведения

// {ЭКЗАМЕН

Если ЕстьИзмененияВТаблице(ДанныеТаблиц,"ДвиженияэкзаменОбъемДоступныхУслугИзменение") Тогда

МассивКонтролей.Добавить(Врег("экзаменОбъемДоступныхУслуг"));

ТекстЗапроса = ТекстЗапроса + "ВЫБРАТЬ

| Таблица.Организация,

| Таблица.Номенклатура,

| Таблица.Месяц,

| Таблица.КоличествоПланОборот - Таблица.КоличествоРезервОборот - Таблица.КоличествоФактОборот КАК Количество

|ИЗ

| РегистрНакопления.экзаменОбъемДоступныхУслуг.Обороты(

| ,

| ,

| ,

| (Организация, Номенклатура, Месяц) В

| (ВЫБРАТЬ

| Товары.Организация,

| Товары.Номенклатура,

| Товары.Месяц

| ИЗ

| ДвиженияэкзаменОбъемДоступныхУслугИзменение КАК Товары)) КАК Таблица

|ГДЕ

| Таблица.КоличествоПланОборот - Таблица.КоличествоРезервОборот - Таблица.КоличествоФактОборот < 0;

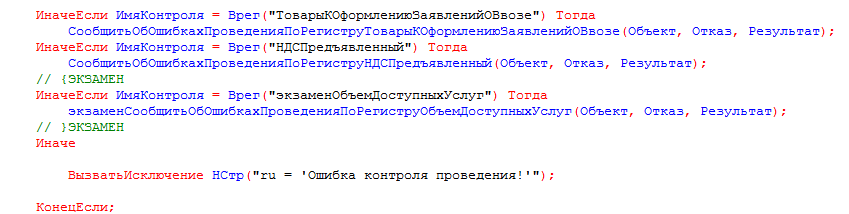
|///////////////////////////////////////////////////////////////////

|";

КонецЕсли;

// }ЭКЗАМЕН

Далее в этой же процедуре

**

Модуль набора записей РН экзаменОбъемДоступныхУслуг

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или ВнешнееСоединение Тогда

#Область ОбработчикиСобытий

Процедура ПередЗаписью(Отказ, Замещение)

Если ОбменДанными.Загрузка Или Не ПроведениеСервер.РассчитыватьИзменения(ДополнительныеСвойства) Тогда

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

ТребуетсяКонтроль = Истина;

СтруктураВременныеТаблицы = ДополнительныеСвойства.ДляПроведения.СтруктураВременныеТаблицы;

БлокироватьДляИзменения = Истина;

// Текущее состояние набора помещается во временную таблицу "ДвиженияэкзаменОбъемДоступныхУслугПередЗаписью",

// чтобы при записи получить изменение нового набора относительно текущего.

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Регистратор", Отбор.Регистратор.Значение);

Запрос.УстановитьПараметр("ЭтоНовый", ДополнительныеСвойства.ЭтоНовый);

Запрос.МенеджерВременныхТаблиц = СтруктураВременныеТаблицы.МенеджерВременныхТаблиц;

Запрос.Текст =

"ВЫБРАТЬ

| Таблица.Организация КАК Организация,

| Таблица.Номенклатура КАК Номенклатура,

| Таблица.Месяц КАК Месяц,

| Таблица.КоличествоРезерв КАК КоличествоРезервПередЗаписью,

| Таблица.КоличествоПлан КАК КоличествоПланПередЗаписью

|ПОМЕСТИТЬ ДвиженияэкзаменОбъемДоступныхУслугПередЗаписью

|ИЗ

| РегистрНакопления.экзаменОбъемДоступныхУслуг КАК Таблица

|ГДЕ

| Таблица.Регистратор = &Регистратор

| И НЕ &ЭтоНовый";

Запрос.Выполнить();

КонецПроцедуры

Процедура ПриЗаписи(Отказ, Замещение)

Если ОбменДанными.Загрузка Или Не ПроведениеСервер.РассчитыватьИзменения(ДополнительныеСвойства) Тогда

Возврат;

КонецЕсли;

СтруктураВременныеТаблицы = ДополнительныеСвойства.ДляПроведения.СтруктураВременныеТаблицы;

// Рассчитывается изменение нового набора относительно текущего с учетом накопленных изменений

// и помещается во временную таблицу.

Запрос = Новый Запрос;

Запрос.УстановитьПараметр("Регистратор", Отбор.Регистратор.Значение);

Запрос.МенеджерВременныхТаблиц = СтруктураВременныеТаблицы.МенеджерВременныхТаблиц;

Запрос.Текст =

"ВЫБРАТЬ

| ТаблицаИзменений.Номенклатура КАК Номенклатура,

| ТаблицаИзменений.Организация КАК Организация,

| ТаблицаИзменений.Месяц КАК Месяц,

| СУММА(ТаблицаИзменений.КоличествоРезервИзменение) КАК КоличествоРезервИзменение,

| СУММА(ТаблицаИзменений.КоличествоПланИзменение) КАК КоличествоПланИзменение

|ПОМЕСТИТЬ ДвиженияэкзаменОбъемДоступныхУслугИзменение

|ИЗ

| (ВЫБРАТЬ

| Таблица.Номенклатура КАК Номенклатура,

| Таблица.Организация КАК Организация,

| Таблица.Месяц КАК Месяц,

| Таблица.КоличествоРезервПередЗаписью КАК КоличествоРезервИзменение,

| Таблица.КоличествоПланПередЗаписью КАК КоличествоПланИзменение

| ИЗ

| ДвиженияэкзаменОбъемДоступныхУслугПередЗаписью КАК Таблица

|

| ОБЪЕДИНИТЬ ВСЕ

|

| ВЫБРАТЬ

| Таблица.Номенклатура КАК Номенклатура,

| Таблица.Организация КАК Организация,

| Таблица.Месяц КАК Месяц,

| -Таблица.КоличествоРезерв КАК КоличествоРезервИзменение,

| -Таблица.КоличествоПлан КАК КоличествоПланИзменение

| ИЗ

| РегистрНакопления.экзаменОбъемДоступныхУслуг КАК Таблица

| ГДЕ

| Таблица.Регистратор = &Регистратор) КАК ТаблицаИзменений

|

|СГРУППИРОВАТЬ ПО

| ТаблицаИзменений.Номенклатура,

| ТаблицаИзменений.Организация,

| ТаблицаИзменений.Месяц

|

|ИМЕЮЩИЕ

| (СУММА(ТаблицаИзменений.КоличествоРезервИзменение) <> 0

| ИЛИ СУММА(ТаблицаИзменений.КоличествоПланИзменение) <> 0)

|;

|

|////////////////////////////////////////////////////////////////////////////////

|УНИЧТОЖИТЬ ДвиженияэкзаменОбъемДоступныхУслугПередЗаписью";

Выборка = Запрос.ВыполнитьПакет()[0].Выбрать();

Выборка.Следующий();

// Добавляется информация о ее существовании и наличии в ней записей об изменении.

СтруктураВременныеТаблицы.Вставить("ДвиженияэкзаменОбъемДоступныхУслугИзменение", Выборка.Количество > 0);

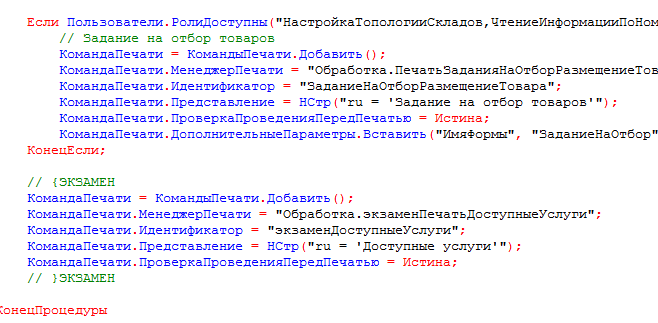
КонецПроцедуры

#КонецОбласти

#КонецЕсли

## Печатная форма

Далее в модуле менеджера ЗаказКлиента, в процедуре ДобавитьКомандыПечати

**

Вставить обработку в кофигурацию