2		КУРС 1С:ПРОГРАММИРОВАНИЕ – ЗА 21 ДЕНЬ! Код решений занятий.

		Курсы по прикладным решениям "1С:Предприятие 8"
1	Отчет по конф E:\Documents and Settings\Admin\Мои документы\ОтчетПоКонфигурации.txt	ДОМАШНЕЕ ЗАДАНИЕ №1 Установите учебную конфигурацию 1С:Бухгалтерия 8 и создайте новую информационную базу на основе шаблона 1С:Бухгалтерии 8. Учебная версия . Выполните произвольные действия в интерфейсе Такси , используя демонстрационную базу, опубликованную в Интернете:
2		ДОМАШНЕЕ ЗАДАНИЕ №2 Организация ООО «КанцОптТорг» занимается продажей канцелярских товаров. Необходимо автоматизировать основные торговые процессы данного предприятия. Организация состоит из одного юридического лица и одного склада (расширение не планируется). Создайте иерархические справочники: • Номенклатура • Контрагенты • Сотрудники Для контрагента необходимо определить тип отношений: • Клиент • Поставщик • Прочее (например, банк или налоговая инспекция) При этом контрагент может иметь одновременно несколько типов отношений (например, является и поставщиком, и покупателем). Для этого в справочнике можно определить несколько реквизитов с типом «Булево». Для каждого контрагента необходимо указывать ответственного менеджера. Создайте новую пустую информационную базу (для разработки) и определите в ней необходимые объекты конфигурации.
3		ДОМАШНЕЕ ЗАДАНИЕ №3 Создайте документ Поступление товаров , который отражает закупку товаров у поставщика. В документе должен быть реквизит Ответственный , который содержит сотрудника, отвечающего за этот документ. Создайте документ Реализация товаров , отражающий продажу товаров. Документ также должен содержать реквизит Ответственный . Обеспечьте ввод реализации на основании поступления товаров. Создайте журнал документов Товарные документы , в котором должны быть графы Контрагент и Ответственный . Сформируйте печатную форму Реализации товаров .

4	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("ДокументСсылка.ЗаказКлиента") Тогда
                // Заполнение шапки
                Клиент = ДанныеЗаполнения.Клиент;
                Склад = ДанныеЗаполнения.Склад;
                Для Каждого ТекСтрокаТовары Из
ДанныеЗаполнения.Товары Цикл
                        НоваяСтрока = Товары.Добавить();
                        НоваяСтрока.Количество =
ТекСтрокаТовары.Количество;
                        НоваяСтрока.Сумма =
ТекСтрокаТовары.Сумма;
                        НоваяСтрока.Товар =
ТекСтрокаТовары.Товар;
                        НоваяСтрока.Цена =
ТекСтрокаТовары.Цена;
                КонецЦикла;
        КонецЕсли;
        //}}__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
КонецПроцедуры
```

5

```
Процедура ПечатьСчетаНаОплату(ТабДок, Ссылка) Экспорт
        //{{_КОНСТРУКТОР_ПЕЧАТИ(ПечатьСчетаНаОплату)
        Макет =
Документы.ЗаказКлиента.ПолучитьМакет("ПечатьСчетаНаОплату");
        Запрос = Новый Запрос;
        Запрос.Текст =
        "ВЫБРАТЬ
        |       ЗаказКлиента.Дата,
        |       ЗаказКлиента.Клиент,
        |       ЗаказКлиента.Номер,
        |       ЗаказКлиента.Склад,
        |       ЗаказКлиента.Товары.(
        |               НомерСтроки,
        |               Товар,
        |               Количество,
        |               Цена,
        |               Сумма
        |       )
        |ИЗ
        |       Документ.ЗаказКлиента КАК ЗаказКлиента
        |ГДЕ
        |       ЗаказКлиента.Ссылка В (&Ссылка)";
        Запрос.Параметры.Вставить("Ссылка", Ссылка);
        Выборка = Запрос.Выполнить().Выбрать();

        ОбластьЗаголовок =
Макет.ПолучитьОбласть("Заголовок");
        Шапка = Макет.ПолучитьОбласть("Шапка");
        ОбластьТоварыШапка =
Макет.ПолучитьОбласть("ТоварыШапка");
        ОбластьТовары = Макет.ПолучитьОбласть("Товары");
        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(ОбластьТоварыШапка);
                ВыборкаТовары = Выборка.Товары.Выбрать();
                Пока ВыборкаТовары.Следующий() Цикл

        ОбластьТовары.Параметры.Заполнить(ВыборкаТовары);
                        ТабДок.Вывести(ОбластьТовары,
ВыборкаТовары.Уровень());
                КонецЦикла;

                Подвал.Параметры.Заполнить(Выборка);
                ТабДок.Вывести(Подвал);

                ВставлятьРазделительСтраниц = Истина;
        КонецЦикла;
        //}}
КонецПроцедуры
```

6

7		**ДОМАШНЕЕ ЗАДАНИЕ №4** Создайте регистр накопления Остатки товаров . Регистр должен содержать информацию об остатках номенклатуры. Обеспечьте проведение документов Реализация товаров и Поступление товаров по этому регистру. Добавьте команду отображения движений на формы документов. Выполните проведение существующих документов в пользовательском режиме.

8	Процедура ОбработкаПроведения(Отказ, Режим)
	//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	// Данный фрагмент построен конструктором.
	// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
	
	// регистр Взаиморасчеты Расход
	Движения.Взаиморасчеты.Записывать = Истина;
	Движение = Движения.Взаиморасчеты.Добавить();
	Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
	Движение.Период = Дата;
	Движение.Контрагент = Поставщик;
	Движение.Сумма = СуммаДокумента;
	
	//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	КонецПроцедуры	

9		**ДОМАШНЕЕ ЗАДАНИЕ №5** Создайте регистр накопления для учета данных о количестве и сумме закупок в разрезе номенклатуры и поставщиков. Перепроведите существующие документы поступления в пользовательском режиме. Создайте независимый регистр для учета цен номенклатуры. Заполните его данными. Создайте 2 подсистемы: Продажи и Закупки. Классифицируйте объекты по этим подсистемам. При этом справочники Номенклатура и Контрагенты должны относится к обоим подсистемам. На начальной странице системы отобразите данные журнала документов Товарные документы .

10	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("ДокументСсылка.ЗаказКлиента") Тогда
	// Заполнение шапки
	Клиент = ДанныеЗаполнения.Клиент;
	Склад = ДанныеЗаполнения.Склад;
	Для Каждого ТекСтрокаТовары Из ДанныеЗаполнения.Товары Цикл
	НоваяСтрока = Товары.Добавить();
	НоваяСтрока.Количество = ТекСтрокаТовары.Количество;
	НоваяСтрока.Сумма = ТекСтрокаТовары.Сумма;
	НоваяСтрока.Товар = ТекСтрокаТовары.Товар;
	НоваяСтрока.Цена = ТекСтрокаТовары.Цена;
	КонецЦикла;
	КонецЕсли;
	//}}__КОНСТРУКТОР_ВВОД_НА_ОСНОВАНИИ
	КонецПроцедуры
	
	Процедура ОбработкаПроведения(Отказ, Режим)
	//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ	

	// Данный фрагмент построен конструктором.
	// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
	
	// регистр Взаиморасчеты Приход
	Движения.Взаиморасчеты.Записывать = Истина;
	Движение = Движения.Взаиморасчеты.Добавить();
	Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
	Движение.Период = Дата;
	Движение.Контрагент = Клиент;
	Движение.Сумма = СуммаДокумента;
	
	// регистр Продажи
	Движения.Продажи.Записывать = Истина;
	Для Каждого ТекСтрокаТовары Из Товары Цикл
	Движение = Движения.Продажи.Добавить();
	Движение.Период = Дата;
	Движение.Клиент = Клиент;
	Движение.Номенклатура = ТекСтрокаТовары.Товар;
	Движение.Количество = ТекСтрокаТовары.Количество;
	Движение.Сумма = ТекСтрокаТовары.Сумма;
	КонецЦикла;
	
	//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	КонецПроцедуры	

11		**ДОМАШНЕЕ ЗАДАНИЕ №6** Создайте переменные – ДвухзначноеЧисло и Степень. Присвойте им определенные целочисленные значения. Обеспечьте возведение ДвухзначногоЧисла в Степень. Далее, нужно сообщить получившийся результат. Также нужно сообщить превышает ли полученное значение 10 000.

12	Перем СчастливоеЧисло;
	
	//Сообщить("Привет, мир!");
	//Сообщить("Hello, world!")
	
	СчастливоеЧисло = 777;
	//Сообщить(СчастливоеЧисло);
	СчастливоеЧисло = 888;
	//Сообщить(СчастливоеЧисло);
	
	//Если СчастливоеЧисло = 777 Тогда
	// Сообщить("Все правильно");
	//ИначеЕсли СчастливоеЧисло = 888 Тогда
	// Сообщить("Восьмерки");
	//Иначе
	// Сообщить("Ошибка");
	//КонецЕсли
	
	//Для Счетчик = 5 По 20 Цикл
	//
	// Сообщить(Счетчик);
	// Сообщить(Счетчик*Счетчик);
	//
	//КонецЦикла
	
	ИскомоеЧисло = 1;
	Пока ИскомоеЧисло*ИскомоеЧисло<30000 Цикл
	//Продолжить;
	ИскомоеЧисло = ИскомоеЧисло + 1;
	КонецЦикла;
	
	Сообщить(ИскомоеЧисло-1);	

13		**ДОМАШНЕЕ ЗАДАНИЕ №7** Сообщите на какой день недели выпадает 31 декабря в ближайшие 10 лет. Kursy-po-1C.ru Название дня недели нужно выдать в текстовом виде, например – понедельник, вторник, … Подсказка. Для определения дня недели используйте встроенную функцию ДеньНедели.

14		
16		**ДОМАШНЕЕ ЗАДАНИЕ №8** При записи документа Реализация товаров сохраните общую сумму документа в реквизит Сумма документа . Также необходимо выполнить проверку, что в документе Реализация товаров выбран контрагент, у которого указан флаг Клиент . Подсказка 1. Для проверки корректности заполнения реквизитов используйте событие ОбработкаПроверкиЗаполнения. Подсказка 2. Чтобы получить значение реквизита обращайтесь через точку . Например, Сообщить(Контрагент.ЭтоКлиент).
17	Процедура ОбработкаПроверкиЗаполнения(Отказ, ПроверяемыеРеквизиты) // Вставить содержимое обработчика. КонецПроцедуры Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения) СуммаДокумента = 0; Для Каждого СтрокаТЧ ИЗ Товары Цикл СуммаДокумента = СуммаДокумента + СтрокаТЧ.Сумма; КонецЦикла; КонецПроцедуры	
18		

19	Перем СчастливоеЧисло;
	
	Процедура Приветствие()
	
	Сообщить("Привет, мир!");
	Сообщить("Hello, world!")
	
	КонецПроцедуры
	
	Процедура УсловныеОператоры()
	
	Если СчастливоеЧисло = 777 Тогда
	Сообщить("Все правильно");
	ИначеЕсли СчастливоеЧисло = 888 Тогда
	Сообщить("Восьмерки");
	Иначе
	Сообщить("Ошибка");
	КонецЕсли;
	
	НовоеСчастливоеЧисло = 555;
	
	КонецПроцедуры
	
	Процедура ЦиклическиеОператоры()
	
	ИскомаяСумма = КвадратыЧисел(10, 15);
	Сообщить(ИскомаяСумма);
	
	ИскомоеЧисло = 1;
	Пока ИскомоеЧисло*ИскомоеЧисло<30000 Цикл
	//Продолжить;
	ИскомоеЧисло = ИскомоеЧисло + 1;
	КонецЦикла;
	
	Сообщить(ИскомоеЧисло-1);
	
	КонецПроцедуры
	
	Функция КвадратыЧисел(НачальноеЗначение, КонечноеЗначение)
	
	СуммаКвадратов = 0;
	
	Для Счетчик = НачальноеЗначение По КонечноеЗначение Цикл
	
	Сообщить(Счетчик);
	Сообщить(Счетчик*Счетчик);	

```
                    СуммаКвадратов =
СуммаКвадратов+Счетчик*Счетчик;

        КонецЦикла;

        Возврат СуммаКвадратов;

КонецФункции

Процедура УсловныеОператорыДень7()

        Если СчастливоеЧисло>=800 И СчастливоеЧисло<=1000
Тогда
                    Сообщить("Счастливое число входит в диапазон
от 800 до 1000");
        КонецЕсли;

        Если СчастливоеЧисло<800 ИЛИ СчастливоеЧисло>1000
Тогда
                    Сообщить("Счастливое число не входит в
диапазон от 800 до 1000");
        КонецЕсли;

        Если НЕ (СчастливоеЧисло>1000) Тогда
                    Сообщить("Счастливое число меньше или равно
1000");
        КонецЕсли;

КонецПроцедуры

Процедура РаботаСПримитивнымиТипамиДанных()

        Строка1 = "   Привет";
        Строка2 = "мир!";
        ИтоговаяСтрока = Строка1 + ", "+ Строка2+"   ";
        Сообщить(ИтоговаяСтрока);

        Сообщить(СчастливоеЧисло%3);

        Сообщить(ТекущаяДата()+24*3600);

        ЗаданоСчастливоеЧисло = Ложь;
        Сообщить(ЗаданоСчастливоеЧисло);
        ЗаданоСчастливоеЧисло = НЕ ЗаданоСчастливоеЧисло;
        Сообщить(ЗаданоСчастливоеЧисло);

        Сообщить(ИтоговаяСтрока);
        Сообщить(СокрЛП(ИтоговаяСтрока));
        СтрокаБезПробелов = СокрЛП(ИтоговаяСтрока);
        Сообщить(СтрДлина(СтрокаБезПробелов));

        Сообщить(СтрЗаменить(ИтоговаяСтрока, "!", "!!!"));

        Результат = СчастливоеЧисло/39;
        Сообщить(Результат);
        Сообщить(Цел(Результат));
        Сообщить(Окр(Результат));

        Сообщить(КонецМесяца(ТекущаяДата()));

        Сообщить(ДобавитьМесяц(ТекущаяДата(), 3));

КонецПроцедуры

Процедура ПередНачаломРаботыСистемы(Отказ)
        //Отказ = Истина;
КонецПроцедуры

Процедура ПередЗавершениемРаботыСистемы(Отказ)

        ТекущаяДата1800 = КонецДня(ТекущаяДата())-6*3600+1;
        Если ТекущаяДата()<ТекущаяДата1800 Тогда

                    //Сообщить("Солнце еще высоко!");
                    //Отказ = Истина;

        КонецЕсли;

КонецПроцедуры

СчастливоеЧисло = 777;
//СчастливоеЧисло = 888;

//Приветствие();
```

	//УсловныеОператоры();
	
	//ЦиклическиеОператоры();
	
	//УсловныеОператорыДень7();
	
	//РаботаСПримитивнымиТипамиДанных();	

20		**ДОМАШНЕЕ ЗАДАНИЕ №9** Необходимо самостоятельно поработать с типом данных Массив . Массив – это упорядоченная коллекция объектов любых типов данных. В модуле управляемого приложения создайте объект тип данных Массив - для этого используйте конструктор Новый . Например, МассивЧисел = Новый Массив. Добавьте в массив два числа, для этого используйте метод Добавить . Например, МассивЧисел.Добавить(999). С помощью цикла по универсальным коллекциям обойдите элементы массива и сообщите их значения. В формах документов обеспечьте автоматический расчет суммы в табличной части, как произведение цены и количества. Также учтите ситуацию изменения суммы – в этом случае нужно пересчитать цену.
21	Реал тов и услуг - Форма &НаКлиенте Процедура ТоварыКоличествоПриИзменении(Элемент) ТекущиеДанныеТовары = Элементы.Товары.ТекущиеДанные; ТекущиеДанныеТовары.Сумма = ТекущиеДанныеТовары.Количество * ТекущиеДанныеТовары.Цена; КонецПроцедуры &НаКлиенте Процедура ТоварыЦенаПриИзменении(Элемент) ТекущиеДанныеТовары = Элементы.Товары.ТекущиеДанные; ТекущиеДанныеТовары.Сумма = ТекущиеДанныеТовары.Количество * ТекущиеДанныеТовары.Цена; КонецПроцедуры	
22		**ДОМАШНЕЕ ЗАДАНИЕ №10** Создайте отчет «Закупки товаров». Отчет должен выводить информацию о количестве и сумме закупки в разрезе номенклатуры и ответственного за закупку. Пользователь должен иметь возможность задавать период формирования отчета.
23	ВЫБРАТЬ Контрагенты.Код, Контрагенты.Наименование ИЗ Справочник.Контрагенты КАК Контрагенты	
24		
25	ВЫБРАТЬ Номенклатура.ОсновнойПоставщик, Номенклатура.Ссылка КАК Номенклатура ИЗ Справочник.Номенклатура КАК Номенклатура	
26		
27	ВЫБРАТЬ РеализацияТоваровИУслугТовары.Товар, РеализацияТоваровИУслугТовары.Количество, РеализацияТоваровИУслугТовары.Сумма, РеализацияТоваровИУслугТовары.Ссылка.Клиент ИЗ Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары ГДЕ	

	РеализацияТоваровИУслугТовары.Ссылка.Дата МЕЖДУ &НачалоПериода И &КонецПериода И РеализацияТоваровИУслугТовары.Ссылка.Проведен	
28		ДОМАШНЕЕ ЗАДАНИЕ №11 Необходимо создать отчет Остатки товаров по данным документа . Отчет должен выводить остатки по товарам на указанную пользователем дату. При этом в форме отчета пользователь должен указать документ Реализация товаров , именно по товарам из табличной части этого документа необходимо выводить данные об остатках.
29	ВЫБРАТЬ ПродажиОбороты.Номенклатура, ПродажиОбороты.КоличествоОборот КАК Количество, ПродажиОбороты.СуммаОборот КАК Сумма ИЗ РегистрНакопления.Продажи.Обороты(, , , Номенклатура В (ВЫБРАТЬ РеализацияТоваровИУслугТовары.Товар ИЗ Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары ГДЕ РеализацияТоваровИУслугТовары.Ссылка = &ДокументПродажи)) КАК ПродажиОбороты	
30		
31	Номенклатура В (ВЫБРАТЬ РеализацияТоваровИУслугТовары.Товар ИЗ Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары ГДЕ РеализацияТоваровИУслугТовары.Ссылка = &ДокументПродажи)	
32		
33	ОстаткиВзаиморасчетов	
34	ВЫБРАТЬ ВзаиморасчетыОстатки.Контрагент, ВзаиморасчетыОстатки.СуммаОстаток КАК Долг ИЗ РегистрНакопления.Взаиморасчеты.Остатки КАК ВзаиморасчетыОстатки	
35	ПродажиТоваровПоДокументам	
36	ВЫБРАТЬ РеализацияТоваровИУслугТовары.Товар, РеализацияТоваровИУслугТовары.Количество, РеализацияТоваровИУслугТовары.Сумма, РеализацияТоваровИУслугТовары.Ссылка.Клиент ИЗ Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары ГДЕ РеализацияТоваровИУслугТовары.Ссылка.Дата МЕЖДУ &НачалоПериода И &КонецПериода И РеализацияТоваровИУслугТовары.Ссылка.Проведен	
37	ПродажиТоваров	
38	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.Номенклатура, ПродажиОбороты.КоличествоОборот КАК Количество, ПродажиОбороты.СуммаОборот КАК Сумма ИЗ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
39		ДОМАШНЕЕ ЗАДАНИЕ №12 Обеспечьте контроль остатков для оперативно проводимых документов Реализация товаров . Также решите задачу расчета себестоимости при продаже товаров.

		Для проверки результата введите ряд документов поступления и реализации. Курсы по прикладным решениям "1С:Предприятие 8"

40	Процедура ОбработкаПроведения(Отказ, Режим)
	
	// регистр Взаиморасчеты Приход
	Движения.Взаиморасчеты.Записывать = Истина;
	Движение = Движения.Взаиморасчеты.Добавить();
	Движение.ВидДвижения =
	ВидДвиженияНакопления.Приход;
	Движение.Период = Дата;
	Движение.Контрагент = Клиент;
	Движение.Сумма = СуммаДокумента;
	
	// регистр Продажи
	Движения.Продажи.Записывать = Истина;
	Для Каждого ТекСтрокаТовары Из Товары Цикл
	Движение = Движения.Продажи.Добавить();
	Движение.Период = Дата;
	Движение.Клиент = Клиент;
	Движение.Номенклатура =
	ТекСтрокаТовары.Товар;
	Движение.Количество =
	ТекСтрокаТовары.Количество;
	Движение.Сумма = ТекСтрокаТовары.Сумма;
	КонецЦикла;
	
	// регистр ТоварыНаСкладах Расход
	Движения.ТоварыНаСкладах.Записывать = Истина;
	Для Каждого ТекСтрокаТовары Из Товары Цикл
	Движение =
	Движения.ТоварыНаСкладах.Добавить();
	Движение.ВидДвижения =
	ВидДвиженияНакопления.Расход;
	Движение.Период = Дата;
	Движение.Номенклатура =
	ТекСтрокаТовары.Товар;
	Движение.Склад = Склад;
	Движение.Количество =
	ТекСтрокаТовары.Количество;
	КонецЦикла;
	
	Движения.Записать();
	
	Если Режим = РежимПроведенияДокумента.Оперативный
	Тогда
	
	Запрос = Новый Запрос;
	Запрос.Текст =
	"ВЫБРАТЬ
	|
	ТоварыНаСкладахОстатки.Номенклатура,
	|
	ТоварыНаСкладахОстатки.Склад,
	| -
	ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
	|ИЗ
	|
	РегистрНакопления.ТоварыНаСкладах.Остатки(
	| ,
	| Склад =
	&Склад
	| И
	Номенклатура В
	|
	(ВЫБРАТЬ
	|
	РеализацияТоваровИУслугТовары.Товар
	|
	ИЗ
	|
	Документ.РеализацияТоваровИУслуг.Товары КАК
	РеализацияТоваровИУслугТовары
	|
	ГДЕ
	|
	РеализацияТоваровИУслугТовары.Ссылка =
	&Ссылка)) КАК ТоварыНаСкладахОстатки
	|ГДЕ
	|
	ТоварыНаСкладахОстатки.КоличествоОстаток < 0";
	
	Запрос.УстановитьПараметр("Склад", Склад);	

```
                    Запрос.УстановитьПараметр("Ссылка", Ссылка);

                    РезультатЗапроса = Запрос.Выполнить();

                    Если НЕ РезультатЗапроса.Пустой() Тогда

                              Отказ = Истина;

                              ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

                              Пока
ВыборкаДетальныеЗаписи.Следующий() Цикл

                                        Сообщить("Недостаточно
товара "+ВыборкаДетальныеЗаписи.Номенклатура+" в количестве
"+ВыборкаДетальныеЗаписи.Количество);

                              КонецЦикла;

                    КонецЕсли;

          КонецЕсли;

          Если НЕ Отказ Тогда

                    Движения.СебестоимостьТоваров.Записывать =
Истина;

                    Запрос = Новый Запрос;
                    Запрос.Текст =
                              "ВЫБРАТЬ
                              |
          СебестоимостьТоваровОстатки.Номенклатура,
                              |
          СебестоимостьТоваровОстатки.СуммаОстаток КАК Сумма,
                              |
          СебестоимостьТоваровОстатки.КоличествоОстаток КАК
Количество
                              |ИЗ
                              |
          РегистрНакопления.СебестоимостьТоваров.Остатки(
                              |
          &МоментВремени,
                              |                                   Номенклатура
В
                              |
          (ВЫБРАТЬ
                              |
          РеализацияТоваровИУслугТовары.Товар
                              |                                         ИЗ
                              |
          Документ.РеализацияТоваровИУслуг.Товары КАК
РеализацияТоваровИУслугТовары
                              |                                         ГДЕ
                              |
          РеализацияТоваровИУслугТовары.Ссылка = &Ссылка))
КАК СебестоимостьТоваровОстатки";

                    Запрос.УстановитьПараметр("МоментВремени",
МоментВремени());
                    Запрос.УстановитьПараметр("Ссылка", Ссылка);

                    РезультатЗапроса = Запрос.Выполнить();

                    ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

                    Пока ВыборкаДетальныеЗаписи.Следующий()
Цикл

                              Если
ВыборкаДетальныеЗаписи.Количество <> 0 Тогда
                                        СебестоимостьЕдиницы =
ВыборкаДетальныеЗаписи.Сумма/ВыборкаДетальныеЗаписи.Коли
чество;
                              Иначе
                                        СебестоимостьЕдиницы = 0;
                              КонецЕсли;

                              Движение =
Движения.СебестоимостьТоваров.Добавить();
                              Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
                              Движение.Период = Дата;
```

	СтрокаТЧ = Товары.Найти(ВыборкаДетальныеЗаписи.Номенклатура, "Товар"); Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура; Движение.Количество = СтрокаТЧ.Количество; Движение.Сумма = СебестоимостьЕдиницы * СтрокаТЧ.Количество; КонецЦикла; КонецЕсли; КонецПроцедуры	
41		
42	ВЫБРАТЬ ТоварыНаСкладахОстатки.Номенклатура, ТоварыНаСкладахОстатки.Склад, -ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество ИЗ РегистрНакопления.ТоварыНаСкладах.Остатки(, Склад = &Склад И Номенклатура В (ВЫБРАТЬ РеализацияТоваровИУслугТовары.Товар ИЗ Документ.РеализацияТоваровИУслуг.Товары КАК РеализацияТоваровИУслугТовары ГДЕ РеализацияТоваровИУслугТовары.Ссылка = &Ссылка)) КАК ТоварыНаСкладахОстатки ГДЕ ТоварыНаСкладахОстатки.КоличествоОстаток < 0	
43		**ДОМАШНЕЕ ЗАДАНИЕ №13** Обеспечьте учет суммы выручки, количества проданного товара и списанной себестоимости в оборотном регистре накопления «Продажи». Перепроведите существующие документы «Реализация товаров». Создайте отчет «Валовая прибыль». В нем должна быть представлена информация в виде таблицы. В строках – клиенты, в колонках – номенклатура, на пересечении – сумма валовой прибыли. Также в этом отчете постройте гистограмму, где в качестве точек выбраны контрагенты, а в качестве серий – номенклатура.

44	Процедура ОбработкаПроведения(Отказ, Режим)
	
	// регистр Взаиморасчеты Приход
	Движения.Взаиморасчеты.Записывать = Истина;
	Движение = Движения.Взаиморасчеты.Добавить();
	Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
	Движение.Период = Дата;
	Движение.Контрагент = Клиент;
	Движение.Сумма = СуммаДокумента;
	
	// регистр ТоварыНаСкладах Расход
	Движения.ТоварыНаСкладах.Записывать = Истина;
	Для Каждого ТекСтрокаТовары Из Товары Цикл
	Движение = Движения.ТоварыНаСкладах.Добавить();
	Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
	Движение.Период = Дата;
	Движение.Номенклатура = ТекСтрокаТовары.Товар;
	Движение.Склад = Склад;
	Движение.Количество = ТекСтрокаТовары.Количество;	

```
КонецЦикла;

Движения.Записать();

Если Режим = РежимПроведенияДокумента.Оперативный
Тогда

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|
ТоварыНаСкладахОстатки.Номенклатура,
|
ТоварыНаСкладахОстатки.Склад,
|
-
ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
|ИЗ
|
РегистрНакопления.ТоварыНаСкладах.Остатки(
|
,
|
Склад =
&Склад
|
И
Номенклатура В
|
(ВЫБРАТЬ
|
РеализацияТоваровИУслугТовары.Товар
|
ИЗ
|
Документ.РеализацияТоваровИУслуг.Товары КАК
РеализацияТоваровИУслугТовары
|
ГДЕ
|
РеализацияТоваровИУслугТовары.Ссылка =
&Ссылка)) КАК ТоварыНаСкладахОстатки
|ГДЕ
|
ТоварыНаСкладахОстатки.КоличествоОстаток < 0";

Запрос.УстановитьПараметр("Склад", Склад);
Запрос.УстановитьПараметр("Ссылка", Ссылка);

РезультатЗапроса = Запрос.Выполнить();

Если НЕ РезультатЗапроса.Пустой() Тогда

Отказ = Истина;

ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

Пока
ВыборкаДетальныеЗаписи.Следующий() Цикл

Сообщить("Недостаточно
товара "+ВыборкаДетальныеЗаписи.Номенклатура+" в количестве
"+ВыборкаДетальныеЗаписи.Количество);

КонецЦикла;

КонецЕсли;

КонецЕсли;

Если НЕ Отказ Тогда

Движения.СебестоимостьТоваров.Записывать =
Истина;
// регистр Продажи
Движения.Продажи.Записывать = Истина;

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|
СебестоимостьТоваровОстатки.Номенклатура,
|
СебестоимостьТоваровОстатки.СуммаОстаток КАК Сумма,
|
СебестоимостьТоваровОстатки.КоличествоОстаток КАК
Количество
|ИЗ
|
```

```
        РегистрНакопления.СебестоимостьТоваров.Остатки(
                                |
        &МоментВремени,
                                |                                Номенклатура
В
                                |
        (ВЫБРАТЬ
                                |
        РеализацияТоваровИУслугТовары.Товар
                                |                                        ИЗ
                                |
        Документ.РеализацияТоваровИУслуг.Товары КАК
РеализацияТоваровИУслугТовары
                                |                                        ГДЕ
                                |
        РеализацияТоваровИУслугТовары.Ссылка = &Ссылка))
КАК СебестоимостьТоваровОстатки";

                Запрос.УстановитьПараметр("МоментВремени",
МоментВремени());
                Запрос.УстановитьПараметр("Ссылка", Ссылка);

                РезультатЗапроса = Запрос.Выполнить();

                ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

                Пока ВыборкаДетальныеЗаписи.Следующий()
Цикл

                        Если
ВыборкаДетальныеЗаписи.Количество <> 0 Тогда
                                СебестоимостьЕдиницы =
ВыборкаДетальныеЗаписи.Сумма/ВыборкаДетальныеЗаписи.Коли
чество;
                        Иначе
                                СебестоимостьЕдиницы = 0;
                        КонецЕсли;

                        Движение =
Движения.СебестоимостьТоваров.Добавить();
                        Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
                        Движение.Период = Дата;

                        СтрокаТЧ =
Товары.Найти(ВыборкаДетальныеЗаписи.Номенклатура, "Товар");

                        Движение.Номенклатура =
ВыборкаДетальныеЗаписи.Номенклатура;
                        Движение.Количество =
СтрокаТЧ.Количество;
                        СебестоимостьСписания =
СебестоимостьЕдиницы * СтрокаТЧ.Количество;
                        Движение.Сумма =
СебестоимостьСписания;

                        // Движения по регистру Продажи
                        Движение =
Движения.Продажи.Добавить();
                        Движение.Период = Дата;
                        Движение.Клиент = Клиент;
                        Движение.Номенклатура =
ВыборкаДетальныеЗаписи.Номенклатура;
                        Движение.Количество =
СтрокаТЧ.Количество;
                        Движение.Сумма = СтрокаТЧ.Сумма;
                        Движение.Себестоимость =
СебестоимостьСписания;

                КонецЦикла;

        КонецЕсли;

КонецПроцедуры
```

45

46

```
ВЫБРАТЬ
        ПродажиОбороты.Клиент,
        ПродажиОбороты.Номенклатура,
        ПродажиОбороты.СуммаОборот КАК Выручка,
        ПродажиОбороты.СуммаОборот -
ПродажиОбороты.СебестоимостьОборот КАК ВаловаяПрибыль,
        (ПродажиОбороты.СуммаОборот -
ПродажиОбороты.СебестоимостьОборот) /
```

47

	ПродажиОбороты.СуммаОборот * 100 КАК Рентабельность ИЗ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
48		
49	Сумма(ВаловаяПрибыль)	
50	Окр(Сумма(ВаловаяПрибыль)/Сумма(Выручка)*100, 2)	
51	КонецПериода(&КонецПериода, "День")	
52		
53	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.Номенклатура, ПродажиОбороты.КоличествоОборот КАК Количество, ПродажиОбороты.СуммаОборот КАК Сумма ИЗ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
54	ГрафикПродажПоКлиентам	
55	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.СуммаОборот КАК Сумма, ПродажиОбороты.Период ИЗ РегистрНакопления.Продажи.Обороты(, , День,) КАК ПродажиОбороты	
56	ВаловаяПрибыль	
57	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.Номенклатура, ПродажиОбороты.СуммаОборот КАК Выручка, ПродажиОбороты.СуммаОборот - ПродажиОбороты.СебестоимостьОборот КАК ВаловаяПрибыль, (ПродажиОбороты.СуммаОборот - ПродажиОбороты.СебестоимостьОборот) / ПродажиОбороты.СуммаОборот * 100 КАК Рентабельность ИЗ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
58		**ДОМАШНЕЕ ЗАДАНИЕ №14** Создайте план счетов Бух учет , определите в нем счета, необходимые для отражения операций • Поступление товаров • Отгрузка товаров • Оплата клиента и оплата поставщику Создайте регистр бухгалтерии для отражения проводок по бухгалтерскому учету
59	Имя Код Наименование Вид Забалансовый Порядок Счета Активы 01 Активы Активный Товары 01.1 Товары Активный Покупатели 01.2 Покупатели Активный РасчетныйСчет 01.3 Расчетный счет Активный Себестоимость 01.4 Себестоимость Активный Пассивы 02 Пассивы Пассивный Поставщики 02.1 Поставщики Пассивный Выручка02.2 ВыручкаПассивный	
60		**ДОМАШНЕЕ ЗАДАНИЕ №15** Обеспечьте проведение документов Поступление товаров и Реализация товаров по бухгалтерскому учету. Перепроведите существующие документы. Создайте универсальный документ Бухгалтерская операция . Введите проводку по взносу 100 000 руб. в кассу предприятия учредителем.
61	Процедура ОбработкаПроведения(Отказ, Режим) //{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ // Данный фрагмент построен конструктором.	

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

        // регистр Взаиморасчеты Расход
        Движения.Взаиморасчеты.Записывать = Истина;
        Движение = Движения.Взаиморасчеты.Добавить();
        Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Контрагент = Поставщик;
        Движение.Сумма = СуммаДокумента;

        Движения.ТоварыНаСкладах.Записывать = Истина;
        Движения.СебестоимостьТоваров.Записывать = Истина;

        Для Каждого ТекСтрокаТовары Из Товары Цикл
                // регистр ТоварыНаСкладах Приход
                Движение =
Движения.ТоварыНаСкладах.Добавить();
                Движение.ВидДвижения =
ВидДвиженияНакопления.Приход;
                Движение.Период = Дата;
                Движение.Номенклатура =
ТекСтрокаТовары.Товар;
                Движение.Склад = Склад;
                Движение.Количество =
ТекСтрокаТовары.Количество;

                // регистр СебестоимостьТоваров Приход
                Движение =
Движения.СебестоимостьТоваров.Добавить();
                Движение.ВидДвижения =
ВидДвиженияНакопления.Приход;
                Движение.Период = Дата;
                Движение.Номенклатура =
ТекСтрокаТовары.Товар;
                Движение.Сумма = ТекСтрокаТовары.Сумма;
                Движение.Количество =
ТекСтрокаТовары.Количество;
        КонецЦикла;

        // регистр РегистрБухУчет
        Движения.РегистрБухУчет.Записывать = Истина;
        Движение = Движения.РегистрБухУчет.Добавить();
        Движение.СчетДт =
ПланыСчетов.БухгалтерскийУчет.Товары;
        Движение.СчетКт =
ПланыСчетов.БухгалтерскийУчет.Поставщики;
        Движение.Период = Дата;
        Движение.Сумма = СуммаДокумента;

        //}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

        СуммаДокумента = Товары.Итог("Сумма");

КонецПроцедуры
```

62

63

```
Процедура ОбработкаПроведения(Отказ, Режим)

        // регистр Взаиморасчеты Приход
        Движения.Взаиморасчеты.Записывать = Истина;
        Движение = Движения.Взаиморасчеты.Добавить();
        Движение.ВидДвижения =
ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Контрагент = Клиент;
        Движение.Сумма = СуммаДокумента;

        // регистр ТоварыНаСкладах Расход
        Движения.ТоварыНаСкладах.Записывать = Истина;
        Для Каждого ТекСтрокаТовары Из Товары Цикл
                Движение =
Движения.ТоварыНаСкладах.Добавить();
                Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
                Движение.Период = Дата;
                Движение.Номенклатура =
ТекСтрокаТовары.Товар;
                Движение.Склад = Склад;
                Движение.Количество =
ТекСтрокаТовары.Количество;
```

64

```
КонецЦикла;

Движения.Записать();

Если Режим = РежимПроведенияДокумента.Оперативный
Тогда

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|
ТоварыНаСкладахОстатки.Номенклатура,
|
ТоварыНаСкладахОстатки.Склад,
|
-
ТоварыНаСкладахОстатки.КоличествоОстаток КАК Количество
|ИЗ
|
РегистрНакопления.ТоварыНаСкладах.Остатки(
|
,
|
Склад =
&Склад
|
И
Номенклатура В
|
(ВЫБРАТЬ
|
РеализацияТоваровИУслугТовары.Товар
|
ИЗ
|
Документ.РеализацияТоваровИУслуг.Товары КАК
РеализацияТоваровИУслугТовары
|
ГДЕ
|
РеализацияТоваровИУслугТовары.Ссылка =
&Ссылка)) КАК ТоварыНаСкладахОстатки
|ГДЕ
|
ТоварыНаСкладахОстатки.КоличествоОстаток < 0";

Запрос.УстановитьПараметр("Склад", Склад);
Запрос.УстановитьПараметр("Ссылка", Ссылка);

РезультатЗапроса = Запрос.Выполнить();

Если НЕ РезультатЗапроса.Пустой() Тогда

Отказ = Истина;

ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

Пока
ВыборкаДетальныеЗаписи.Следующий() Цикл

Сообщить("Недостаточно
товара "+ВыборкаДетальныеЗаписи.Номенклатура+" в количестве
"+ВыборкаДетальныеЗаписи.Количество);

КонецЦикла;

КонецЕсли;

КонецЕсли;

Если НЕ Отказ Тогда

Движения.СебестоимостьТоваров.Записывать =
Истина;
// регистр Продажи
Движения.Продажи.Записывать = Истина;

Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
|
СебестоимостьТоваровОстатки.Номенклатура,
|
СебестоимостьТоваровОстатки.СуммаОстаток КАК Сумма,
|
СебестоимостьТоваровОстатки.КоличествоОстаток КАК
Количество
|ИЗ
|
```

```
РегистрНакопления.СебестоимостьТоваров.Остатки(
                                        |
&МоментВремени,
                                        |                                       Номенклатура
В
                                        |
(ВЫБРАТЬ
                                        |
РеализацияТоваровИУслугТовары.Товар
                                        |                                       ИЗ
                                        |
Документ.РеализацияТоваровИУслуг.Товары КАК
РеализацияТоваровИУслугТовары
                                        |                                       ГДЕ
                                        |
РеализацияТоваровИУслугТовары.Ссылка = &Ссылка))
КАК СебестоимостьТоваровОстатки";

                    Запрос.УстановитьПараметр("МоментВремени",
МоментВремени());
                    Запрос.УстановитьПараметр("Ссылка", Ссылка);

                    РезультатЗапроса = Запрос.Выполнить();

                    ВыборкаДетальныеЗаписи =
РезультатЗапроса.Выбрать();

                    СуммаСебестоимости = 0;

                    Пока ВыборкаДетальныеЗаписи.Следующий()
Цикл

                                        Если
ВыборкаДетальныеЗаписи.Количество <> 0 Тогда
                                                    СебестоимостьЕдиницы =
ВыборкаДетальныеЗаписи.Сумма/ВыборкаДетальныеЗаписи.Коли
чество;
                                        Иначе
                                                    СебестоимостьЕдиницы = 0;
                                        КонецЕсли;

                                        Движение =
Движения.СебестоимостьТоваров.Добавить();
                                        Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
                                        Движение.Период = Дата;

                                        СтрокаТЧ =
Товары.Найти(ВыборкаДетальныеЗаписи.Номенклатура, "Товар");

                                        Движение.Номенклатура =
ВыборкаДетальныеЗаписи.Номенклатура;
                                        Движение.Количество =
СтрокаТЧ.Количество;
                                        СебестоимостьСписания =
СебестоимостьЕдиницы * СтрокаТЧ.Количество;
                                        Движение.Сумма =
СебестоимостьСписания;

                                        // Движения по регистру Продажи
                                        Движение =
Движения.Продажи.Добавить();
                                        Движение.Период = Дата;
                                        Движение.Клиент = Клиент;
                                        Движение.Номенклатура =
ВыборкаДетальныеЗаписи.Номенклатура;
                                        Движение.Количество =
СтрокаТЧ.Количество;
                                        Движение.Сумма = СтрокаТЧ.Сумма;
                                        Движение.Себестоимость =
СебестоимостьСписания;

                                        СуммаСебестоимости =
СуммаСебестоимости + СебестоимостьСписания;

                    КонецЦикла;

          КонецЕсли;

          Движения.РегистрБухУчет.Записывать = Истина;

          Проводка = Движения.РегистрБухУчет.Добавить();
          Проводка.Период = Дата;
          Проводка.СчетДт =
ПланыСчетов.БухгалтерскийУчет.Покупатели;
          Проводка.СчетКт =
```

```
ПланыСчетов.БухгалтерскийУчет.Выручка;
        Проводка.Сумма = СуммаДокумента;

        Проводка = Движения.РегистрБухУчет.Добавить();
        Проводка.Период = Дата;
        Проводка.СчетДт =
ПланыСчетов.БухгалтерскийУчет.Себестоимость;
        Проводка.СчетКт =
ПланыСчетов.БухгалтерскийУчет.Товары;
        Проводка.Сумма = СуммаСебестоимости;

КонецПроцедуры
```

65

66

```
Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

        Для каждого Проводка ИЗ Движения.РегистрБухУчет Цикл

                Проводка.Период = Дата;

        КонецЦикла;

КонецПроцедуры
```

67

68

```
Процедура ОбработкаПроведения(Отказ, Режим)

        Движения.Взаиморасчеты.Записывать = Истина;
        Движения.РегистрБухУчет.Записывать = Истина;

        Для Каждого ТекСтрокаПриходДенежныхСредств Из
ПриходДенежныхСредств Цикл
                // регистр Взаиморасчеты Расход
                Движение =
Движения.Взаиморасчеты.Добавить();
                Движение.ВидДвижения =
ВидДвиженияНакопления.Расход;
                Движение.Период = Дата;
                Движение.Контрагент =
ТекСтрокаПриходДенежныхСредств.Контрагент;
                Движение.Сумма =
ТекСтрокаПриходДенежныхСредств.Сумма;

                // регистр РегистрБухУчет
                Движение =
Движения.РегистрБухУчет.Добавить();
                Движение.СчетДт =
ПланыСчетов.БухгалтерскийУчет.РасчетныйСчет;
                Движение.СчетКт =
ПланыСчетов.БухгалтерскийУчет.Покупатели;
                Движение.Период = Дата;
                Движение.Сумма =
ТекСтрокаПриходДенежныхСредств.Сумма;

        КонецЦикла;

        Движения.Взаиморасчеты.Записывать = Истина;
        Для Каждого ТекСтрокаРасходДенежныхСредств Из
РасходДенежныхСредств Цикл
                // регистр Взаиморасчеты Приход
                Движение =
Движения.Взаиморасчеты.Добавить();
                Движение.ВидДвижения =
ВидДвиженияНакопления.Приход;
                Движение.Период = Дата;
                Движение.Контрагент =
ТекСтрокаРасходДенежныхСредств.Контрагент;
                Движение.Сумма =
ТекСтрокаРасходДенежныхСредств.Сумма;
                // регистр РегистрБухУчет
                Движение =
Движения.РегистрБухУчет.Добавить();
                Движение.СчетДт =
ПланыСчетов.БухгалтерскийУчет.Поставщики;
                Движение.СчетКт =
ПланыСчетов.БухгалтерскийУчет.РасчетныйСчет;
                Движение.Период = Дата;
                Движение.Сумма =
ТекСтрокаРасходДенежныхСредств.Сумма;
        КонецЦикла;

КонецПроцедуры
```

69

70

ДОМАШНЕЕ ЗАДАНИЕ №16

		Создайте отчет Остатки по счетам БУ Отчет должен выводить остаток по выбранному счету на указанную пользователем дату Если счет не задан – должны получаться остатки по всем счетам Обеспечьте закрытие месяца, необходимо сформировать проводки вида Дт Выручка Кт Прибыль Дт Прибыль Кт Себестоимость Выполните закрытие месяца. на сумму остатка выручки на сумму остатка себестоимости
71	ВЫБРАТЬ РегистрБухУчетОстатки.Счет, РегистрБухУчетОстатки.СуммаОстатокДт, РегистрБухУчетОстатки.СуммаОстатокКт ИЗ РегистрБухгалтерии.РегистрБухУчет.Остатки КАК РегистрБухУчетОстатки	
72	ДобавитьКДате(КонецПериода(&Период, "День"), "Секунда", 1)	
73		
74	Функция ОстатокПоСчету(Счет, Период) Запрос = Новый Запрос; Запрос.Текст = "ВЫБРАТЬ | РегистрБухУчетОстатки.СуммаОстатокДт, | РегистрБухУчетОстатки.СуммаОстатокКт |ИЗ | РегистрБухгалтерии.РегистрБухУчет.Остатки(&Период, Счет = &Счет,) КАК РегистрБухУчетОстатки"; Запрос.УстановитьПараметр("Период", Период); Запрос.УстановитьПараметр("Счет", Счет); РезультатЗапроса = Запрос.Выполнить(); Если РезультатЗапроса.Пустой() Тогда Возврат 0; КонецЕсли; ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать(); ВыборкаДетальныеЗаписи.Следующий(); Если Счет.Вид = ВидСчета.Активный Тогда Возврат ВыборкаДетальныеЗаписи.СуммаОстатокДт; Иначе Возврат ВыборкаДетальныеЗаписи.СуммаОстатокКт; КонецЕсли; КонецФункции Процедура ОбработкаПроведения(Отказ, РежимПроведения) КонецПериода = КонецМесяца(Дата)+1; ОстатокВыручка = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Выручка, КонецПериода); ОстатокСебестоимость = ОстатокПоСчету(ПланыСчетов.БухгалтерскийУчет.Себестоимость, КонецПериода); Движения.РегистрБухУчет.Записывать = Истина; Проводка = Движения.РегистрБухУчет.Добавить(); Проводка.Период = Дата; Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Выручка; Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Прибыль; Проводка.Сумма = ОстатокВыручка; Проводка = Движения.РегистрБухУчет.Добавить();	

	Проводка.Период = Дата; Проводка.СчетДт = ПланыСчетов.БухгалтерскийУчет.Прибыль; Проводка.СчетКт = ПланыСчетов.БухгалтерскийУчет.Себестоимость; Проводка.Сумма = ОстатокСебестоимость; КонецПроцедуры	
75		
76	ВЫБРАТЬ РегистрБухУчетОстатки.СуммаОстатокДт, РегистрБухУчетОстатки.СуммаОстатокКт ИЗ РегистрБухгалтерии.РегистрБухУчет.Остатки(&Период, Счет = &Счет,) КАК РегистрБухУчетОстатки	
77		
78	Баланс	
79	ВЫБРАТЬ РегистрБухУчетОстатки.Счет, РегистрБухУчетОстатки.СуммаОстатокДт, РегистрБухУчетОстатки.СуммаОстатокКт ИЗ РегистрБухгалтерии.РегистрБухУчет.Остатки КАК РегистрБухУчетОстатки	
80	Сумма(СуммаОстатокДт)	
81		**ДОМАШНЕЕ ЗАДАНИЕ №17** Создайте план видов расчета Начисления , в нем будут содержаться виды расчетов, имеющие протяженность во времени. Создайте предопределенные элементы – Оклад по тарифу и Невыход сотрудника. Создайте документ Начисление оклада и Невыход сотрудника , определите структуру этих документов. Создайте регистр расчета Начисления . Он будет хранить данные о начисленной заработной плате в разрезе сотрудников. Создайте регистр сведений, описывающий график работы предприятия. Заполните данными этот регистр. Укажите связь регистра расчета с графиком.
82	Процедура ОбработкаПроведения(Отказ, Режим) Движения.ГрафикРаботы.Записывать = Истина; ДеньГрафика = ДатаНачала; Пока ДеньГрафика<=ДатаОкончания Цикл Движение = Движения.ГрафикРаботы.Добавить(); Движение.Дата = ДеньГрафика; Если ДеньНедели(ДеньГрафика)<6 Тогда Движение.РабочийДень = 1; Иначе Движение.РабочийДень = 0; КонецЕсли; ДеньГрафика = ДеньГрафика + 24*3600; КонецЦикла; КонецПроцедуры	
83		**ДОМАШНЕЕ ЗАДАНИЕ №18** Обеспечьте расчет оклада сотрудника по дневном тарифу, расчет выполняется по формуле Оклад по тарифу = Дневная оплата * Количество отработанных дней в месяце Отработанные дни месяца определяются методом отклонений Создайте новый вид расчета Премия суммой - данный вид расчета не зависит ни от каких данных Обеспечьте начисление премий сотрудникам,

		указанным в табличной части, вновь созданного документа В пользовательском режиме введите документы, отражающие начисление оклада и премий

84	Процедура ОбработкаПроведения(Отказ, Режим)
	//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	// Данный фрагмент построен конструктором.
	// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
	
	// регистр ЗаработнаяПлата
	Движения.ЗаработнаяПлата.Записывать = Истина;
	Движение = Движения.ЗаработнаяПлата.Добавить();
	Движение.Сторно = Ложь;
	Движение.ВидРасчета = ПланыВидовРасчета.Начисления.Прогул;
	Движение.ПериодДействияНачало = ДатаНачала;
	Движение.ПериодДействияКонец = КонецДня(ДатаОкончания);
	Движение.ПериодРегистрации = Дата;
	Движение.Сотрудник = Сотрудник;
	
	//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	КонецПроцедуры	

85		

86	Процедура ОбработкаПроведения(Отказ, Режим)
	
	// регистр ЗаработнаяПлата
	Движения.ЗаработнаяПлата.Записывать = Истина;
	Движение = Движения.ЗаработнаяПлата.Добавить();
	Движение.Сторно = Ложь;
	Движение.ВидРасчета = ПланыВидовРасчета.Начисления.Оклад;
	Движение.ПериодДействияНачало = НачалоМесяца(Дата);
	Движение.ПериодДействияКонец = КонецМесяца(Дата);
	Движение.ПериодРегистрации = Дата;
	Движение.Сотрудник = Сотрудник;
	Движение.РасчетныеДанные = Сумма;
	
	Движения.Записать();
	
	Запрос = Новый Запрос;
	Запрос.Текст =
	"ВЫБРАТЬ
	|
	ЗаработнаяПлатаДанныеГрафика.РабочийДеньПериодДействия КАК Норма,
	|
	ЗаработнаяПлатаДанныеГрафика.РабочийДеньФактическийПериодДействия КАК Факт,
	|
	ЗаработнаяПлатаДанныеГрафика.РасчетныеДанные
	|ИЗ
	|
	РегистрРасчета.ЗаработнаяПлата.ДанныеГрафика(Регистратор = &Регистратор) КАК ЗаработнаяПлатаДанныеГрафика";
	
	Запрос.УстановитьПараметр("Регистратор", Ссылка);
	
	РезультатЗапроса = Запрос.Выполнить();
	
	Если НЕ РезультатЗапроса.Пустой() Тогда
	
	ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
	ВыборкаДетальныеЗаписи.Следующий();
	
	Если ВыборкаДетальныеЗаписи.Норма <> 0 Тогда
	Оклад = ВыборкаДетальныеЗаписи.РасчетныеДанные * ВыборкаДетальныеЗаписи.Факт / ВыборкаДетальныеЗаписи.Норма;
	Движение.Сумма = Оклад;
	
	Движения.ЗаработнаяПлата.Записать();
	Иначе
	Сообщить("Неверно заполнен график!");
	КонецЕсли;
	
	КонецЕсли;	

	КонецПроцедуры	
87		

88	Процедура ОбработкаПроведения(Отказ, Режим)
	
	// регистр ЗаработнаяПлата
	Движения.ЗаработнаяПлата.Записывать = Истина;
	Движение = Движения.ЗаработнаяПлата.Добавить();
	Движение.Сторно = Ложь;
	Движение.ВидРасчета = ПланыВидовРасчета.Начисления.Премия;
	Движение.ПериодДействияНачало = НачалоМесяца(Дата);
	Движение.ПериодДействияКонец = КонецМесяца(Дата);
	Движение.ПериодРегистрации = Дата;
	Движение.Сотрудник = Сотрудник;
	Движение.РасчетныеДанные = Процент;
	
	Движения.Записать();
	
	Запрос = Новый Запрос;
	Запрос.Текст =
	"ВЫБРАТЬ
	|
	ЗаработнаяПлатаБазаЗаработнаяПлата.СуммаБаза,
	|
	ЗаработнаяПлатаБазаЗаработнаяПлата.РасчетныеДанные
	|ИЗ
	|
	РегистрРасчета.ЗаработнаяПлата.БазаЗаработнаяПлата(&Измерения, &Измерения, , Регистратор = &Ссылка) КАК ЗаработнаяПлатаБазаЗаработнаяПлата";
	
	Измерения = Новый Массив;
	Измерения.Добавить("Сотрудник");
	
	Запрос.УстановитьПараметр("Измерения", Измерения);
	Запрос.УстановитьПараметр("Ссылка", Ссылка);
	
	РезультатЗапроса = Запрос.Выполнить();
	
	Если НЕ РезультатЗапроса.Пустой() Тогда
	
	ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();
	ВыборкаДетальныеЗаписи.Следующий();
	
	Движение.Сумма = ВыборкаДетальныеЗаписи.СуммаБаза * ВыборкаДетальныеЗаписи.РасчетныеДанные / 100;
	
	Движения.ЗаработнаяПлата.Записать();
	
	КонецЕсли;
	
	КонецПроцедуры	

89		**ДОМАШНЕЕ ЗАДАНИЕ №19** Отразите процессы выплаты заработной платы. Для этого потребуется создание регистра накопления. Проведите документы, отражающие выплату, в пользовательском режиме. Создайте отчет Задолженность по зарплате , показывающий на определенную дату, долги перед сотрудниками.
90	ВЫБРАТЬ ЗаработнаяПлата.Сотрудник, ЗаработнаяПлата.ВидРасчета, ЗаработнаяПлата.Сумма ИЗ РегистрРасчета.ЗаработнаяПлата КАК ЗаработнаяПлата	
91		
92	ОтработанноеВремя -ФормаОтчета	
93	&НаСервере Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка) Запрос = Новый Запрос; Запрос.Текст = "ВЫБРАТЬ | ЗаработнаяПлатаФактическийПериодДействия.Сотрудник,	

	| ЗаработнаяПлатаФактическийПериодДействия.ВидРасчета, | ЗаработнаяПлатаФактическийПериодДействия.ПериодДействияНачало, | ЗаработнаяПлатаФактическийПериодДействия.ПериодДействияКонец |ИЗ | РегистрРасчета.ЗаработнаяПлата.ФактическийПериодДействия КАК ЗаработнаяПлатаФактическийПериодДействия"; РезультатЗапроса = Запрос.Выполнить(); ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать(); Пока ВыборкаДетальныеЗаписи.Следующий() Цикл Точка = ДиаграммаГантаРасчет.УстановитьТочку(ВыборкаДетальныеЗаписи.Сотрудник); Серия = ДиаграммаГантаРасчет.УстановитьСерию(ВыборкаДетальныеЗаписи.ВидРасчета); Значение = ДиаграммаГантаРасчет.ПолучитьЗначение(Точка, Серия); Интервал = Значение.Добавить(); Интервал.Начало = ВыборкаДетальныеЗаписи.ПериодДействияНачало; Интервал.Конец = ВыборкаДетальныеЗаписи.ПериодДействияКонец; КонецЦикла; КонецПроцедуры	
94		

95	Процедура ОбработкаПроведения(Отказ, Режим)
	//{{__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	// Данный фрагмент построен конструктором.
	// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
	
	// регистр ДолгиПоЗаработнойПлате Расход
	Движения.ДолгиПоЗаработнойПлате.Записывать = Истина;
	Для Каждого ТекСтрокаСотрудники Из Сотрудники Цикл
	Движение = Движения.ДолгиПоЗаработнойПлате.Добавить();
	Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
	Движение.Период = Дата;
	Движение.Сотрудник = ТекСтрокаСотрудники.Сотрудник;
	Движение.Сумма = ТекСтрокаСотрудники.Сумма;
	КонецЦикла;
	
	//}}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
	КонецПроцедуры	

96		**ДОМАШНЕЕ ЗАДАНИЕ №20** Постройте отчет, показывающий ABC-классификацию номенклатуры по параметру Валовая прибыль . Введите такой набор данных о продаже, чтобы в каждый класс входило, как минимум, 2 номенклатуры.
97	ВаловаяПрибыль	
98	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.Номенклатура, ПродажиОбороты.СуммаОборот КАК Выручка, ПродажиОбороты.СуммаОборот - ПродажиОбороты.СебестоимостьОборот КАК ВаловаяПрибыль, (ПродажиОбороты.СуммаОборот - ПродажиОбороты.СебестоимостьОборот) / ПродажиОбороты.СуммаОборот * 100 КАК Рентабельность ИЗ	

	РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
99	Сумма(ВаловаяПрибыль)	
100	Окр(Сумма(ВаловаяПрибыль)/Сумма(Выручка)*100, 2)	
1	КонецПериода(&КонецПериода, "День")	
2		
3	ЭффективностьРаботыМенеджеров	
4	ВЫБРАТЬ ЭффективностьПродажОбороты.Сотрудник, ЭффективностьПродажОбороты.ВыручкаОборот КАК Выручка, ЭффективностьПродажОбороты.ПрибыльОборот КАК Прибыль, ЭффективностьПродажОбороты.КоличествоЗаказовОборот КАК КоличествоЗаказов, ЭффективностьПродажОбороты.КоличествоОтгрузокОборот КАК КоличествоОтгрузок ИЗ РегистрНакопления.ЭффективностьПродаж.Обороты КАК ЭффективностьПродажОбороты	
5	Сумма(Выручка)	
6	Сумма(КоличествоЗаказов)	
7		
8	КлассификацияКлиентов	
9	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.СуммаОборот КАК Выручка, ПродажиОбороты.СуммаОборот КАК КлассАВС ИЗ РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты	
10	Сумма(Выручка)	
11	КлассификацияАВС("Сумма(КлассАВС)", 3)	
12		
13	Макет	
14	КлассАВС Класс	
16	Выбор Когда КлассАВС = 1 Тогда "Класс А" Когда КлассАВС=2 Тогда "Класс В" Иначе "Класс С" Конец	
17		
18	ГрафикПродажПоКлиентам	
19	ВЫБРАТЬ ПродажиОбороты.Клиент, ПродажиОбороты.СуммаОборот КАК Сумма, ПродажиОбороты.Период ИЗ РегистрНакопления.Продажи.Обороты(, , День,) КАК ПродажиОбороты	
20		**ДОМАШНЕЕ ЗАДАНИЕ №21** Реализуйте бизнес-процесс тестирования кандидата при приеме на работу.
21	БП ПриемСотрудникаНаРаботу	
22	форма	
23	&НаСервере Процедура ПриЧтенииНаСервере(ТекущийОбъект) КартаМаршрута = ТекущийОбъект.ПолучитьКартуМаршрута(); КонецПроцедуры	
24	Старт; Линия1; ПодготовкаРабочегоМеста; Линия2; ЗнакомствоСОфисом; Линия3; ЗнакомствоСКоллективом; Линия4; Завершение	

25		
26	НастройчиваяПродажа	
27	&НаСервере Процедура ПриЧтенииНаСервере(ТекущийОбъект) КартаМаршрута = ТекущийОбъект.ПолучитьКартуМаршрута(); КонецПроцедуры	
28	Старт; Линия1; СборДанныхОКлиенте; Линия2; ФормированиеКоммерческогоПредложения; Линия3; УсловияПродажСогласованы; Линия4; Линия5; ПодписаниеДоговора; Линия6; ОтгрузкаТоваров; Линия7; Завершение; СнизитьЦены; Линия8	
29	Процедура УсловияПродажСогласованыПроверкаУсловия(ТочкаМаршрутаБизнесПроцесса, Результат) Результат = УсловияПродажСогласованы; КонецПроцедуры	
30	&НаКлиенте Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды) //{{_КОНСТРУКТОР_ПЕЧАТИ(ПечатьСчетаНаОплату) ТабДок = Новый ТабличныйДокумент; ПечатьСчетаНаОплату(ТабДок, ПараметрКоманды); ТабДок.ОтображатьСетку = Ложь; ТабДок.Защита = Ложь; ТабДок.ТолькоПросмотр = Ложь; ТабДок.ОтображатьЗаголовки = Ложь; ТабДок.Показать(); //}} КонецПроцедуры &НаСервере Процедура ПечатьСчетаНаОплату(ТабДок, ПараметрКоманды) Документы.ЗаказКлиента.ПечатьСчетаНаОплату(ТабДок, ПараметрКоманды); КонецПроцедуры	