

Глава 11. Торговля и производство

Общие свойства таблиц наименований	2
ТекстСвертки.....	2
Товар	3
ТипНоменклатуры	4
НаимОснования	5
КоличествоВКомплексе	6
Номенклатура.....	6
ДобавитьЦены.....	6
КодПоставщика	7
ОсновнойПоставщик.....	8
КодКонкурента	9
ОсновнойКонкурент.....	10
Карточка складской картотеки	11
НайтиНаСкладе	11
СоздатьНаСкладе	11
НайтиНаСкладеПоНомНомеру.....	12
ЛогистическиеПараметры.....	13
СоздатьСкладскуюКарточку	14
НайтиСклад	15
ВРезервеПрих.....	15
ВРезервеРасх.....	16
ПересчитатьОстаток	17
ДляВсех(Приходов) – для склада	17
ДляВсех(Расходов) – для склада	18
Анализ движения товара	19
ПриходПоРасходу	19
ПриходПоПартии	20
НТДДляРасхода	21
ОстНач	22
ОстКнц	24
ОбрПрих.....	25
ОбрРасх.....	27
Выписано	28
Заказано.....	30
ДляВсех(Остатков)	31
ДляВсех(ХарактеристикПартии)	32
Складские документы	33
СчетУчета.....	33
Склад	34
НайтиПрайс.....	35
ЕстьИИспользуется.....	36
РассчитатьЦену	37
НайтиПоКлючевымСловам.....	38
НайтиПоКлючевымСловамВКартотеке	39
НайтиПоКлючевымСловамСочетаниями	41
РазбитьНаКлючевыеСлова	41
ЗаполнитьКлючевымиСловами	42
Производство	43
НайтиТехоперацию	43



ТарифРаботы	44
ДляВсех(Компонентов)	45
ДляВсех(КомпонентовНаименования)	46
ДляВсех(РаботНаименования)	47
ДляВсех(СписанийНаименования)	48
ДляВсех(ОтходовНаименования)	49
ДляВсех(Приходов)	50
ДляВсех(СписанийСборки)	51
ЕстьСостав	52
ПереопределитьСостав	53
ПереопределитьРаботы	54
Ценообразование	56
ПолучитьЦеныПоПрайсу	56
ДобавитьВидЦены	57
ДобавитьВидыЦенКарты	58
СуммыПоКарте	59
ОчиститьВидыЦен	61
РассчитатьЦены	62
ЦеныНаДату	63

Общие свойства таблиц наименований

ТекстСвертки

Возвращает список составляющих комплекта из справочника «Номенклатура». Тип номенклатуры указывается в поле «Тип» на вкладке «Общее». Список составляющих комплекта указывается на вкладке «Состав» карточки номенклатуры.

Синтаксис

Запись . ТекстСвертки ()

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования документов.

Возвращает



Текст – перечень составляющих через запятую. Если товар не является комплектом, то возвращает наименование товара.

Примеры

Узнаем состав комплекта в приходной накладной:

```
oНакладная = Таблица ("Документы") ;  
oНакладная .ТипДокумента = "НаклПрих" ;  
oНакладная .Дата = 06.04.05 ;  
oНакладная .Номер = 54 ;  
Найти (oНакладная , "Номер" ) ;  
ДляВсех (oНакладная .СвернутыхНаименований ( ) )  
Сообщить (ТекстСвертки ( ) ) ;
```

См. также:

- ДляВсех(СвёрнутыхНаименований)

Товар

Возвращает запись складской картотеки из соответствующего справочника.

Синтаксис

Запись .Товар ()

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования складских документов и отчетов по складу. Объект может использоваться во всех местах, где встречается товар или материал из складской картотеки. Данные объекта содержат все те поля, что доступны в складской картотеке.

В контексте наименований продаж функция возвращает запись справочника «Номенклатура». В остальных случаях – запись справочника «Складская картотека».

Возвращает

Объект – запись складской картотеки.

Примеры

Узнаем остаток товаров на складе, перебирая записи в накладной:

```
объект оНакладная ;
ДляВсех (оНакладная . Наименований ( ) )
{
    Сообщить ("На складе № " +
Наим . Товар ( ) . НомерСклада + " числится " + " товара "
+ Наим . Товар ( ) . Название +
" в количестве " + Наим . Товар ( ) . Остаток + " " +
Наим . Товар ( ) . ЕДИЗмерения) ;
}
```

См. также:

- [ДляВсех\(Наименований\)](#)

ТипНоменклатуры

Возвращает запись типа номенклатуры, указанный в поле «Тип номенклатуры» в карточке номенклатуры.

Синтаксис

```
Запись . ТипНоменклатуры ( )
```

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования складских документов и отчетов по складу.

Возвращает

Объект – запись из таблицы «Типы номенклатуры».

Примеры

Узнаем тип номенклатуры в приходной накладной:

```
функция НаДиалогСохранить ( )
{
```



```
ДляВсех (Наименований ( ) )  
    Сообщить (Наим . ТипНоменклатуры ( ) . Название) ;  
    вернуть Следфункция ;  
}
```

См. также:

- [ДляВсех\(Наименований\)](#)

НаимОснования

Возвращает запись наименования документа-основания (документа, указанного в таблице «По документам»).

Синтаксис

Запись . НаимОснования ()

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования складских документов.

Возвращает

Объект – запись из таблицы «Документы».

Примеры

Необходимо поменять базовую цену наименования в документе-основании:

```
функция НаДиалогСохранить ( )  
{  
    ДляВсех (Наименований ( ) )  
    {  
        оНаим=Наим . НаимОснования ( ) ;  
        оНаим . БазоваяЦена==15000 ;  
    }  
    вернуть Следфункция ;  
}
```

См. также:



- [ДляВсех\(Наименований\)](#)

КоличествоВКомплекте

Рассчитывает для наименования документа (расходной накладной, счета, и т.д.), являющегося составляющей комплекта или модификатором, количество, в котором оно присутствует в исходном составе комплекта или наименования с модификаторами.

Синтаксис

Наим . КоличествоВКомплекте ()

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования складских документов (расходной накладной, счета, и т.д.). У модификатора обязательно должна быть установлена галка «Обязательный».

Возвращает

Число – количество наименование документа, являющегося составляющей комплекта или модификатором, в котором оно присутствует в исходном составе комплекта или наименования с модификаторами.

См. также:

- [ДляВсех\(Наименований\)](#)

Номенклатура

ДобавитьЦены

Добавляет цены на вкладке «Цены» карточки номенклатуры на указанную дату и время.

Синтаксис

6



```
Запись.ДобавитьЦены(Цена1[, Цена2, Цена3], Дата)  
Запись.ДобавитьЦены(Цена1, Цена2, Дата)  
Запись.ДобавитьЦены(Цена1, Цена2, Цена3, Цена4,  
Дата[, Время])
```

Параметры

Цена1, Цена2, ...(число) – вводимые цены номенклатуры.

Дата (дата) – дата действия введенных цен.

Время (время) – время, с которого будут действовать введенные цены. Если параметр не указан, функция установит цены, действующие на текущее время.

Возвращает

Да (логическое значение) – если цены добавлены.

Нет (логическое значение) – если цены не добавлены.

Примеры

Введем новые цены для номенклатуры, действующие с начала указанной даты:

```
oЛицо = Лицо("X04", "Номенклатура");  
oЛицо.Лицо_.ДобавитьЦены(100, 120, 150, 01.01.05);
```

КодПоставщика

Возвращает код основного поставщика товара, указанного в поле «Код у поставщика» (закладка «Поставка и продажа») в карточке номенклатуры.

Синтаксис

```
Товар.КодПоставщика(Поставщик)
```

Параметры

Поставщик (объект) – запись поставщика данного товара из таблицы «Лица».

Возвращает

Текст – код товара у основного поставщика.

Комментарии

Функция работает в контексте записи номенклатуры или карточки складской картотеки. В противном случае будет выдано сообщение об ошибке. Поставщик обязательно должен быть из категории «Основной поставщик». Функция не учитывает наследование кода поставщика с папки номенклатуры, он обязательно должен быть указан в карточке номенклатуры.

Примеры

Необходимо вывести код товара у основного поставщика:

```
oЛицо = Лицо ("7714123412", "Организации");
oТовар = Выборка ("Номенклатура");
oТовар.НомНомер = 10903;
Найти (oТовар, "НомНомер");
Сообщить ("Код товара "+oТовар.Наименование+" =
"+oТовар.КодПоставщика (oЛицо) );
```

См. также:

- [КодКонкурента](#)

ОсновнойПоставщик

Возвращает запись основного поставщика, указанного в карточке номенклатуры (закладка «Поставка и продажа»).

Синтаксис

```
Товар.ОсновнойПоставщик ( )
```

Параметры

Не указываются.

Возвращает

Объект – запись поставщика из таблицы «Лица».

Комментарии

Функция работает в контексте записи номенклатуры или карточки складской картотеки. В противном случае будет выдано сообщение об ошибке.



Поставщик обязательно должен быть из категории «Основной поставщик».

Примеры

Необходимо определить основного поставщика для карточки с номенклатурным номером "M01_03":

```
oТовар=Выборка ("Номенклатура" );  
oТовар.НомНомер="M01_03" ;  
Найти (oТовар , "НомНомер" ) ;  
oЛицо=oТовар.ОсновнойПоставщик ( ) ;  
Сообщить ("Основной поставщик товара " +  
oТовар.Наименование+ " = " +oЛицо.Название) ;
```

См. также:

- [ОсновнойКонкурент](#)

КодКонкурента

Возвращает код товара у конкурента, указанного в поле «Код у конкурента» (закладка «Поставка и продажа») в карточке номенклатуры.

Синтаксис

Товар . КодКонкурента (Конкурент)

Параметры

Конкурент (объект) – запись конкурента по данному товару из таблицы «Лица».

Возвращает

Текст – код товара у конкурента.

Комментарии

Функция работает в контексте записи номенклатуры или карточки складской картотеки. В противном случае будет выдано сообщение об ошибке. Конкурент обязательно должен быть основным, т.е. в его карточке должен быть установлен признак «Основной конкурент». Функция не учитывает наследование кода конкурента с папки номенклатуры, он обязательно должен быть указан в карточке номенклатуры.

Примеры

Необходимо вывести код товара у основного конкурента:

```
оЛицо = Лицо ("7714123412", "Организации");  
оТовар = Выборка ("Номенклатура");  
оТовар.НомНомер = 10903;  
Найти (оТовар, "НомНомер");  
Сообщить ("Код товара "+оТовар.Наименование+" =  
"+оТовар.КодКонкурента (оЛицо) );
```

См. также:

- [КодПоставщика](#)

ОсновнойКонкурент

Возвращает запись основного конкурента, указанного в карточке номенклатуры (закладка «Поставка и продажа»).

Синтаксис

```
Товар.ОсновнойКонкурент ( )
```

Параметры

Не указываются.

Возвращает

Объект – запись конкурента из таблицы «Лица».

Комментарии

Функция работает в контексте записи номенклатуры или карточки складской картотеки. В противном случае будет выдано сообщение об ошибке. Конкурент обязательно должен быть основным, т.е. в его карточке должен быть установлен признак «Основной конкурент».

Примеры

Необходимо определить основного конкурента для карточки с номенклатурным номером "M01_03":

```
оТовар=Выборка ("Номенклатура" ) ;
```



```
oТовар.НомНомер="M01_03";  
Найти(oТовар, "НомНомер");  
oЛицо=oТовар.ОсновнойКонкурент();  
Сообщить("Основной конкурент товара " +  
oТовар.Наименование+ " = " +oЛицо.Название);
```

См. также:

- [ОсновнойПоставщик](#)

Карточка складской картотеки

НайтиНаСкладе

Ищет карточку на складе для указанной номенклатуры.

Синтаксис

`ЗаписьНоменклатуры.НайтиНаСкладе(НомерСклада)`

Параметры

НомерСклада (число) – номер склада, на котором ищется карточка.

Возвращает

Объект – запись карточки на складе для указанной номенклатуры. Если карточки нет, то функция возвращает пустой объект.

Примеры

Найдем карточку номенклатуры с указанным номером на складе №6:

```
oЛицо = Лицо("10901", "Номенклатура");  
oН = oЛицо.Лицо_.НайтиНаСкладе(6);
```

См. также:

- [НайтиНаСкладеПоНомНомеру](#)

СоздатьНаСкладе

Создаёт карточку на складе для указанной номенклатуры.

Синтаксис

ЗаписьНоменклатуры.СоздатьНаСкладе (Номер)

Параметры

Номер (целое) – номер склада, на котором необходимо создать карточку для указанной номенклатуры.

Возвращает

Объект, содержащий запись складской карточки для указанной номенклатуры.

Комментарии

Для работы функции необходимо наличие контекстного объекта с записью таблицы «Складская картотека».

Примеры

Необходимо продублировать определенную карточку номенклатуры на склад с указанным номером:

```
оЛицо = Лицо ("10902", "Номенклатура") ;  
оН = оЛицо.Лицо_.СоздатьНаСкладе (2) ;
```

См. также:

- [НайтиНаСкладе](#)
- [НайтиСклад](#)

НайтиНаСкладеПоНомНомеру

Ищет карточку складского учёта.

Синтаксис

НайтиНаСкладеПоНомНомеру (НомНомер , НомерСклада)

Параметры

НомНомер (текст) – номенклатурный номер товара.

НомерСклада (целое) – номер склада .



Возвращает

Объект – карточку складского учета.

Примеры

Найдем карточку с номером «УО02» на указанном складе:

`oКарточка = НайтиНаСкладеПоНомНомеру ("УО02" , 1) ;`

См. также:

- [НайтиНаСкладе](#)
- [НайтиПоКлючевымСловам](#)
- [НайтиПоКлючевымСловамВКартотеке](#)
- [НайтиПоКлючевымСловамСочетаниями](#)

ЛогистическиеПараметры

Возвращает объект, содержащий логистические параметры карточки складской картотеки: мин. запас на складе, макс. запас на складе и др. Все эти параметры указаны в карточке на вкладке «**Логистика**» в таблице «Логистические параметры».

Синтаксис

`ЗаписьСкладскойКартотеки . ЛогистическиеПараметры (Дата)`

Параметры

Дата (дата) – указывается дата, на которую необходимо получить всю историю запаса по конкретному товару. Если на эту дату истории не оказывается, то берётся более ранняя дата.

Возвращает

Объект – объект, содержащий запись из таблицы «**История запаса**» для указанного товара на указанную дату. Если записи на данную дату нет, то функция возвратит ближайшую запись с более ранней датой. Если вообще ничего нет, то пустой объект.



Примеры

Необходимо для выбранной номенклатурной карточки на складе №1 выяснить, есть ли нехватка товара:

```
оЛицо = Лицо ("Т18_07", "Номенклатура");
оСклКарточка = оЛицо.Лицо_НайтиНаСкладе (1);
оЛогПараметры = оСклКарточка.ЛогистическиеПараметры (
ТекДата ( ) );
пМинЗапас = оЛогПараметры.МинЗапас;
оОстаток = оСклКарточка.ОстКнц (ТекДата ( ) );
пОстаток = оОстаток.Кол_во;
Если (пОстаток < пМинЗапас)
    Сообщить ( "На складе №1 недостаток товара " +
оСклКарточка.Наименование + "!" );
```

См. также:

- [НайтиНаСкладе](#)
- [НайтиНаСкладеПоНомНомеру](#)

СоздатьСкладскуюКарточку

Создает карточку в справочнике «Складская картотека».

Синтаксис

**СоздатьСкладскуюКарточку (НомерСклада , НомНомер ,
Наименование)**

Параметры

НомерСклад (целое) – номер склада .

НомНомер (текст) – номенклатурный номер товара.

Наименование (текст) – наименование товара в справочнике.

Возвращает

Объект – новую карточку складского учета.



Примеры

Создадим на складе №1 новую карточку «Товар с НДС 10%» с номенклатурным номером «P1»:

```
пК = СоздатьСкладскуюКарточку( 4, "P1", "Товар с НДС 10%" );
```

См. также:

- [СоздатьНаСкладе](#)
- [НайтиНаСкладе](#)

НайтиСклад

Возвращает запись склада с указанным номером из таблицы «Складская картотека».

Синтаксис

НайтиСклад (НомерСклада)

Параметры

НомерСклада (число) – номер искомого склада.

Возвращает

Объект – запись искомого склада из таблицы «Складская картотека». Если записи с указанным номером склада не существует, то в этом случае функция вернёт пустой объект.

Примеры

Узнаем номенклатурный номер склада оСклад:

```
оСклад = НайтиСклад(4);  
Сообщить(оСклад.НомНомер);
```

См. также:

- [Склад](#)
- [НайтиНаСкладе](#)

ВРезервеПрих

Возвращает количество зарезервированного товара по непроведенным приходным документам (приходные накладные, документы внутреннего перемещения и т.д.).

Синтаксис

Запись .ВРезервеПрих ()

Параметры

Не указываются.

Возвращает

Число – количество зарезервированного товара по документам.

Примеры

Узнаем о наличии резерва на товар на складе№1:

```
оКарт = НайтиНаСкладеПоНомНомеру ("10901", 1);
Если (!оКарт.ВРезервеПрих () == 0)
    Сообщить ("Зарезервировано товара в количестве " +
оКарт.ВРезервеПрих ())
Иначе
    Сообщить ("Товара в резерве нет");
```

См. также:

- [ВРезервеРасх](#)
- [НайтиНаСкладе](#)

ВРезервеРасх

Возвращает количество зарезервированного товара по непроведенным расходным документам (расходные накладные, документы внутреннего перемещения и т.д.).

Синтаксис

Запись .ВРезервеРасх ()

Параметры



Не указываются.

Возвращает

Число – количество зарезервированного товара по документам.

Примеры

Узнаем количество зарезервированного товара по расходу:

```
оКарт = НайтиНаСкладеПоНомеру ("10901", 1);  
Сообщить (оКарт. ВРезервеРасх ());
```

См. также:

- [ВРезервеПрих](#)
- [НайтиНаСкладе](#)

ПересчитатьОстаток

Пересчитывает остатки товара на складе.

Синтаксис

Запись . **ПересчитатьОстаток** ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – остатки пересчитаны.

Нет (логическое значение) – остатки не пересчитаны.

См. также:

- [ОстНач](#)
- [ОстКнц](#)

ДляВсех(Приходов) – для склада

Перебирает все приходы для определенного товара за указанный период.

Синтаксис

`ДляВсех (Запись . Приходов ([ДатНач , ДатКнц]))`

Параметры

ДатНач, ДатКнц (даты) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.

Возвращает

Ничего не возвращает.

Комментарии

Функция устанавливает контекстный объект с именем «Приход», который содержит запись таблицы «Приход».

См. также:

- [ДляВсех\(Расходов\) – для склада](#)

ДляВсех(Расходов) – для склада

Перебирает все расходы для определенного товара за указанный период.

Синтаксис

`ДляВсех (Запись . Расходов ([ДатНач , ДатКнц]))`

Параметры

ДатНач, ДатКнц (даты) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.

Возвращает

Ничего не возвращает.

Комментарии



Функция устанавливает контекстный объект с именем «Расход», который содержит запись таблицы «Расход».

Примеры

Необходимо подсчитать общее количество проданного товара за указанный период:

```
оКарт = НайтиНаСкладеПоНомНомеру ("10902", 3);  
ДатНач = Дата (01.01.06); ДатКнц = Дата (07.01.06);  
Колво = 0;  
ДляВсех (оКарт.Расходов (ДатНач, ДатКнц) )  
{  
    Колво = Колво + .оКарт.РАСХОД.Кол_во;  
}  
Сообщить (Колво);
```

См. также:

- [ДляВсех\(Приходов\) – для склада](#)

Анализ движения товара

ПриходПоРасходу

Находит первоначальную партию прихода товара по записи расхода.

Синтаксис

Запись . ПриходПоРасходу ()

Параметры

Не указываются.

Комментарии

Данная функция вызывается в контексте каждого наименования расхода и возвращает объект из первой списываемой партии или если такой она не нашла то НТД и страна – производитель берутся с карточки номенклатуры, с которой связано наименование.

Возвращает

Объект – запись о первоначальной партии прихода, с которой был списан данный товар. Даже если были перемещения товара по складам, будет возвращена именно исходная приходная партия. Если приходная партия не найдена, то возвращается пустой объект.

Примеры

Сообщить входную цену списываемого товара:

```
ДляВсех (оДок . Списаний ( ) )
{
    оПрих = Наим . ПриходПоРасходу ( ) ;
    Если (Есть (оПрих . Кол_во ) )
        Сообщить (оПрих . СуммаСебест / оПрих . Кол_во ) ;
}
```

См. также:

- [ДляВсех\(Приходов\)](#)
- [ПриходПоПартии](#)
- [НТДДляРасхода](#)

ПриходПоПартии

Возвращает первый приход по партии, с которой был списан расход.

Синтаксис

Запись . ПриходПоПартии ()

Параметры

Не указываются.

Комментарии

Данная функция вызывается в контексте записи **ПАРТИЯРАСХОДЫ** и возвращает партию прихода, с которой списан был расход, принимаемый в качестве аргумента, поднимаясь до последней партии, на которой обрывается цепь перемещений.



Возвращает

Объект – запись из таблицы «Приход», первый приход по партии, с которой был списан расход.

Примеры

Необходимо узнать стоимость партии, по которой списывается расход:

```
функция НаДокументЗакреть ( )
{
    ДляВсех (Списаний ( ) )
    {
        оПрих = ПартияРасходы.ПриходПоПартии ( ) ;
        Если (оПрих)
            Сообщить (оПрих.СуммаСебест) ;
    }
}
```

См. также:

- [ДляВсех\(Приходов\)](#)
- [ПриходПоРасходу](#)
- [НТДДляРасхода](#)

НТДДляРасхода

Возвращает объект, содержащий номер таможенной декларации (в дальнейшем НТД) и Страну из списываемой партии товара.

Синтаксис

Запись . НТДДляРасхода ()

Параметры

Не указываются.

Комментарии

Данная функция вызывается в контексте каждого наименования расхода и возвращает объект из первой списываемой партии или если такой она не нашла то НТД и страна – производитель берутся с карточки номенклатуры, с которой связано наименование.

Возвращает

Объект – объект, содержащий НТД и страну - производителя.

Примеры

Узнаем номера таможенной декларации у наименований указанной накладной:

```
оДок = Таблица ("Документы" );
оДок.ТипДокумента = "НаклРасж" ;
оДок.Дата = 21.03.05 ;
оДок.Номер = 4500017 ;
пн = Найти (оДок , "Номер" ) ;
ДляВсех (оДок .Списаний ( ) )
    Сообщить (Наим .НТДДляРасхода ( ) .нтд) ;
```

См. также:

- [ДляВсех\(Приходов\)](#)
- [ПриходПоРасходу](#)

ОстНач

Рассчитывает остатки номенклатуры на складах хранения этой номенклатуры, либо остатки по конкретной карточке складской картотеки на начало дня. Аналогичные остатки можно узнать, построив «Ведомость остатков» на начало дня.

Синтаксис

```
ЗаписьНоменклатуры. ОстНач ( ДатНач [ , НомерСклада ,
Поставщик , [ ВидОтгрузки [ , Тема [ , ВидСобственности
]]]] )
ЗаписьСкладскойКартотеки. ОстНач ( ДатНач ,
НомерСклада , Поставщик , [ ВидОтгрузки [ , Тема
[ , ВидСобственности ] ] ] )
```

Параметры

ДатНач (дата) – на утро этой даты рассчитываются остатки номенклатуры или складской карточки. Если параметр не указан, берётся переменная



«ДатНач», если же и она не определена, берётся дата начала рабочего периода из конфигурации системы.

НомерСклада (число) – номер склада, по которому считаются остатки. Если параметр не указан, то остатки берутся по всем складам хранения для номенклатуры и по текущему складу для складской карточки.

Поставщик (объект) – запись лица, от которого поступила номенклатура.

ВидОтгрузки (текст) – вид отгрузки приходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение». Если не указан, или указана пустая строка, берутся все приходные документы. Параметр имеет смысл только при партионном учёте.

Тема (текст) – тема приходных документов, по которым считаются остатки. Если параметр не указан, остатки считаются по всем приходным документам. Параметр имеет смысл только при партионном учёте.

ВидСобственности (текст) – название вида собственности из справочника «Виды собственности».

Возвращает

Объект – количественный остаток товара. Кроме того, функция устанавливает дополнительные переменные: «Кол_во» - количественный остаток товара, «СуммаЦен» - сумма остатка товара по отпускной цене, «СуммаСебест» - сумма остатка товара по себестоимости (приходной цене), «СуммаСебестБезНДС» - сумма остатка товара по себестоимости без НДС, « $\$$ СуммаЦен» - сумма остатка товара по отпускной цене в валюте; « $\$$ СуммаСебест» - сумма остатка товара по себестоимости (приходной цене) в валюте.

Примеры

Узнаем остатки номенклатуры по всем складам на начало указанной даты:

```
оЛицо = Лицо ("10901", "Номенклатура") ;  
оОстНаНачДня = оЛицо.Лицо_ОстНач (28.03.05) ;  
Сообщить (оОстНаНачДня.Кол_во)
```

См. также:

- [ОстКнци](#)
- [ОбрПрих](#)

- [ОбрРасх](#)

ОстКнц

Рассчитывает остатки номенклатуры на складах хранения этой номенклатуры, либо остатки по конкретной карточке складской картотеки на конец дня. Аналогичные остатки можно узнать, построив «Ведомость остатков» на конец дня.

Синтаксис

```
ЗаписьНоменклатуры.ОстКнц( ДатКнц[ , НомерСклада ,  
Поставщик , [ВидОтгрузки[ , Тема [ , ВидСобственности] ] ] )  
ЗаписьСкладскойКартотеки.ОстКнц( ДатКнц ,  
НомерСклада , Поставщик , [ВидОтгрузки[ , Тема [ ,  
ВидСобственности] ] ] )
```

Параметры

ДатКнц (дата) – на вечер этой даты рассчитываются остатки номенклатуры и складской карточки. Если параметр не указан, берётся переменная «ДатКнц», если же и она не определена, берётся дата начала рабочего периода из конфигурации системы.

НомерСклада (число) – номер склада, по которому считаются остатки. Если не указан, то остатки берутся по всем складам хранения для номенклатуры и по текущему складу для складской карточки.

Поставщик (объект) – запись лица, от которого поступила номенклатура.

ВидОтгрузки (текст) – вид отгрузки приходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение». Если не указан, или указана пустая строка, берутся все приходные документы. Параметр имеет смысл только при партионном учёте.

Тема (текст) – тема приходных документов, по которым считаются остатки. Если параметр не указан, остатки считаются по всем приходным документам. Параметр имеет смысл только при партионном учёте.

ВидСобственности (текст) – название вида собственности из справочника «Виды собственности».



Возвращает

Объект – количественный остаток товара. Кроме того, функция устанавливает дополнительные переменные «Кол_во» - количественный остаток товара, «СуммаЦен» - сумма остатка товара по отпускной цене, «СуммаСебест» - сумма остатка товара по себестоимости (приходной цене), «СуммаСебестБезНДС» - сумма остатка товара по себестоимости без НДС, «\$СуммаЦен» - сумма остатка товара по отпускной цене в валюте; «\$СуммаСебест» - сумма остатка товара по себестоимости (приходной цене) в валюте.

Примеры

Узнаем остатки товара на текущем складе на вечер указанной даты:

```
оКарточка = НайтиНаСкладеПоНомеру("10901", 6);  
оОстНаКнцДня = оКарточка.ОстКнц(16.02.05);  
Сообщить(оОстНаКнцДня.Кол_во)
```

См. также:

- [ОстНач](#)
- [ОбрПрих](#)
- [ОбрРасх](#)

ОбрПрих

Рассчитывает приход товара на указанный склад за период.

Синтаксис

```
ЗаписьНоменклатуры.ОбрПрих( ДатНач, ДатКнц[,  
НомерСклада[, Поставщик[, ВидОтгрузки[, Тема, [  
ВидСобственности]]]], [ВремяНач, ВремяКнц])  
ЗаписьСкладскойКартотеки.ОбрПрих( ДатНач, ДатКнц[,  
НомерСклада[, Поставщик[, ВидОтгрузки[, Тема, [  
ВидСобственности]]]], [ВремяНач, ВремяКнц])
```

Параметры

ДатНач, ДатКнц (дата) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.



НомерСклада (число) – номер склада, по которому рассчитывается приход товара. Если не указан, анализируется текущий склад (для складской картотеки), либо все склады (для номенклатуры).

Поставщик (объект) – запись поставщика, от которого поступил товар. Если не указан, то по всем поставщикам.

ВидОтгрузки (текст) – значение поля «Вид отгрузки» у анализируемых приходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение». Параметр имеет смысл только при партионном учёте.

Тема (текст) – тема анализируемых приходных документов. Если не указана, анализируются все приходные документы. Параметр имеет смысл только при партионном учёте.

ВидСобственности (текст) – название вида собственности из справочника «Виды собственности».

ВремяНач, ВремяКнц (время) – время прихода товара.

Возвращает

Объект – количество оприходованного товара. Кроме того, устанавливаются переменные «Кол_во» - количество оприходованного товара, «СуммаЦен» - сумма оприходованного товара по отпускной цене, «СуммаСебест» - сумма оприходованного товара по себестоимости (приходной цене), « $\$$ Суммацен» - сумма оприходованного товара по отпускной цене в валюте, « $\$$ СуммаСебест» - сумма оприходованного товара по себестоимости в валюте. Расчёт валютной себестоимости возможен только, если в приходных накладных заполнялось поле « $\$$ Себест» (использовался валютный учёт).

Примеры

Рассчитаем приход на склад №6 товара с номером «10901» за указанный период:

```
оЛицо = Лицо ( "10901" , "Номенклатура" ) ;  
оКолПрих = оЛицо . Лицо _ . ОбрПрих ( 01 . 01 . 05 , 25 . 03 . 05 , 6 ) ;  
Сообщить ( оКолПрих . Кол _ во ) ;
```

См. также:



- [ОстНач](#)
- [ОстКнц](#)
- [ОбрРасх](#)

ОбрРасх

Рассчитывает отгрузку товара со склада за указанный период.

Синтаксис

```
ЗаписьНоменклатуры.ОбрРасх( ДатНач, ДатКнц[,  
НомерСклада, [Поставщик[, Покупатель]], [  
ВидОтгрузкиРасх[, [ТемаПрх[, ТемаРасх[,  
ВидСобственности]]]]], [ВремяНач, ВремяКнц] )  
ЗаписьСкладскойКартотеки.ОбрРасх( ДатНач, ДатКнц[,  
НомерСклада, [Поставщик[, Покупатель]], [  
ВидОтгрузкиРасх[, [ТемаПрх[, ТемаРасх[,  
ВидСобственности]]]]], [ВремяНач, ВремяКнц] )
```

Параметры

ДатНач, ДатКнц (дата) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.

НомерСклада (число) – номер склада, по которому рассчитывается расход. Если не указан, анализируется текущий склад (для складской картотеки), либо все склады (для номенклатуры).

ВидОтгрузкиРасх (текст) – значение поля «Вид отгрузки» у анализируемых расходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение», «Возврат поставщику», «Реализация по лицу», «Возврат по теме», «Реализация по теме», «Замена по гарантии», «Неопределенный». Параметр имеет смысл только при партионном учёте.

ТемаПрих (текст) – тема анализируемых приходных документов, по которым отбирается расход. Если не указана, анализируются все приходные документы. Параметр имеет смысл только при партионном учёте.

ТемаРасх (текст) – тема анализируемых расходных документов. Если не указана, анализируются все документы. Параметр имеет смысл только при партионном учёте.

ВидСобственности (текст) – название вида собственности из справочника «Виды собственности».

ВремяНач, ВремяКнц (время) – время отгрузки товара.

Возвращает

Объект – количество отгруженного товара. Кроме того, устанавливаются переменные «Кол_во» - количество отгруженного товара, «СуммаЦен» - сумма отгруженного товара по отпускной цене, «СуммаСебест» - сумма отгруженного товара по себестоимости (по приходной цене), « $\$$ СуммаЦен» - сумма отгруженного товара по валютной отпускной цене.

Примеры

Рассчитать расход по товару с номером «10901» по складу №6:

```
оЛицо = Лицо ( "10901", "Номенклатура" );
оН = оЛицо.Лицо_.ОбрРасж ( 6 );
```

См. также:

- [ОстНач](#)
- [ОстКнц](#)
- [ОбрПрих](#)

Выписано

Рассчитывает количество выписанного товара по исходящим счетам.

Синтаксис

```
ЗаписьНоменклатуры.Выписано ([ДатНач, ДатКнц] [,
НомерСклада] [, Покупатель [, Грузополучатель]] [,
ВидОтгрузки [, Оплачено [, Тема]])
ЗаписьСкладскойКартотеки.Выписано ([ДатНач, ДатКнц] [,
НомерСклада] [, Покупатель [, Грузополучатель]] [,
ВидОтгрузки [, Оплачено [, Тема]])
```

Параметры

ДатНач, ДатКнц (дата) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соот-



ветственно, если же и их нет, то берётся рабочий период из конфигурации системы.

НомерСклада (число) – номер склада, по которому рассчитывается выписанный товар. Если не указан, то по всем складам.

Покупатель (объект) - запись покупателя, которому выписан товар. Если не указан, то по всем покупателям.

Грузополучатель (объект) - запись грузополучателя. Если не указан, то по всем грузополучателям.

ВидОтгрузки (текст) – значение поля «Вид отгрузки» у анализируемых приходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение». Параметр имеет смысл только при партионном учёте.

Оплачено (текст) – определяет, брать только оплаченные документы – значение "Оплачено", или все документы – пустая строка "".

Тема (текст) – тема выбираемых документов. Если не указана, выбираются все документы.

Возвращает

Объект – количество товара, выписанного исходящими счетами за указанный период. Кроме того, устанавливаются переменные «Кол_во» - количество выписанного товара, «СуммаЦен» - сумма выписанного товара по отпускной цене, «\$Сумма» - сумма выписанного товара по отпускной цене в валюте.

Комментарий

Функция рассчитывает количество товара по всем выписанным исходящим счетам, независимо от того, закрыты они или нет, резервировался товар в счетах или нет, отгружался он со склада или нет.

Примеры

Узнаем количество товара, выписанного за указанный период со склада №6:

```
оЛицо = Лицо ("10901", "Номенклатура");  
оВыписано = оЛицо.Лицо_Выписано (01.03.05, 28.03.05,  
6);  
Сообщить (оВыписано.Кол_во);
```

См. также:

- [Заказано](#)

Заказано

Рассчитывает количество заказанного товара по входящим счетам.

Синтаксис

```
ЗаписьНоменклатуры.Заказано ([ДатНач, ДатКнц] [,  
Склад] [, Поставщик [, Грузоотправитель]] [,  
ВидОтгрузки [, Оплачено [, Тема]]])  
ЗаписьСкладскойКартотеки.Заказано ([ДатНач, ДатКнц] [,  
Склад] [, Поставщик [, Грузоотправитель]] [,  
ВидОтгрузки [, Оплачено [, Тема]]])
```

Параметры

ДатНач, ДатКнц (дата) – даты, определяющие анализируемый период. Если даты не указаны, то берутся переменные «ДатНач» и «ДатКнц» соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.

Склад (число) – номер склада, по которому рассчитывается заказанный товар. Если не указан, то по всем складам.

Поставщик (объект) - запись покупателя, которому выписан товар. Если не указан, то по всем покупателям.

Грузоотправитель (объект) - запись грузополучателя. Если не указан, то по всем грузоотправителям.

ВидОтгрузки (текст) – значение поля «Вид отгрузки» у анализируемых приходных документов, возможные варианты: «Расход/Приход», «Внутренние нужды», «Брак/Бой/Списание», «На реализацию/хранение». Параметр имеет смысл только при партионном учёте.

Оплачено (текст) – определяет, брать только оплаченные документы – значение "Оплачено", или все документы – пустая строка "".

Тема (текст) – тема выбираемых документов. Если не указана, выбираются все документы.



Возвращает

Объект – количество товара, заказанного входящими счетами за указанный период. Кроме того, устанавливаются переменные «Кол_во» - количество заказанного товара, «СуммаЦен» - сумма заказанного товара по отпускной цене, «\$Сумма» - сумма заказанного товара по отпускной цене в валюте.

Комментарий

Функция рассчитывает количество товара по всем выписанным входящим счетам, независимо от того, закрыты они или нет, резервировался товар в счетах или нет, отгружался он со склада или нет.

Примеры

Узнаем общее количество товара, заказанного за указанный период:

```
оЛицо = Лицо ( "10901" , "Номенклатура" ) ;  
оЗаказано = оЛицо . Лицо _ . Заказано ( 01 . 03 . 05 ,  
01 . 04 . 05 ) ;  
Сообщить ( оЗаказано . Кол _ во ) ;
```

См. также:

- [Выписано](#)

ДляВсех(Остатков)

Перебирает остатки товара в разрезе партии на указанную дату.

Синтаксис

```
ДляВсех ( ЗаписьСкладскойКартотеки . Остатков ( [ДатаКнц] )
```

Параметры

ДатаКнц (дата) – дата, на конец которой выводятся остатки по партии. Если дата не указана, то остатки выводятся на конец рабочей даты.

Комментарии

Внутри цикла создаётся контекстный объект «**ОстатокПартии**», в котором хранится вся информация об остатке по партии. Для обращения к полям этого объекта используется выражение типа «**ОстатокПартии.Кнц**» (остаток партии). Если партия распределена между МОЛ, то функция бу-

дет выводить остаток партии в отдельности по каждому МОЛ. При этом склад должен иметь признак именного списания (в карточке склада д.б. установлен флаг «Склад реализации/именного списания»).

Выполнение функции аналогично построению отчета «Остатки по партиям» (<Shift+F2>) на складской карточке. Все поля, отображаемые в отчете, будут доступны и в объекте «**ОстатокПартии**».

Возвращает

Ничего не возвращает.

Примеры

Необходимо узнать остатки партии товара и сумма себестоимости со склада именного списания на текущую дату:

```
oКарточка = НайтиНаСкладеПоНомеру ("Т0_03" , 6) ;
ДляВсех (oКарточка . Остатков ( ) )
{
    Сообщить (ОстатокПартии . Кнц) ;
    Сообщить (ОстатокПартии . СебКнц) ;
}
```

ДляВсех(ХарактеристикПартии)

Перебирает характеристики партии.

Синтаксис

```
ДляВсех (ПартииПриходы . ХарактеристикиПартии ( ) )
```

Параметры

Не указываются.

Возвращает

Ничего не возвращает.

Комментарии

Функция вызывается в контексте партии. В отличие от функции **ДляВсех(Параметров)** в результирующем контексте «ПараметрПартии», кроме фактических значений, содержит и соответствующие нормативные значе-



ния характеристик. Набор характеристик партии и их нормативные значения задаются в справочнике «Номенклатура».

Кроме того, функция перебирает все характеристики партии, даже те, для которых не заданы фактические значения. Для того, чтобы определить задано ли фактическое значение характеристики, следует использовать переменную "Откуда". Если Откуда = Родной, то фактическое значение характеристики партии введено, в противном случае нет.

Примеры

Чтобы определить, задано ли фактическое значение характеристики, можно использовать следующий фрагмент:

```
функция НаДиалогСохранить ()
{
  ДляВсех (Наименований () )
  {
    ДляВсех (Наим . ПартияПриходы . ХарактеристикиПартии)
    {
      Если (ПараметрПартии . Откуда=="Родной")
        Сообщить (ПараметрПартии . Название+" :
"+ПараметрПартии . Значение)
    }
  }
  вернуть Следфункция ;
}
```

См. также:

- [ДляВсех\(Параметров\)](#)

Складские документы

СчетУчета

Возвращает счёт учёта наименования, указанного в документах движения по складу.

Синтаксис

```
Запись . СчетУчета ()
```

Параметры

Не указываются.

Комментарии

Функция вызывается в контексте наименования складских документов. По сути, определяющим звеном в работе функции является определение вида собственности у партии прихода. Вследствие этого происходит следующее:

Если вид собственности не указан - программа обращается к карточке склада и определяет установленную на нём схему учёта; программа анализирует карточку наименования: в карточке наименования обращается к типу номенклатуры; в списке схем учёта у типа номенклатуры находит ту, которая указана для склада и соответственно этой схеме определяется нужный счёт учёта.

Если указан вид собственности – программа действует аналогичным образом, что и в предыдущем случае, только в процесс ещё вовлекается вид собственности, указанный на партии. В соответствии с этим программа должна найти схему учёта с указанным на складском документе видом собственности. И только когда найдётся нужное сочетание, функция возвратит нужный счёт учёта.

Возвращает

Строка – номер счёта учёта наименования из утверждённого плана счетов.

Склад

Возвращает склад, указанный в поле «Склад» у текущего документа, например, приходной накладной.

Синтаксис

Документ . Склад ()

Параметры

Не указываются.



Возвращает

Объект, содержащий запись склада, указанного в поле «Склад» текущего документа. Если у документа нет такого поля (например, у платёжного поручения), то возвращается пустой объект.

Комментарии

Для работы функции необходимо наличие контекстного объекта с записью таблицы «Документ».

Примеры

Узнаем значения параметров у найденного склада:

```
ДляВсех (Документов ("НаклРасх"))
{
    оСклад = Склад ();
    Если (оСклад.Название=="Основной склад")
    {
        ДляВсех (оСклад.Параметров ())
            Сообщить ("Значение параметра " +
                Параметр.Название + " равно " + Параметр.Значение);
    }
}
```

См. также:

- [НайтиСклад](#)

НайтиПрайс

Возвращает запись из таблицы видов цен, связанного с прайсом.

Синтаксис

```
НайтиПрайс (Документ [ , Дата ] )
```

Параметры

Документ (объект) – любой складской документ, например, расходная накладная, имеющий связь с видом цен, по которому существует связь с прайсом. Причем, если несколько видов цен имеют прайсы, то берется первый попавшийся.

Дата (дата) – берется действующий на указанную дату прайс по данному виду цен, который присутствует в складском документе. Если дата не указана, берется прайс на дату документа.

Возвращает

Объект – запись из таблицы видов цен. Если на указанную дату прайса не существует или существует пустой прайс (рассматривается как исключение из правил), то функция возвратит пустой объект.

Примеры

Узнаем номер и дату прайса по расходным накладным:

```
ДляВсех (Документов ("НаклРасх" , "" , "ОПТ" ) )
{
    оПрайс = НайтиПрайс (Документ) ;
    Если (оПрайс)
        Сообщить ("Прайс № " + оПрайс . Документ_ . Номер +
" от " +
                    оПрайс . Документ_ . Дата) ;
}
```

См. также:

- [Документ](#)

ЕстьИИспользуется

Проверяет, зарегистрировано ли наименование прайс-листа поставщика/конкурента в базе СБИС++ и помечено ли оно как используемое.

Синтаксис

```
Прайс . ЕстьИИспользуется (НаименованиеПрайса)
```

Параметры

НаименованиеПрайса (объект) – запись наименования прайс-листа поставщика/конкурента. Из этой записи берутся поля «Код» (код наименования у поставщика/конкурента) и «Комментарий» (название наименования у поставщика/конкурента). По этим данным и ищется наименование в базе СБИС++.

Комментарий

Если наименование по указанным данным найдено в СБиС++, и хотябы одна складская карточка используется (установлен признак «Использование»), то в прайс-листе устанавливается связь с номенклатурной карточкой (поле «НоменклатураЦены»). Если наименование в СБиС++ не зарегистрировано, то проверяется, не помечено ли оно как незагружаемое. Если «Да», то на наименовании прайс-листа устанавливается соответствующий признак.

Возвращает

Да (логическое значение) – если наименование зарегистрировано и помечено как используемое.

Нет (логическое значение) – если наименование не зарегистрировано и помечено как незагружаемое.

Примеры

Проверим, зарегистрировано ли указанное наименование поставщика в базе данных СБиС++. Если "Да", то узнаем его наименование в справочнике "Номенклатура":

```
функция НаДокументЗакрыть ()
{
    оЗапись=Выборка ("Наименования прайса поставщика") ;
    оЗапись . Код=123 ;
    Если (ЕстьИспользуется (оЗапись) )
        Сообщить (ОЗАПИСЬ . НоменклатураЦены . Наименование) ;
}
```

РассчитатьЦену

Рассчитывает цену наименования документа.

Синтаксис

```
Запись . РассчитатьЦену ()
```

Параметры

Не указываются.

Комментарии

Данная функция работает только в контексте документа.

Возвращает

Да (логическое значение) – удалось рассчитать цену наименования.

Нет (логическое значение) – не удалось рассчитать цену наименования или функция была вызвана не в контексте документа.

Пример

При закрытии документа необходимо рассчитать цену каждого наименования документа:

```
функция НаДокументЗакрыть ()
{
    ДляВсех (Наименований ())
    {
        Наим . РассчитатьЦену ();
    }
}
```

См. также:

- [ОчиститьВидыЦен](#)
- [РассчитатьЦены](#)

НайтиПоКлючевымСловам

Производит поиск записей по ключевым словам в справочнике «Номенклатура». Список ключевых слов находится в карточке номенклатуры на вкладке «Ключевые слова».

Синтаксис

НайтиПоКлючевымСловам (КлючевыеСлова , СтрогоЛи)

Параметры

КлючевыеСлова (текст) – строка, содержащая через пробел слова, по которым нужно вести поиск. Регистр не важен.

СтрогоЛи (логическое значение) – «Да» означает искать только строгие соответствия. По умолчанию считается равным «Да».



Возвращает

Массив, содержащий позиции лиц, соответствующих найденным карточкам номенклатуры.

Комментарии

Функция работает аналогично поиску по ключевым словам, вызываемому по <Ctrl+F>.

Примеры

Найдем в справочнике «Номенклатура» запись по ключевым словам. Если заданным ключевым словам соответствует единственная запись, выведем ее название. Это делается так:

```
перем олицо = Таблица ("Лица" );
мНоменклатура=НайтиПоКлючевымСловам ("зонт мужской
черный" );
Если (Размер (мНоменклатура) == 1)
{
    олицо.Загрузить (мНоменклатура [1] );
    Сообщить (олицо. Лицо_. Название) ;
}
```

См. также:

- [НайтиНаСкладе](#)
- [НайтиНаСкладеПоНомНомеру](#)
- [НайтиПоКлючевымСловамВКартотеке](#)
- [НайтиПоКлючевымСловамСочетаниями](#)
- [РазбитьНаКлючевыеСлова](#)
- [ЗаполнитьКлючевымиСловами](#)

НайтиПоКлючевымСловамВКартотеке

Производит поиск записей по ключевым словам в справочнике «Складская картотека». Список ключевых слов находится в карточке картотеки на вкладке «Ключевые слова».

Синтаксис

НайтиПоКлючевымСловамВКартотеке (КлючСлова ,



НомераСкладов, **Используемые** [, **СтрогоЛи**])

Параметры

КлючСлова (текст) – строка, содержащая через пробел слова, по которым нужно вести поиск. Регистр не важен.

НомераСкладов (массив) – массив с номерами складов, по которым нужно вести поиск.

Используемые (логическое значение) - «Да» искать только среди используемых карточек, иначе искать везде.

СтрогоЛи (логическое значение) – «Да» означает искать только строгие соответствия. По умолчанию считается равным «Да».

Возвращает

Массив, содержащий адреса номеров карточек «Складской картотеки».

Комментарии

Функция работает аналогично поиску по ключевым словам, вызываемому по <Ctrl+F>.

Примеры

Необходимо найти все используемые карточки по складу с номером 1 справочника «Складская картотека» с заданными ключевыми словами:

```
перем пНомераСкладов [1] ;  
пНомераСкладов [1] = 1 ;  
мНоменклатура=НайтиПоКлючевымСловамВКартотеке ("удлин  
итель телефонный 2м", пНомераСкладов, Да) ;  
# в мНоменклатура адреса карточек, содержащие  
заданные ключевые слова
```

См. также:

- [НайтиНаСкладе](#)
- [НайтиНаСкладеПоНомНомеру](#)
- [НайтиПоКлючевымСловам](#)
- [НайтиПоКлючевымСловамСочетаниями](#)



- [РазбитьНаКлючевыеСлова](#)
- [ЗаполнитьКлючевымиСловами](#)

НайтиПоКлючевымСловамСочетаниями

Выполняет поиск максимально подходящих по ключевым словам записей в справочнике «Номенклатура». Список ключевых слов находится в карточке номенклатуры на закладке «Ключевые слова».

Синтаксис

НайтиПоКлючевымСловамСочетаниями (КлючевыеСлова)

Параметры

КлючевыеСлова (текст) – строка, содержащая через пробел слова, по которым нужно вести поиск. Регистр не важен.

Возвращает

Массив, содержащий позиции лиц, соответствующих найденным карточкам номенклатуры.

Примеры

Требуется найти карточку номенклатуры с максимально подходящим к указанному набором ключевых слов:

НайтиПоКлючевымСловамСочетаниями ("зонт мужской") ;

См. также:

- [НайтиНаСкладе](#)
- [НайтиНаСкладеПоНомНомеру](#)
- [НайтиПоКлючевымСловам](#)
- [НайтиПоКлючевымСловамВКартотеке](#)
- [РазбитьНаКлючевыеСлова](#)
- [ЗаполнитьКлючевымиСловами](#)

РазбитьНаКлючевыеСлова



Разбивает указанную строку на отдельные слова.

Синтаксис

НайтиПоКлючевымСловамСочетаниями (КлючевыеСлова)

Параметры

КлючевыеСлова (текст) – строка, которую необходимо разбить на ключевые слова. Резделителями являются символы «"», «,», «:», «:», «?», «.», «/», «(», «)», \n, \r, \\. .

Возвращает

Массив, содержащий отдельные слова.

Примеры

Требуется разбить строку с комментариями в карточке складской картотеки на ключевые слова на:

```
# предварительно следует извлечь запись нужной  
карточки номенклатуры в переменную oНаим  
mСлова = РазбитьНаКлючевыеСлова (oНаим.Комментарий)
```

См. также:

- [ЗаполнитьКлючевымиСловами](#)
- [НайтиПоКлючевымСловам](#)
- [НайтиПоКлючевымСловамВКартотеке](#)
- [НайтиПоКлючевымСловамСочетаниями](#)

ЗаполнитьКлючевымиСловами

Обновляет автоматически сформированные ключевые слова у записи лица. Список ключевых слов находится в диалоге на вкладке «Ключевые слова».

Синтаксис

ЗаполнитьКлючевымиСловами (Лицо , КлючевыеСлова)

Параметры



Лицо (объект) – объект таблицы Лица.

КлючевыеСлова (текст) – строка, содержащая ключевые слова, которые должны быть добавлены. Регистр не важен.

Возвращает

Да (логическое значение) – ключевые слова были добавлены к записи лица.

Нет (логическое значение) – ключевые слова не были добавлены.

Примеры

Необходимо добавить к списку ключевых слов карточки номенклатуры все слова наименования товара:

оЛицо = Лицо (1001, "Номенклатура") ;
ЗаполнитьКлючевымиСловами (оЛицо ,
оЛицо . Наименование) ;

См. также:

- [РазбитьНаКлючевыеСлова](#)
- [НайтиПоКлючевымСловам](#)
- [НайтиПоКлючевымСловамВКартотеке](#)
- [НайтиПоКлючевымСловамСочетаниями](#)
- [НайтиПоКлючевымСловамСТипом](#)

Производство

НайтиТехоперацию

Ищет техоперацию в справочнике «Виды работ/Техпроцессы» по указанному виду работы.

Синтаксис

НайтиТехоперацию (ВидРаботы [, Номер])

Параметры

ВидРаботы (текст) – код вида работы, по которому будет производиться поиск техоперации.

НомНомер (текст) – номенклатурный номер изделия, с которым связана искомая техоперация. Если параметр указан, то функция будет искать техоперацию, привязанную только к указанному изделию.

Возвращает

Объект – запись техоперации из справочника, найденную по указанному виду работы.

Примеры

Найдем описание техоперации по указанному изделию с номером «1001»:

```
oТехОпер = НайтиТехоперацию("01РАСП", "1001");
```

ТарифРаботы

Возвращает запись тарифа работы по документу на указанную дату. Данные по тарифу работы указаны в описании техоперации справочника «Виды работ/Техоперации».

Синтаксис

```
Запись . ТарифРаботы (Дата [ , Номенклатура ] )
```

Параметры

Дата (дата) – дата, на которую необходимо узнать показатели тарифа работы (общего или на конкретное изделие).

Номенклатура (объект) – запись номенклатуры, для которой указаны тарифы на закладке «Тарифы и ресурсы изделий». Если параметр не указан, запись тарифа определяется на закладке «Общие тарифы и ресурсы».

Комментарии

Функция работает в контексте документа на выполнение конкретной работы, например, наряд-заказ. В документе задается описание этой работы с указанием вида работы, который содержит описание общих тарифов и тарифов конкретного изделия. В зависимости от количества параметров,

передаваемых при вызове функции, записи тарифа берутся либо на закладке «Общие тарифы и ресурсы», либо на закладке «Тарифы и ресурсы изделий».

Возвращает

Объект – запись тарифа работы на указанную дату.

Примеры

Узнаем тариф работы, описанной в документе наряд-заказа, изделия с номером «10102» на указанную дату:

```
ДляВсех(Записей("Работы в наряде", Документ))
{
    оНом = Лицо("10102", "Номенклатура");
    оТариф = Запись.ТарифРаботы(Документ.Дата);
    оТариф2 = Запись.ТарифРаботы(07.04.05,
    оНом.Лицо_);
    Отладить();
}
```

ДляВсех(Компонентов)

Перебирает все компоненты, входящие в состав изделия.

Синтаксис

```
ДляВсех(Запись.Компонентов([СпособРаскрытия][, ПолныйСостав][, ПоЦехам]))
```

Параметры

СпособРаскрытия (текст) – определяет, как раскрывать состав компонентов, входящих в состав изделия:

”**Без раскрытия**” - состав компонентов не раскрывается.

”**Раскрыть комплекты**” - раскрываются составы компонентов с типом «Комплект».

”**Раскрыть всё**” - раскрываются составы всех компонентов (используется по умолчанию, если параметр не указан). При этом раскрываемые компоненты в результирующую выборку не попадают.

ПолныйСостав (текст) – если в этом параметре указана строка ”Полный состав”, то отбираются все компоненты вне зависимости от условий их

вхождения в состав. Если параметр не указан, то функция перебирает только компоненты с условием «Включено в состав».

ПоЦехам (текст) – если в этом параметре указана строка ”По цехам”, то при отборе компонентов, в поле «Цех» определяется, в каких производственных подразделениях (цехах) они списываются. Если параметр не указан или указана пустая строка, расцеховка не производится.

Комментарии

Функция устанавливает контекстный объект с именем «Товар», который содержит запись о компоненте.

Примеры

Перебрать все компоненты, входящие в состав изделия с раскрытием составов комплектов:

```
ДляВсех(оИзделие . Компонентов (“Раскрыть комплекты”))  
Сообщить(Название) ;
```

См. также:

- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)

ДляВсех(КомпонентовНаименования)

Перебирает все компоненты наименования плана производства в документе «План производства».

Синтаксис

```
ДляВсех(Запись . КомпонентовНаименования ( ) )
```

Параметры

Не указываются.

Комментарии

Функция работает только в контексте наименования документа планирования производства. При этом создаются два контекстных объекта с оди-

наковым названием «Наим»: наименование документа и наименование компонента плана производства. Чтобы в цикле функции обратиться к данным наименования документа, нужно переобозначить первый объект «Наим», например, оНаим = Наим.

Возвращает

Ничего не возвращает.

Примеры

Узнаем названия всех компонентов, из которых будет изготовлено изделие, указанное в текущем документе планирования:

```
ДляВсех (Наименований)  
  ДляВсех (КомпонентовНаименования ( ) )  
    Сообщить (Название) ;
```

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)
- [ДляВсех\(Приходов\)](#)
- [ДляВсех\(СписанийСборки\)](#)

ДляВсех(РаботНаименования)

Перебирает все работы по наименованию выпуска в документе «Акт выпуска готовой продукции».

Синтаксис

```
ДляВсех (Запись . РаботНаименования ( ) )
```

Параметры

Не указываются.

Комментарии

Функция работает в контексте наименования выпуска и перебирает все работы по данному наименованию, перечисленные на закладке «Работы». Чтобы эта закладка появилась, необходимо, чтобы на папке актов выпуска

готовой продукции на закладке «Доп. настройки» был установлен флаг «Учитывать работы». Внутри цикла создается контекстный объект «Работа», к данным которого можно в дальнейшем обращаться.

Возвращает

Ничего не возвращает.

Примеры

Узнаем название и исполнителя работы выпускаемого изделия:

```
функция НаДокументЗакрыть ( )
{
    ДляВсех (Наименований ( ) )
        ДляВсех (РаботНаименования ( ) )
            {
                Сообщить (Наименование) ;
                Сообщить (Исполнитель) ;
            }
}
```

См. также:

- [ПереопределитьРаботы](#)
- [ДляВсех\(Наименований\)](#)

ДляВсех(СписанийНаименования)

Перебирает все списания наименования выпуска в документах акта выпуска готовой продукции.

Синтаксис

```
ДляВсех (Запись .СписанийНаименования ( ) )
```

Параметры

Не указываются.

Комментарии

Функция работает только в контексте наименования документа акта выпуска готовой продукции. В производстве при выпуске изделия указыва-



ются составляющие, из которого оно изготавливается, и отходы (их может и не быть), полученные в результате выпуска этого изделия. Так вот функция «ДляВсех(СписанийНаименования)» перебирает составляющие готового изделия, которые описываются в таблице «Списания выпуска».

При этом создаются два контекстных объекта с одинаковым названием «Наим»: наименование выпуска и наименование списания. Чтобы в цикле функции обратиться к данным наименования документа, нужно переобозначить первый объект «Наим», например, оНаим = Наим.

Возвращает

Ничего не возвращает.

Примеры

Узнаем состав изделия, указанного в акте выпуска готового изделия:

```
ДляВсех (Наименований ( ) )  
    ДляВсех (СписанийНаименования ( ) )  
        Сообщить (Наименование)
```

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)
- [ДляВсех\(Приходов\)](#)
- [ДляВсех\(СписанийСборки\)](#)

ДляВсех(ОтходовНаименования)

Перебирает все отходы наименования выпуска в документах акта выпуска готовой продукции.

Синтаксис

```
ДляВсех (Запись .ОтходовНаименования ( ) )
```

Параметры

Не указываются.

Комментарии

Функция работает только в контексте наименования документа акта выпуска готовой продукции. В производстве при выпуске изделия указываются составляющие, из которого оно изготавливается, и отходы (их может и не быть), полученные в результате выпуска этого изделия. Так вот функция «ДляВсех(ОтходовНаименования)» перебирает записи отходов готового изделия, которые описываются в таблице «Отходы выпуска».

При этом создаются два контекстных объекта с одинаковым названием «Наим»: наименование выпуска и наименование отхода. Чтобы в цикле функции обратиться к данным наименования документа, нужно переобозначить первый объект «Наим», например, oНаим = Наим.

Возвращает

Ничего не возвращает.

Примеры

Узнаем общую сумму отходов изделия, указанного в текущем акте выпуска готовой продукции:

```
перем пСумма ;
ДляВсех (Наименований ( ) )
{
    ДляВсех (ОтходовНаименования ( ) )
        пСумма+=СуммаСебест ;
}
Сообщить (пСумма) ;
```

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(СписанийНаименования\)](#)

ДляВсех(Приходов)

Перебирает все наименования разборки в документах акта разборки.

Синтаксис



ДляВсех (Запись . Приходов ())

Параметры

Не указываются.

Комментарии

Функция применяется в тех случаях, когда необходимо произвести разуконплектацию изделия, т.е. вернуть составляющие этого изделия на прежние места хранения, и работает только в контексте наименования документа акта разборки. При этом создаются два контекстных объекта с одинаковым названием «Наим»: наименование выпуска и наименование отхода. Чтобы в цикле функции обратиться к данным наименования документа, нужно переобозначить первый объект «Наим», например, oНаим = Наим.

Возвращает

Ничего не возвращает.

Примеры

Узнаем составляющие разборки, указанной в текущем акте разборки изделия:

```
ДляВсех (Наименований ( ) )
    ДляВсех (Приходов ( ) )
        Сообщить (Наименование)
```

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)

ДляВсех(СписанийСборки)

Перебирает все наименования списаний сборки изделия с типом «Сборка», т.е. в карточке изделия на закладке «Общие» указывается тип «Сборка».

Синтаксис

ДляВсех (Запись .СписанийСборки ())

Параметры

Не указываются.

Комментарии

При списании наименований сборки важно указать партионные остатки по этому изделию, которые формируются при закрытии документа акта выпуска по этому изделию. Программа извлечет необходимые данные, ссылаясь на этот документ. Имеется одна особенность: если в акте выпуска мы указываем первоначальную сборку изделия и хотим некоторые компоненты добавить или удалить, то состав сборки наименования выпуска будет изменен.

Функция работает только в контексте наименования документа. При этом создаются два контекстных объекта с одинаковым названием «Наим»: наименование документа и наименование списания сборки. Чтобы в цикле функции обратиться к данным наименования документа, нужно переобозначить первый объект «Наим», например, oНаим = Наим.

Возвращает

Ничего не возвращает.

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)
- [ЕстьСостав](#)
- [ПереопределитьСостав](#)

ЕстьСостав

Проверяет наличие компонентов в составе изделия.

Синтаксис



Товар.ЕстьСостав ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – если в состав изделия входит хотя бы один компонент. Функция вернет "Да" даже в том случае, если на текущую дату все компоненты изделия не актуальны или не введены в действие

Нет (логическое значение) – если изделие не является составом.

Комментарии

Функция работает в контексте записи номенклатуры или карточки складской картотеки. В противном случае будет выдано сообщение об ошибке.

Примеры

Необходимо узнать, является ли изделие с заданным номером составом:

```
oТовар = Выборка ("Номенклатура" );  
oТовар.НомНомер = Номер;  
Найти (oТовар, "НомНомер" ); # извлекаем запись  
изделия с заданным номером из номенклатуры  
Сообщить (oТовар.ЕстьСостав) ;
```

См. также:

- [ДляВсех\(Компонентов\)](#)
- [ДляВсех\(КомпонентовНаименования\)](#)
- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)
- [ДляВсех\(СписанийСборки\)](#)
- [ПереопределитьСостав](#)

ПереопределитьСостав

Очищает состав текущего изделия и вставляет состав вновь, разрешая возможные неопределенности в пользу идущих первыми компонентов.

Синтаксис

Наим . ПереопределитьСостав (Документ)

Параметры

Документ (объект) – акт выпуска готовой продукции.

Возвращает

Ничего не возвращает.

Комментарии

Функция работает в контексте наименования акта выпуска. В противном случае будет выдано сообщение об ошибке.

Функция действует аналогично кнопке «Состав» в диалоге наименования акта выпуска: очищает состав, указанный в наименовании выпуска (закладка «Состав») и вставляет состав, заданный в складской карточке этого наименования (закладка «Описание изделия», таблица «Состав»).

Примеры

Допустим, имеется акт выпуска готовой продукции с номером 1. Необходимо для всех изделий, имеющих состав, переопределить его:

```
oАкт = Выборка ("Акты выпуска" );  
oАкт.Номер = 1 ;  
Найти (oАкт , "Номер" ) ;  
ДляВсех (oАкт . Наименований ( ) )  
    Если (Товар . ЕстьСостав )  
        Наим . ПереопределитьСостав (Документ) ;
```

См. также:

- [ДляВсех\(СписанийНаименования\)](#)
- [ДляВсех\(ОтходовНаименования\)](#)
- [ЕстьСостав](#)
- [ПереопределитьРаботы](#)

ПереопределитьРаботы

Очищает перечень работ по текущему изделию и вставляет вновь на основе техпроцесса изготовления, приведенного в описании изделия (справочник «Описания изделий», диалог «Изделие», закладка «Техпроцесс», или складская и номенклатурная карточки, закладка «Работы»), разрешая возможные неопределенности в пользу идущих первыми работ.

Синтаксис

Наим . ПереопределитьРаботы (Документ)

Параметры

Документ (объект) – акт выпуска готовой продукции, в котором находится наименование с перечислением работ по нему.

Возвращает

Ничего не возвращает.

Комментарии

Работы указываются в диалоге наименования выпуска на закладке "Работы". Чтобы эта закладка появилась, необходимо, чтобы на папке актов выпуска готовой продукции на закладке "Доп. настройки" был установлен флаг "Учитывать работы".

Функция вызывается в контексте наименования выпуска и действует по аналогии с нажатием кнопки "Работы" в наименовании выпуска изделия: очищает перечень работ, указанный в наименовании выпуска (закладка "Работы") и вставляет перечень работ, заданный в описании изделия, складской или номенклатурной карточке этого наименования.

Примеры

Необходимо при закрытии акта для всех выпускаемых изделий переопределить работы:

```
функция НаДокументЗакрыть ( )
{
    ДляВсех (Наименований ( ) )
        Наим . ПереопределитьРаботы (Документ) ;
}
```

См. также:

- [ДляВсех\(РаботНаименования\)](#)

- [ПереопределитьСостав](#)

Ценообразование

ПолучитьЦеныПоПрайсу

Возвращает объект с ценами из прайса для указанной номенклатуры.

Синтаксис

```
ПолучитьЦеныПоПрайсу (ЗапНом, КодУпаковки,  
ЗапВидЦены[, Дата])  
ПолучитьЦеныПоПрайсу (АдресЗапНом, КодУпаковки,  
АдресЗапВидЦены[, Дата])
```

Параметры

ЗапНом (объект) – запись номенклатуры, для которой нужно получить цены из прайса.

АдресЗапНом (целое) – адрес записи номенклатуры, для которой нужно получить цены из прайса.

КодУпаковки (целое) – код упаковки, указанный в карточке номенклатуры. Если кода упаковки нет, то надо указать в качестве параметра «0».

ЗапВидЦены (объект) – запись вида цен из таблицы «Виды цен», связанного с прайсом, по которому берутся цены.

АдресЗапВидЦены (целое) – адрес записи вида цен из таблицы «Виды цен», связанного с прайсом, по которому берутся цены.

Дата (дата) - берется действующий на указанную дату прайс по данному виду цен. Если дата не указана, берется прайс, действующий на текущую дату.

Возвращает

Объект – объект с ценами из прайса для указанной номенклатуры.

Примеры

Узнаем цены из прайса для указанной номенклатуры:



```
оВидЦены = Таблица ("Виды цен");  
оВидЦены.Загрузить (4);  
оЛицо = Лицо ("ОС1", "Номенклатура");  
оЦены = ПолучитьЦеныПоПрайсу (оЛицо.Лицо_, 0,  
оВидЦены);  
Сообщить ("Цена1: " + оЦены.Цена1 + "<br>" + "Цена2: " +  
оЦены.Цена2 + "<br>" + "Цена3: " + оЦены.Цена3);
```

См. также:

- [НайтиПрайс](#)
- [ДобавитьВидЦены](#)
- [ДобавитьВидыЦенКарты](#)
- [ОчиститьВидыЦен](#)
- [РассчитатьЦены](#)

ДобавитьВидЦены

Добавляет указанный вид цены.

Синтаксис

Документ . **ДобавитьВидЦены** (*ИмяВидаЦены*)

Параметры

ИмяВидаЦены (текст) – имя цены, которое нужно добавить в выборку «Виды цен». Стоит вспомнить, что поле «Имя» должно быть уникально для каждого нового вида цены в отличие от поля «Название». Если данный вид цены уже добавлен в документ, то выдаст сообщение об ошибке.

Возвращает

Да (логическое значение) – удалось добавить указанный вид цены.

Нет (логическое значение) – не удалось добавить указанный вид цены.

Комментарии

Работает только в контексте документа, содержащего выборку «Виды цен».

Пример

Необходимо добавить вид цены с именем «Скидка10» в определенной расходной накладной. После надо пересчитать цены заново и сохранить извлеченный документ.

```
oНакл = Выборка ("Расходные накладные");  
oНакл.Номер = "1";  
Найти (oНакл, "Номер"); # извлекаем расходную  
накладную с номером 1  
переименуем ВидЦены = "Скидка10";  
oНакл.ДобавитьВидЦены (переименуем ВидЦены); # добавляем вид  
цены  
oНакл.РассчитатьЦены (); # пересчитываем цены  
oНакл.Сохранить; # сохраняем изменения в документе
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [ДобавитьВидыЦенКарты](#)
- [ОчиститьВидыЦен](#)
- [РассчитатьЦены](#)

ДобавитьВидыЦенКарты

Добавляет в выборку «Виды цен» документа все виды цен, привязанные к указанной карте.

Синтаксис

Документ . ДобавитьВидыЦенКарты (НомерКарты)

Параметры

НомерКарты (текст) – номер карты, виды цен из которой нужно добавить в выборку «Виды цен». Если цены уже были добавлены, то выдаст сообщение об ошибке.

Возвращает

Да (логическое значение) – удалось добавить виды цен из указанной карте.



Нет (логическое значение) – не удалось добавить виды цен.

Комментарии

Работает только в контексте документа, содержащего выборку «Виды цен».

Пример

Необходимо добавить виды цен по карте с заданным номером в определенную расходную накладную. После надо пересчитать цены заново и сохранить извлеченный документ.

```
oНакл = Выборка ("Расходные накладные") ;  
oНакл.Номер = "1" ;  
Найти (oНакл, "Номер") ; # извлекаем расходную  
накладную с номером 1  
перем pНомерКарты = "2" ;  
oНакл.ДобавитьВидыЦенКарты (pНомерКарты) ; # добавляем  
виды цен из карты с номером 2  
oНакл.РассчитатьЦены () ; # пересчитываем цены  
oНакл.Сохранить ; # сохраняем изменения в документе
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [ДобавитьВидыЦен](#)
- [ОчиститьВидыЦен](#)
- [РассчитатьЦены](#)

СуммыПоКарте

Позволяет подсчитать определенные суммы по карте за указанный период.

Синтаксис

```
Документ.СуммыПоКарте ([Дата1, Дата2])  
Документ.СуммыПоКарте (КолвоМесяцев)
```

Параметры

Дата1, Дата2 (дата) – период, за который необходимо подсчитать суммы. Если даты не указаны, то берутся переменные "ДатНач" и "ДатКнц" соответственно, если же и их нет, то берётся рабочий период из конфигурации системы.

КолвоМесяцев (целое) – количество месяцев. В этом случае суммы будут рассчитываться за указанное количество последних месяцев от текущей даты.

Возвращает

Объект, содержащий следующий набор полей: **Держатель** (имя владельца карты), **Карта** (адрес карты, по которой ведётся расчет), **ВидЦены** (вид цены по карте), **СуммаПокупок** (суммарный оборот по карте за указанный период, включая начальную сумму на карте), **СуммаСкидок** (общая сумма скидок за период), **СуммаПремий** (общая сумма премий за указанный период), **Кол_воВизитов** (число отгрузок по карте).

Комментарии

Работает в контексте документа, содержащего выборку «Виды цен», или в контексте карты.

Примеры

Необходимо написать выражение для вида цены, которое определяет размер скидки 5% при обороте по карте за последний год более 10000:

```
пСкидка = 0;  
оСуммы = СуммыПоКарте(12);  
Если (оСуммы.СуммаПокупок >= 10000)  
    пСкидка = 5;  
СуммаСкидки = БазЦена * пСкидка / 100;  
Вернуть БазЦена - СуммаСкидки;
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [ЦеныНаДату](#)

ОчиститьВидыЦен

Удаляет все записи из выборки «Виды цен» документа.

Синтаксис

`Документ . ОчиститьВидыЦен ()`

Параметры

Не указываются.

Возвращает

Да (логическое значение) – удалось очистить виды цен.

Нет (логическое значение) – не удалось очистить виды цен или функция была вызвана не в контексте документа.

Комментарии

Работает только в контексте документа, содержащего выборку «Виды цен».

Пример

Необходимо заменить все виды цен на вид цены с именем «Скидка10» в определенной расходной накладной. Сначала очищается таблица видов цен, потом добавляется вид цены с Именем "Скидка10". После необходимо пересчитать цены заново и сохранить извлеченный документ.

```
oНакл = Выборка ("Расходные накладные");
oНакл.Номер = "1";
Найти(oНакл, "Номер"); # извлекаем расходную
накладную с номером 1
переимяВидыЦены = "Скидка10";
Если(oНакл.ОчиститьВидыЦен())
{
    oНакл.ДобавитьВидыЦены(переимяВидыЦены); # добавляем
виды цен
    oНакл.РассчитатьЦены(); # пересчитываем цены
    oНакл.Сохранить; # сохраняем изменения в
документе
}
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [ДобавитьВидЦены](#)
- [ДобавитьВидыЦенКарты](#)
- [РассчитатьЦены](#)

РассчитатьЦены

Пересчитывает цены документа.

Синтаксис

`Документ.РассчитатьЦены()`

Параметры

Не указываются.

Возвращает

Да (логическое значение) – удалось пересчитать цены.

Нет (логическое значение) – не удалось пересчитать цены или функция была вызвана не в контексте документа.

Комментарии

Работает только в контексте документа, содержащего выборку «Виды цен».

Пример

Необходимо удалить все виды цен в определенной расходной накладной:

```
oНакл = Выборка("Расходные накладные");  
oНакл.Номер = "1";  
Найти(oНакл, "Номер"); # извлекаем расходную  
накладную с номером 1  
Если(ОчиститьВидыЦен())  
{  
    РассчитатьЦены(); # пересчитывает цены документа  
oНакл.Сохранить; # сохраняем изменения в
```

```
документе  
}
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [ДобавитьВидЦены](#)
- [ДобавитьВидыЦенКарты](#)
- [ОчиститьВидыЦен](#)

ЦеныНаДату

Возвращает запись из истории цен номенклатуры или запись наименования указанного прайса на указанную дату и время. Список существующих цен можно просмотреть на вкладке «Цены» карточки номенклатуры.

Синтаксис

Запись .ЦеныНаДату ([Прайс , КодУпаковки ,] Дата [,
Время])

Параметры

Прайс (объект) – вид цены, к которому привязываются прайсы.

КодУпаковки (целое) – один из возможных кодов упаковки для номенклатуры, цены на которую нужно получить.

Дата (дата) – дата действия цен номенклатуры.

Время (время) – время, на которое действуют указанные цены. Если параметр не указан, возвращает цены, действующие на текущее время.

Возвращает

Объект – запись цен на указанную дату и время.

Комментарии

Работает только в контексте записи номенклатуры или карточки складской картотеки.

Примеры



Узнаем цены записи номенклатуры на указанную дату:

```
оЛицо = Лицо ("10901", "Номенклатура") ;  
оН = оЛицо . Лицо _ ЦеныНаДату (20.03.06) ;  
Сообщить (оН . Цена1) ;  
Сообщить (оН . Цена2) ;  
Сообщить (оН . Цена3) ;
```

Хотим узнать цены наименования оТовар в документе оДокум с учетом установленных прайс-листов:

```
Перем оН ;  
ДляВсех (Записей ("Связи видов цен", оДокум))  
оН = оТовар . ЦеныНаДату (ТекущийКонтекст, 1, 21.03.06) ;  
# в оН цены с учетом связанных с документом прайсов .
```

См. также:

- [НайтиПрайс](#)
- [ПолучитьЦеныПоПрайсу](#)
- [СуммыПоКарте](#)