

Глава 3. Окна, меню, диалоги и прочие интерфейсные элементы

Диалог с пользователем	3
Сообщить	3
ОкДляВсех	4
СообщитьОбОшибке	5
Ошибка	6
ОшибкаСПомощью	7
ОшибкаПоКоду	8
ДаНет	10
ДаНетДляВсех	11
ОтвечатьНаВсеВопросы	12
Вопрос	13
ВопросСКартинкой	14
Спросить	14
СпроситьКаталог	16
СпроситьФайл	17
СпроситьДаты	17
Меню	18
Меню(Объект)	20
Гудок	20
СчитатьСКлавиатуры	21
ВывестиСтатус	22
Панель инструментов	23
ПоказатьПанельИнструментов	23
СкрытьПанельИнструментов	24
РазрешитьКнопкиПанели	25
ЗапретитьКнопкиПанели	25
Окна и диалоги	26
Окно	26
Диалог	27
Заголовков	27
Открыть – открытие окна	28
Выполнить	29
ВыполнитьВыбор	31
Закрыть – закрытие окна	31
НайтиОкно	32
БазовоеОкно	33
ДиалогХозяин	33
ИзмениласьЗапись	34
Поля окон и диалога	35
АктивироватьЗакладку	35
АктивнаяЗакладка	35
СпрятатьЗакладку	36
ПоказатьЗакладку	37
НайтиЗакладку	38
АктивироватьВвод	39
АктивныйВвод	40
ЗапретитьВвод	41

ЗапретитьВводРасш.....	41
РазрешитьВвод.....	43
РазрешитьВводРасш.....	44
СпрятатьВвод.....	45
ПоказатьВвод.....	45
СостояниеВвода.....	46
ПометитьНеверныйВвод.....	47
РазметитьНеверныйВвод.....	48
ОбновитьВвод.....	49
ЕстьВвод.....	50
СохранитьПоля.....	51
ИзменилосьПоле.....	51
УстановитьСписокПунктов.....	52
ПолучитьСписокПунктов.....	53
УстановитьТекст.....	54
УстановитьПодсказку.....	55
ЗапретитьФлаги.....	55
РазрешитьФлаги.....	56
Координаты.....	57
УстановитьКоординаты.....	58
Страницы в диалоге-помощнике.....	59
ПоказатьСтраницу.....	59
СброситьПоследниеСтраницы.....	60
СделатьСтраницуПоследней.....	60
ТекущаяСтраница.....	61
Команды для окон и диалогов.....	62
ВыполнитьКоманду – для окна.....	62
ВыполнитьКоманду.....	63
ПоместитьВОчередьКоманд.....	64
ПоместитьНаЭкран.....	65
ПослатьСообщение.....	65
ПослатьСообщениеВсем.....	66
ПраваНаКоманду.....	67
СменитьФоновоеОкно.....	68
Таблицы в окнах и диалогах.....	69
Запись – для окна/диалога.....	69
УстановитьКурсор.....	70
ДобавитьЗапись.....	71
ИзменитьЗапись.....	72
ОтметитьЗапись.....	73
РазметитьЗапись.....	74
СнятьОтметкуЗаписей.....	75
ЕстьОтмеченныеЗаписи.....	76
ДляВсех(ОтмеченныхЗаписей).....	77
ДляВсех(ОтмеченныхИлиВсехЗаписей).....	78
ВставитьПоле.....	79
УдалитьПоле.....	81
СпрятатьПоле.....	81



ПоказатьПоле.....	82
Установка параметров в складских отчетах.....	83
ВыбратьЭлемент.....	83
ОчиститьЭлементы.....	84
ОбновитьПараметры.....	85
УстановитьФильтр.....	85

Диалог с пользователем

Сообщить

Выводит на экран текст сообщения.

Синтаксис

Сообщить (Сообщение1 [, Сообщение2] ...)

Параметры

Сообщение1, **Сообщение2**, ... (текст) – строки с сообщениями. Текст в окне автоматически центрируется. Если сообщение слишком длинное, его можно разбить на несколько строк при помощи тега «
» (новая строка). Если же символы разбиения не указаны, текст разбивается автоматически.

Возвращает

Ничего не возвращает.

Примеры

Удобно формировать строку сообщения из нескольких строк. Например, для приходного кассового ордера при выборе сотрудника можно сообщать сумму его задолженности по 71 счёту:

```
пСумма = СКД("71", Лицо1)
Сообщить ("За " + Лицо1.Название<br>задолженность по
71 счёту: " + пСумма); # сообщает указанный текст.
```

См. также:

- [СообщитьОбОшибке](#)
- [Ошибка](#)

- [ДаНет](#)
- [Вопрос](#)
- [Спросить](#)
- [Меню](#)

ОкДляВсех

Выводит на экран текст сообщения и предлагает пользователю нажать кнопку «Ок» или «ОкДляВсех».

Синтаксис

`ОкДляВсех (Вопрос , ИдентификаторДляВсех)`

Параметры

Вопрос (текст) – сообщение, которое посылается пользователю. Текст сообщения выводится аналогично функции «Сообщить».

ИдентификаторДляВсех (текст) – идентификатор сообщения.

Возвращает

Ничего не возвращает.

Комментарий

Функцию удобно использовать в циклах. После первого вызова функции программа записывает в памяти идентификатор и значение ответа пользователя «ОкДляВсех» на это сообщение. При последующем вызове функции программа по идентификатору находит значение ответа «Ок» и это сообщение для пользователя больше не появляется.

Примеры

Пересчитаем итоговые суммы накладных, удовлетворяющих заданному условию, и сообщим об этом пользователю:

```
ДляВсех (Документов ("НаклПрих" )  
Если (Документ . Сумма > 3000)  
{  
    Документ . Сумма = Документ . Сумма / 5 ;
```



```
ОКДляВсех ("Сумма документа изменилась",  
"Сумма");  
}
```

См. также:

- [Сообщить](#)
- [ДаНет](#)
- [Вопрос](#)
- [Спросить](#)
- [Меню](#)

СообщитьОБОшибке

Выводит на экран окно с сообщением, не прерывая выполнение текущего процесса.

Синтаксис

```
СообщитьОБОшибке (Сообщение1 [ , Сообщение2 ] ...)
```

Параметры

Сообщение1, Сообщение2, ... (текст) – тексты сообщений. Сообщение выводится аналогично функции «Сообщить».

Возвращает

Ничего не возвращает.

Примеры

Чтобы проверить при закрытии документа, установлено ли значение в поле «Лицо2», можно в правиле операции написать так:

```
Если (! (Лицо2.Название) )  
СообщитьОБОшибке ("Не установлено Лицо2!"); #  
появится сообщение о том, что не установлено  
«Лицо2», но документ будет закрыт.
```

См. также:

- [Сообщить](#)

- [Ошибка](#)

Ошибка

Выводит на экран окно с системным значком ошибки в виде красного креста и прерывает выполнение текущего процесса (закрытия документа или построения отчёта).

Синтаксис

Ошибка (Сообщение1 [, Сообщение2] ...)

Параметры

Сообщение1, Сообщение2, ... (текст) – текст сообщения. Сообщение выводится аналогично функции «Сообщить».

Комментарии

Если функция была вызвана в процессе выполнения транзакции над базой данных (например, в процессе закрытия документа), то все изменения базы данных, сделанные в этой транзакции, будут отменены (в частности документ останется не закрытым).

Возвращает

Ничего не возвращает.

Примеры

Чтобы проверить, установлено ли поле «Лицо2» документа, можно в правиле операции написать так:

Если (! (Лицо2 . Название))


Ошибка ("Не установлено Лицо2!"); # этот фрагмент гарантирует, что документ не будет закрыт, пока не будет указано «Лицо2».

См. также:

- [СообщитьОбОшибке](#)
- [ОшибкаСПомощью](#)

ОшибкаСПомощью



Показывает сообщение об ошибке с кнопкой , нажав на которую пользователю открывается справка, и прерывает выполнение текущего процесса.

Синтаксис

ОшибкаСПомощью (ТекстОшибки [, СсылкаНаПомощь])

Параметры

ТекстОшибки (текст) – текст в сообщении об ошибке. Сообщение выводится аналогично функции «Сообщить».

СсылкаНаПомощь (текст) - полная ссылка на помощь (путь к файлу справки (.chm), разделу справки (htm-страница) или месту на странице файла справки, заданного с помощью #). Путь к файлу справки может быть глобальным, или может содержать только название файла справки, если файл справки находится в модулях. Правильное задание пути можно узнать в свойствах htm-страницы (блок «Адрес (URL):»).

Комментарии

Если функция была вызвана в процессе выполнения транзакции над базой данных (например, в процессе закрытия документа), то все изменения базы данных, сделанные в этой транзакции, будут отменены (в частности документ останется не закрытым).

Возвращает

Ничего не возвращает.

Примеры

Перейти к статье №7 закона о бухучете:

```
ОшибкаСПомощью( "Прочитайте справку",  
"ПОМОЩЬ%20БУХГАЛТЕРУ\FZBuchUch.chm: :/129_fz_21_11_1  
996/glava_i.htm#st7" );
```

См. также:

- [СообщитьОбОшибке](#)
- [Ошибка](#)

- [ОшибкаПоКоду](#)

ОшибкаПоКоду

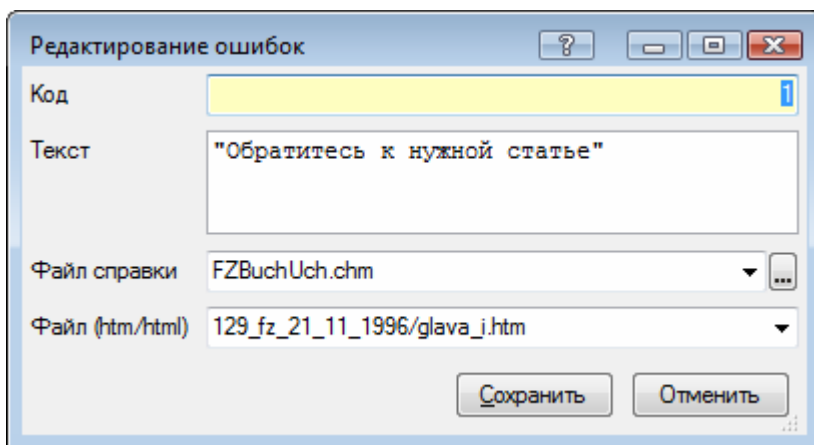
Выводит на экран окно с текстом ошибки по коду и прерывает выполнение текущего процесса.

Синтаксис


`ОшибкаПоКоду (Код [, ЗалкадкаНаСтранице])`


Параметры

Код (целое) – код ошибки. В файле ресурса создается коллекция записей с форматом «Коды ошибок»:



Этот параметр передает код ошибки из этой коллекции записей. Кроме кода в коллекции записей хранится текст сообщения об ошибке, и путь к

файлу справки, который откроется при нажатии на кнопку . Правильное задание пути можно узнать в свойствах htm-страницы (блок «Адрес (URL):»). Если файл справки не указан, то в окне сообщения об ошиб-

ке будет отсутствовать кнопка . Дополнительно можно указать раздел справки (htm-страница) или место на странице, к которому нужно перейти

и заданного с помощью #. Путь к странице необходимо указывать без символов «:/>».

ЗакладкаНаСтранице (текст) – место, к которому нужно перейти на странице справки, указанной в коллекции записей, и заданное с помощью #. Параметр необходимо указывать без символа «#». Если параметр не указан, то будет возможен переход к самой htm-странице, указанной в поле «Файл (htm/html)» (см. рисунок). Здесь же можно прописать и закладку, к которой нужно перейти, уже с использованием якоря (символ #), в этом случае данный параметр передавать нецелесообразно.

Комментарии

Функция выполняет поиск такой коллекции записей по всем ресурсам программы. Если закладка, к которой нужно перейти на странице справки, прописана и в коде ошибки, и передается в качестве второго параметра в функции, то будет выполнен переход к той закладке на странице, которая передается в функции.

Если функция была вызвана в процессе выполнения транзакции над базой данных (например, в процессе закрытия документа), то все изменения базы данных, сделанные в этой транзакции, будут отменены (в частности документ останется не закрытым).

Возвращает

Ничего не возвращает.

Примеры

Необходимо перейти к статье 7 Федерального закона «О бухгалтерском учете» («Помощь бухгалтеру»):

```
ОшибкаПоКоду(1, "st7")
```

См. также:

- [СообщитьОбОшибке](#)
- [Ошибка](#)
- [ОшибкаСПомощью](#)

ДаНет

«Задаёт» пользователю вопрос, указанный в качестве параметра, и «просит» ответить «Да» или «Нет». Если вопрос слишком длинный, его можно разбить на несколько строк при помощи комбинации символов «
».

Синтаксис

ДаНет (Вопрос [, КнопкаПоУмолчанию])

Параметры

Вопрос (текст) – вопрос, который задаётся пользователю. Текст вопроса выводится аналогично функции «Сообщить».

КнопкаПоУмолчанию (целое) – номер кнопки, выбранной по умолчанию. Единица соответствует кнопке «Да», двойка – «Нет». Любое другое значение трактуется как единица.

Возвращает

Да (логическое значение) – если пользователь выбрал «Да».

Нет (логическое значение) – если пользователь выбрал «Нет» или нажал клавишу <Esc>.

Примеры

Например, если значение проверяемой переменной отрицательное, то возникает следующий вопрос:

```
пА=-5;
```

```
Если (пА < 0)
```

```
    Если (ДаНет ("Переменная пА меньше 0. Исправить?"))
```

```
        пА = Модуль (пА) ;
```

```
Сообщить ("Переменная пА равна " + пА + ".") ;
```

в результате, в случае положительного ответа на вопрос переменная будет принимать положительное значение; иначе значение переменной останется без изменений.

См. также:

- [ДаНетДляВсех](#)
- [ОкДляВсех](#)



ДаНетДляВсех

«Задаёт» пользователю вопрос и «просит» ответить «Да», «ДаДляВсех», «Нет» или «НетДляВсех».

Синтаксис

ДаНетДляВсех (Вопрос , ИдентификаторДляВсех [, КнопкаПоУмолчанию])

Параметры

Вопрос (текст) – вопрос, который задаётся пользователю. Текст вопроса выводится аналогично функции «Сообщить».

ИдентификаторДляВсех (текст) – идентификатор вопроса. После первого вызова функции программа записывает в памяти идентификатор и значение ответа пользователя на этот вопрос. При последующем вызове функции программа по идентификатору находит значение ответа и вопрос для пользователя больше не появляется.

КнопкаПоУмолчанию (целое) – номер кнопки, выбранной по умолчанию. 1 соответствует кнопке «Да», 2 – «Да для всех», 3 – «Нет», 4 – «Нет для всех». Любое другое значение трактуется как единица.

Возвращает

Да (логическое значение) – если пользователь выбрал «Да».

Нет (логическое значение) – если пользователь выбрал «Нет» или нажал клавишу <Esc>.

Комментарий

Функцию удобно использовать в циклах. В таком случае, если нажать «Да для всех» или «Нет для всех», вопрос не повторится на экране, а действие будет либо «Да», либо «Нет» соответственно.

Примеры

В реестре приходных накладных пересчитаем сумму документов, удовлетворяющих заданному условию:

```
ДляВсех (Документов ("НаклПрих" )  
Если (Документ . Сумма > 300000)  
Если (ДаНетДляВсех ("Пересчитать сумму  
документа?" , "Сумма" , 2) )
```

`Документ.Сумма = Документ.Сумма/5;`

См. также:

- [ДаНет](#)
- [ОкДляВсех](#)

ОтвечатьНаВсеВопросы

Позволяет задавать значение ответа для серии вопросов ДаНет.

Синтаксис

`ОтвечатьНаВсеВопросы (Ответ)`

Параметры

Ответ (текст) – вопрос, который задаётся пользователю. Текст вопроса выводится аналогично функции «Сообщить».

Возвращает

Ничего не возвращает.

Комментарии

После вызова функции с параметром перестают появляться окна с вопросами, т.е. при вызове функций ДаНет сразу будет возвращаться значение, заданное в параметре. При вызове функции без параметра снова будут появляться окна с вопросами.

Примеры

```
ОтвечатьНаВсеВопросы( Да );  
перем п1 = ДаНет( "Уверен?" );  
перем п2 = ДаНет( "Точно?" );
```

В переменные п1 и п2 сразу будет записан ответ «Да», а окна с вопросами не будут появляться на экране пользователя.

```
ОтвечатьНаВсеВопросы( Да )  
перем п1 = ДаНет( "Уверен?" );  
ОтвечатьНаВсеВопросы( );  
перем п2 = ДаНет( "Точно?" );
```



В переменную `p1` будет записано значение «Да», после этого пользователю будет задан вопрос «Точно?» и в переменную `p2` запишется ответ пользователя.

См. также:

- [ДаНет](#)

Вопрос

Выводит на экран текст вопроса и ответы к нему.

Синтаксис

`Вопрос (Вопрос , Ответ1 , Ответ2...)`

Параметры

`Вопрос` (текст) - текст вопроса.

`Ответ1, Ответ2 ...` (текст) - возможные варианты ответов.

Возвращает

Текст - выбранный вариант ответа или пустую строку, если нажали Esc.

Примеры

Например, если значение проверяемой переменной отрицательное, то возникает следующий вопрос:

```
pA=-5 ;
```

```
Если (pA < 0)
```

```
    Если (Вопрос ("Переменная pA меньше 0 .  
Исправить?" , "Да" , "Нет" ) == "Да" )
```

```
        pA = Модуль (pA) ;
```

```
Сообщить ("Переменная pA равна " + pA + " .") ;
```

в результате, в случае положительного ответа на вопрос переменная будет принимать положительное значение; иначе значение переменной останется без изменений.

См. также:

- [ВопросСКартинкой](#)
- [Спросить](#)

- [Меню](#)

ВопросСКартинкой

Выводит на экран текст вопроса с картинкой и ответы к нему.

Синтаксис

`ВопросСКартинкой(Вопрос, Путь, Ответ1, Ответ2...)`

Параметры

Вопрос (текст) - текст вопроса.

Путь (текст) – полный путь к файлу с картинкой. Файл может иметь любое расширение, поддерживаемое программой.

Ответ1, Ответ2 ... (текст) - возможные варианты ответов.

Возвращает

Текст - выбранный вариант ответа или пустую строку, если нажали «Esc».

Примеры

В электронной отчетности при считывании отчета из файла и выявлении ошибок в нем используется следующий вариант вопроса с картинкой с указанными вариантами выбора дальнейших действий:

```
ВопросСКартинкой("Файл проверен на соответствие  
формату электронной отчетности."+"<br>"+<br>" +  
"Обнаружены ошибки!"+<br>"+<br>" Данные могут быть  
прочитаны некорректно!"+<br>"+<br>" "!Электронная  
отчетность\\Изображения  
\\warning.gif", "Исправить и прочитать", "Прочитать  
с ошибками",  
"Отменить чтение");
```

См. также:

- [Вопрос](#)

Спросить

Запрашивает у пользователя набор значений.

Синтаксис

Спросить ([Метка1 ,] Перем1 , [[Метка2 ,] Перем2 ,] ...)

Параметры

Перем1,Перем2 ... (переменная) – запрашиваемые переменные. Функция создаёт диалог, в котором для каждой переменной, указанной при вызове функции, соответствует своё поле ввода. Тип поля ввода зависит от типа переменной. На момент вызова функции указанные переменные должны существовать, их значения записываются в соответствующие поля диалога.

Метка1,Метка2,...(текст) – метки, размещаемые перед полями ввода. Служат для пояснения, что должен ввести пользователь. Если метка не указана, то в качестве метки используется имя переменной.

Возвращает

Да (логическое значение) – пользователем была нажата клавиша <Enter>, введённые пользователем значения записываются в соответствующие переменные.

Нет (логическое значение) – была нажата клавиша <Esc>, значения переменных не изменяются.

Примеры

Запросить номер счёта, и если он был введен, посчитать по нему сальдо:

```
пСчет = "50";  
Если (Спросить (пСчет) )  
Сообщить (СКД (пСчет) ) ;
```

Запросить количество и цену товара:

```
перем пКол_во = 1;  
перем пЦена = 0;  
Если (Спросить ("Введите количество", пКол_во,  
"и цену товара", пЦена) )  
Сообщить ("Итого: "+пКол_во*пЦена) ;
```

См. также:

- [СпроситьКаталог](#)



- [СпроситьФайл](#)
- [СпроситьДаты](#)

СпроситьКаталог

Открывает окно для выбора нужного каталога и при нажатии пользователем кнопки «ОК» возвращает путь к выбранному каталогу.

Синтаксис

СпроситьКаталог (Путь)

Параметры

Путь (текст) – путь, начиная с которого можно выбрать необходимый каталог.

Комментарии

Если запрашиваемого каталога не существует, то программа предложит его создать.

Возвращает

Текст, путь к выбранному каталогу.

Примеры

```
пПуть = СпроситьКаталог("C:\\СБиС++"); # если  
выбрать каталог "db", то в пПуть будет "  
C:\\СБиС++\\db\"; иначе возвратит пустую строку.
```

См. также:

- [Спросить](#)
- [СпроситьФайл](#)
- [СпроситьДаты](#)

СпроситьФайл

Открывает окно для выбора нужного файла и при нажатии пользователем кнопки «Открыть» возвращает путь к выбранному файлу.

Синтаксис

СпроситьФайл (Путь [, ТипФайлов , Маска])

Параметры

Путь (текст) – путь, начиная с которого можно выбрать необходимый файл.

ТипФайлов (текст) – описание отбираемых файлов. По умолчанию отображает все файлы каталога.

Маска (текст) – маска отображаемых файлов, которые перечисляются через «;». По умолчанию отображает все файлы каталога.

Возвращает

Текст, путь к выбранному файлу.

Примеры

```
пПуть1 = СпроситьФайл("C:\\Sbis\\"); # если выбрать
файл «sbis.exe», то в пПуть1 будет
«C:\Sbis\sbis.exe».
```

Необходимо отображать в открывшемся каталоге только прайс-листы:

```
пПуть2 = СпроситьФайл("P:\\ЕЖЕНЕДЕЛЬНЫЕ\\Квазар-
микро\\", "Прайс-листы (*.xls, *.zip)",
"*.xls;*.zip");
```

См. также:

- [Спросить](#)
- [СпроситьКаталог](#)
- [СпроситьДаты](#)

СпроситьДаты

Запрашивает интервал дат и устанавливает переменные «ДатНач» и «ДатКнц».

Синтаксис

`СпроситьДаты ([Заголовок [, ДатНач , ДатКнц]])`

Параметры

Заголовок (текст) – заголовок окна, в котором будут спрашиваться даты. Если параметр не указан, будет заголовок «Укажите период».

ДатНач, ДатКнц (дата) – даты запрашиваемого периода. Если явно не указаны, то берутся даты из рабочего периода.

Возвращает

Да (логическое значение) – пользователь нажал <Enter>, введённые пользователем значения устанавливаются в переменные «ДатНач» и «ДатКнц».

Нет (логическое значение) – пользователь нажал <Esc>, переменные «ДатНач» и «ДатКнц» не изменяются.

Примеры

Запросить период и посчитать по нему сальдо.

```
СпроситьДаты ("Период для расчёта сальдо");
пСумма = ОД("60-1", Лицо1);
Сообщить ("За период с " + ДатНач + " по " + ДатКнц +
"нами перечислено денежных средств "+ Лицо1.Название
+ " на сумму " +пСумма);
```

См. также:

- [Спросить](#)
- [СпроситьКаталог](#)
- [СпроситьФайл](#)

Меню

Выводит меню для выбора пользователем одного из нескольких пунктов.

Синтаксис

`Меню (Пункты)`
`Меню (ИмяМассива)`



Параметры

Пункты (текст) – список пунктов меню в формате «Пункт1\nПункт2...». То есть пункты меню разделяются комбинацией символов «\n». Если количество пунктов превышает количество строк на экране, окно будет с прокруткой.

ИмяМассива (массив) – массив, содержащий имена пунктов меню.

Возвращает

Число – номер пункта меню или выбранного элемента массива, если пользователь выбрал нужный пункт и нажал на нем <Enter>, нумерация пунктов и элементов массива идёт с единицы.

0 – если пользователь нажал <Esc>.

Примеры

Совместно с функцией «Меню» удобно использовать конструкцию «ВыборПо».

```
ВыборПо (Меню ("Организации\nЧастные  
лица\nСотрудники" )  
{  
  выбор 1:  
    Сообщить ("Список организаций" );  
  выбор 2:  
    Сообщить ("Список частных лиц" );  
  выбор 3:  
    Сообщить ("Список сотрудников" );  
  иначе  
    Сообщить ("Ничего не выбрано" );  
}
```

Пример с массивом:

```
мМ = Массив ("Организации", "Сотрудники", "Частные  
лица" );  
пМеню = Меню (мМ) ; # в пМеню – номер элемента  
массива, либо «0».
```

Строку с пунктами меню можно набирать в зависимости от некоторых данных, например, набрать в меню наименования документа:

```
перем пПункты [ ] ;  
ДляВсех (Наименований)  
  пПункты [Наименование] += Наим.Наименование ;  
  пНомер = Меню (пПункты) ;
```



См. также:

- [Сообщить](#)
- [ДаНет](#)
- [Вопрос](#)
- [Спросить](#)
- [Меню\(Объект\)](#)

Меню(Объект)

Выводит меню для выбора пользователем одного из нескольких пунктов.

Синтаксис

Меню (ИмяОбъекта)

Параметры

ИмяОбъекта (объект) – объект, содержащий имена пунктов меню.

Возвращает

Строка (текст) – имя выбранного элемента объекта, если пользователь выбрал нужный пункт и нажал на нем <Enter>. Если пользователь нажал <Esc>, то возвращает пустую строку.

Примеры

```
перем оМ[ ] ;  
оМ.О = "Организации" ;  
оМ.С = "Сотрудники" ;  
оМ.Ч = "Частные лица" ;  
пМеню = Меню(оМ) ;  
# в пМеню – имя элемента, либо пустая строка.
```

См. также:

- [Меню](#)

Гудок

Воспроизводит звуковой сигнал.

Синтаксис

`Гудок ([Кол_во])`

Параметры

Кол_во (число) – сколько раз воспроизводить сигнал. Если параметр не указан, то функция будет гудеть один раз.

Возвращает

Ничего не возвращает.

Примеры

Может использоваться для привлечения внимания пользователя, например, при закрытии документа проверяется некоторое условие, при срабатывании которого оператору выдаётся звуковой сигнал и сообщение:

```
Если (Сумма > 10000)
{
    Гудок ();
    Ошибка ("Сумма не может быть больше 10000!");
}
```

Можно и более замысловато (гуднуть от суммы):

```
Если (Сумма > 10000)
{
    Гудок (Сумма / 10000);
    Ошибка ("Сумма не может быть больше 10000!");
}
```

СчитатьСКлавиатуры

Ожидает ввод символов с клавиатуры.

Синтаксис

`СчитатьСКлавиатуры ([Время, Кол-во])`

Параметры

Время (целое) – время ожидания ввода. По умолчанию время ожидания 3 секунды (3000 мсек).

Кол-во (целое) – максимальное кол-во вводимых символов. По умолчанию максимальное кол-во для ввода 254 символа.

Комментарии

Для подтверждения ввода символов необходимо нажать «Enter», для отмены – «Esc». Функцию удобно применять, например, при программировании клавиш клавиатуры.

Возвращает

Строка (текст) – строка с введенными символами.

Примеры

Введем с клавиатуры строку, содержащую не более 10 символов:

```
Сообщить (СчитатьСКлавиатуры (10000 , 10) ) ;
```

ВывестиСтатус

Управляет строкой состояния (самой нижней строкой на экране, в которой отображается ход выполнения процесса).

Синтаксис

```
ВывестиСтатус ( [Сообщение [ , Шаг [ , ПоказатьВОкне] ] ] )  
ВывестиСтатус ( [Сообщение , ] ПоказатьВОкне )  
ВывестиСтатус ( Шаг [ , ПоказатьВОкне ] )
```

Параметры

Сообщение (текст) – текст сообщения, выводимого в строке состояния. Если этот параметр опущен, то функция просто обновляет индикатор хода выполнения процесса и время выполнения процесса.

Шаг (число) – если при вызове функции был указан параметр «Сообщение», то параметр «Шаг» определяет общее количество шагов, за которое будет выполнен процесс. Если же указан только параметр «Шаг», то его значение трактуется, как номер текущего шага.

ПоказатьВОкне (логическое) – значение «Да» соответствует тому, что индикатор состояния будет выведен сразу же в новом окне. По умолчанию



задано значение «Нет», т.е. первые 3 секунды состояние отображается в нижней строке экрана.

Возвращает

Ничего не возвращает.

Примеры

Простейший вызов функции, просто информирующий о том, что процесс идёт:

```
ВывестиСтатус ("Процесс пошел...");  
ДляВсех (Наименований) { ... }
```

При таком варианте использования функции пользователь не понимает, работает ли вообще компьютер или он уже «завис». Сделаем так, чтобы в отдельном окне выводилось время выполнения операции:

```
ВывестиСтатус ("Процесс пошел...");  
ДляВсех (Наименований)  
{  
    ВывестиСтатус (Да); # обновляем время.  
    ...  
}
```

В этом случае пользователь видит, что выполняется какой-то определенный процесс.

Панель инструментов

ПоказатьПанельИнструментов

Отображает указанную панель инструментов окна.

Синтаксис

```
ПоказатьПанельИнструментов (НазвПанели)
```

Параметры

НазвПанели (текст) – название панели инструментов.

Возвращает

Да (логическое значение) – указанная панель инструментов отображена.

Нет (логическое значение) – указанная панель инструментов не отображена.

Примеры

`ПоказатьПанельИнструментов("Список задач"); #`
`отображает панель инструментов «Список задач».`

См. также:

- [СкрытьПанельИнструментов](#)
- [РазрешитьКнопкиПанели](#)
- [ЗапретитьКнопкиПанели](#)

СкрытьПанельИнструментов

Скрывает панель инструментов окна.

Синтаксис

`СкрытьПанельИнструментов(НазвПанели)`

Параметры

`НазвПанели` (текст) – название панели инструментов.

Возвращает

Да (логическое значение) – указанная панель инструментов скрыта.

Нет (логическое значение) – указанная панель инструментов не скрыта.

Примеры

`СкрытьПанельИнструментов("Список задач"); #` скрывает
`панель инструментов «Список задач».`

См. также:

- [ПоказатьПанельИнструментов](#)
- [РазрешитьКнопкиПанели](#)
- [ЗапретитьКнопкиПанели](#)



РазрешитьКнопкиПанели

Делает активными указанные кнопки заданной панели инструментов в диалоге.

Синтаксис

`РазрешитьКнопкиПанели (ИмяПанели, Кнопка1, Кнопка2, ...)`

Параметры

`ИмяПанели` (текст) – название панели.

`Кнопка1, Кнопка2, ...` (текст|число) – название команды или номер кнопки, которая должна быть активной.

Возвращает

Ничего не возвращает.

См. также:

- [ЗапретитьКнопкиПанели](#)
- [ПоказатьПанельИнструментов](#)
- [СкрытьПанельИнструментов](#)

ЗапретитьКнопкиПанели

Делает неактивными указанные кнопки заданной панели инструментов в диалоге.

Синтаксис

`РазрешитьКнопкиПанели (ИмяПанели, Кнопка1, Кнопка2, ...)`

Параметры

`ИмяПанели` (текст) – название панели.

`Кнопка1, Кнопка2, ...` (текст|число) – название команды или номер кнопки, которую нужно сделать неактивной.

Возвращает

Ничего не возвращает.

См. также:

- [РазрешитьКнопкиПанели](#)
- [ПоказатьПанельИнструментов](#)
- [СкрытьПанельИнструментов](#)

Окна и диалоги

Окно

Создает объект на основе имени окна.

Синтаксис

Окно (Имя)

Параметры

Имя (текст) - имя загружаемого окна. Описание всех окон находится в файлах ресурсов (файлы с расширением «*.rs»).

Возвращает

Объект, содержащий запись окна.

Примеры

```
oОкно = Окно ("Сотрудники"); # создает окно «Сотрудники» .
```

См. также:

- [Заголовок](#)
- [Открыть – открытие окна](#)
- [Закрыть – закрытие окна](#)
- [Выполнить](#)
- [НайтиОкно](#)

Диалог

Создаёт объект, содержащий окно диалога с записью, для дальнейшего вывода этого окна функцией «Выполнить».

Синтаксис

Диалог (ИмяДиалога [, Запись])

Параметры

ИмяДиалога (текст) – имя загружаемого диалогового окна. Описания всех диалоговых окон находятся в файлах ресурсов (файлы с расширением «*.rs»). В загружаемом диалоговом окне не должно быть полей с типом «Абстрактное поле ввода».

Запись (объект) – запись, добавляемая в диалог для дальнейшего редактирования. Если параметр не указывается, функция создает диалог, не содержащий никаких данных.

Возвращает

Объект, содержащий запись загруженного диалога. Фактически каждому полю ввода в окне диалога соответствует одноимённое поле этой записи.

Примеры

Добавим в справочнике новую статью затрат:

```
oСтатья = Выборка ("Статьи затрат" ) ;
oСтатья.Тема = "РЕМЗД" ;
oСтатья.Название = "Ремонт здания" ;
oДиалог = Диалог ("Статьи затрат", oСтатья) ;
Если (oДиалог.Выполнить ( ) )
    oСтатья.Добавить ( ) ;
```

См. также:

- [Выполнить](#)
- [ДиалогХозяин](#)
- [ИзмениласьЗапись](#)

Заголовок

Устанавливает/возвращает заголовок окна.



Синтаксис

`Окно.Заголовок ([ИмяОкна])`

Параметры

ИмяОкна (текст) – новый заголовок окна. Если параметр не указан, возвращает существующий заголовок окна.

Возвращает

Текст – первоначальный заголовок окна.

Примеры

```
одиалог = Диалог ("Конфигурация системы");  
пИмя = одиалог.Заголовок; # в пИмя будет  
"Конфигурация системы".
```

См. также:

- [Окно](#)
- [Диалог](#)

Открыть – открытие окна

Открывает окно, загруженное функцией «Окно», «Диалог».

Синтаксис

`Окно.Открыть ()`

Параметры

Не указываются.

Возвращает

Да (логическое значение) – если окно открыто.

Нет (логическое значение) – если окно открыть не удалось.

Пример

```
перем оКасса = Окно ("Кассовая книга");  
оКасса.Открыть ( ); # откроется окно "Кассовая книга".
```

См. также:

- [Окно](#)
- [Закреть – закрытие окна](#)

Выполнить

Выполняет диалоговое окно, загруженное функцией «Диалог», «Окно».

Синтаксис

`Окно . Выполнить ()`

Параметры

Не указываются.

Возвращает

Да (логическое значение) – пользователь нажал в диалоге кнопку «Сохранить» или клавишу <Enter>.

Нет (логическое значение) – пользователь нажал в диалоге кнопку «Отменить» или клавишу <Esc>.

В любом случае, введённые пользователем значения помещаются в соответствующие поля записи объекта «Диалог».

Комментарии

В общем случае схема выполнения диалога будет следующая. Сначала создаём диалог:

```
перем оДиалог = Диалог("Паспорт"); # присваиваем  
значения нужным полям  
оДиалог.Номер = 7608;  
оДиалог.Серия = 400403;  
...
```

Даём пользователю поработать с диалогом:

```
Если (оДиалог . Выполнить ( ) )  
{  
    пНомер = оДиалог.Номер; # извлекаем значения  
    пСерия = оДиалог.Серия;  
}
```

Примеры

Чтобы запросить параметры так же, как это делается при построении внешних отчётов в задаче «Баланс и анализ», нужно написать так:

```
оДиалог = Диалог ("Интервал дат");
оДиалог.ДатНач = ДатНач;
оДиалог.ДатКнц = ДатКнц;
Если (оДиалог.Выполнить ())
{
    ДатНач = оДиалог.ДатНач;
    ДатКнц = оДиалог.ДатКнц;
}
```

Здесь использовалось готовое диалоговое окно, но можно создать новое. Для этого в окне с файлом ресурсов нужно нажать <Ins> и выбрать пункт «Окно диалога».

В следующем примере предполагается, что был создан диалог с именем «Запрос имени», в который было вставлено поле «ИмяПользователя».

```
оДиалог = Диалог ("Запрос имени");
пПродолжать = 1;
Пока (пПродолжать )
    Если (оДиалог.Выполнить () и
        оДиалог.ИмяПользователя)
    {
        пПользователь = оДиалог.ИмяПользователя;
        пПродолжать = 0;
    }
    иначе
        Сообщить ("Необходимо ввести имя!");
```

Обратите внимание, что здесь организован цикл, пока пользователь не введёт своё имя. То есть можно один раз создать диалог, а потом несколько раз запускать его на выполнение.

См. также:

- [Диалог](#)
- [Окно](#)
- [ВыполнитьВыбор](#)

ВыполнитьВыбор

Вызывает окно для выбора записи.

Синтаксис

Окно . ВыполнитьВыбор ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – если сделан выбор, нажата клавиша <Enter>.

Нет (логическое значение) – если выбор не сделан, нажата клавиша <Esc>.

Примеры

Узнаем дату начала срока действия выбранной цены:

```
oОкно = Окно ("Виды цен" );  
Если (oОкно . ВыполнитьВыбор)  
{  
    oЗапись = oОкно . Запись ( ) ;  
    Сообщить (oЗапись . ДатНач) ;  
}
```

См. также:

- [Окно](#)
- [Выполнить](#)

Закреть – закрытие окна

Закрывает открытое окно аналогично выполнению команды закрытия окна в программе, вызываемой при нажатии комбинации клавиш <Ctrl+F4>.

Синтаксис

Окно . Закреть ()

Параметры

Не указываются.



Возвращает

Да (логическое значение) – если окно закрыто.

Нет (логическое значение) – если окно закрыть не удалось.

Пример

Закроем в программе справочник «Сотрудники», если он открыт:

```
оОкно = НайтиОкно ("Сотрудники") ;
Если (оОкно)
    оОкно.Закреть ()
Иначе
    Сообщить ("Справочник не открыт")
```

См. также:

- [Окно](#)
- [Открыть – открытие окна](#)
- [Выполнить](#)

НайтиОкно

Возвращает объект на основе имени открытого окна.

Синтаксис

```
НайтиОкно (ИмяОкна)
```

Параметры

ИмяОкна (текст) - имя открытого окна. Описание всех окон находится в файлах ресурсов (файлы с расширением «*.rs»).

Возвращает

Объект – запись открытого окна или пустая запись, если окно не было открытым.

Примеры

```
перем пИмяОкна = "Формы отчетности формы" ;
перем оОкно = НайтиОкно (пИмяОкна) ;
```

```
Если ( оОкно == Нет )  
    Ошибка ( "Не удалось найти окно отчетов." ); # в  
случае, если окно открыто, возвратит объект, с  
которым можно выполнять дальнейшие действия; иначе,  
возвратит пустую запись.
```

См. также:

- [Окно](#)

БазовоеОкно

Возвращает окно, в котором вызывается окно диалога.

Синтаксис

```
Окно . БазовоеОкно ( )
```

Параметры

Не указываются.

Возвращает

Объект – окно, в котором вызывается окно диалога.

Примеры

Создадим внешний обработчик для окна выбора наименования прихода в приходной накладной и узнаем заголовок базового окна:

```
функция ИмяДиалога (пИмя)  
{  
    оБазОкно = БазовоеОкно ( ) ;  
    Сообщить (оБазОкно . Заголовок) ;  
    вернуть СледФункция ;  
}
```

См. также:

- [Окно](#)

ДиалогХозяин

Возвращает диалог предыдущего уровня.



Синтаксис

Диалог.ДиалогХозяин ()

Параметры

Не указываются.

Возвращает

Объект - диалог предыдущего уровня.

Примеры

```
оДиалог = Диалог("Наименование прихода");  
оДиалогХозяин = оДиалог.ДиалогХозяин(); # при  
открытии диалога выбора наименования прихода в  
приходной накладной в оДиалогХозяин будет само  
диалоговое окно этой накладной.
```

См. также:

- [Диалог](#)

ИзмениласьЗапись

Позволяет при изменении основной записи обновить связанные записи диалога.

Синтаксис

Диалог.ИзмениласьЗапись ()

Параметры

Не указываются.

Возвращает

Ничего не возвращает.

См. также:

- [Диалог](#)

Поля окон и диалога

АктивироватьЗакладку

Активирует закладку в указанном объекте в диалоге.

Синтаксис

Окно. АктивироватьЗакладку ([ИмяОбъекта ,]
ИмяЗакладки)

Параметры

ИмяОбъекта (текст) – имя объекта-хозяина, имеющего список закладок. Если объект не указан, то активирует закладку в диалоге, если таковая имеется.

ИмяЗакладки (текст) – имя закладки в указанном диалоге, которая должна быть активна.

Возвращает

Ничего не возвращает. При указании несуществующей закладки, функция возвращает сообщение об ошибке.

Примеры

Необходимо при открытии диалога попасть сразу на закладку «Модули»:

```
оДиалог = Диалог ("О программе") ;  
оДиалог. АктивироватьЗакладку ("Модули") ;  
оДиалог. Выполнить () ;
```

См. также:

- [СпрятатьЗакладку](#)
- [ПоказатьЗакладку](#)
- [НайтиЗакладку](#)
- [Диалог](#)
- [Выполнить](#)

АктивнаяЗакладка

Возвращает имя активной вкладки указанного объекта, имеющего список вкладок, в диалоге.

Синтаксис

Окно . АктивнаяЗакладка ([ИмяОбъекта])

Параметры

ИмяОбъекта (текст) – имя объекта-хозяина, имеющего список вкладок. Если объект не указан, то функция возвращает имя активной вкладки у первого обработавшего объекта в диалоге.

Возвращает

Текст – имя активной вкладки объекта.

Примеры

См. также:

- [АктивироватьЗакладку](#)
- [СпрятатьЗакладку](#)
- [ПоказатьЗакладку](#)
- [НайтиЗакладку](#)
- [Диалог](#)
- [Выполнить](#)

СпрятатьЗакладку

Скрывает закладку в указанном объекте в диалоге.

Синтаксис

Окно . СпрятатьЗакладку ([ИмяОбъекта] , [ИмяЗакладки])



Параметры

ИмяОбъекта (текст) – имя объекта-хозяина, имеющего список закладок. Если объект не указан, то скрывает закладку в диалоге, если таковая имеется.

ИмяЗакладки (текст) – имя закладки, которую необходимо спрятать.

Возвращает

Да (логическое значение) – если закладка спрятана.

Нет (логическое значение) – если закладка не спрятана.

Примеры

Необходимо при открытии диалога скрыть закладку «Модули»:

```
оДиалог = Диалог ("О программе") ;  
оДиалог . СпрятатьЗакладку ("Модули") ;  
оДиалог . Выполнить ( ) ;
```

См. также:

- [ПоказатьЗакладку](#)
- [НайтиЗакладку](#)

ПоказатьЗакладку

В диалоге показывает спрятанную закладку. Функция, обратная функции «СпрятатьЗакладку».

Синтаксис

```
Окно . ПоказатьЗакладку ( [ИмяОбъекта , ] ИмяЗакладки )
```

Параметры

ИмяОбъекта (текст) – имя объекта-хозяина, имеющего список закладок. Если объект не указан, то отображает спрятанную закладку в диалоге, если таковая имеется.

ИмяЗакладки (текст) – имя спрятанной закладки, которую необходимо показать в диалоге.

Возвращает

Да (логическое значение) – если закладка показана.

Нет (логическое значение) – если закладка не показана.

Примеры

Необходимо при открытии диалога показать скрытую закладку «Модули»:

```
оДиалог = Диалог ("О программе") ;
оДиалог . СпрятатьЗакладку ("Модули") ;
оДиалог . Выполнить () ;
оДиалог . ПоказатьЗакладку ("Модули") ;
оДиалог . Выполнить () ;
```

См. также:

- [АктивироватьЗакладку](#)
- [СпрятатьЗакладку](#)
- [НайтиЗакладку](#)

НайтиЗакладку

Ищет в диалоге или у объекта в диалоге закладку с указанным названием.

Синтаксис

```
Окно . НайтиЗакладку (ИмяЗакладки [ , ИмяОбъекта ] )
```

Параметры

ИмяЗакладки (текст) – имя закладки, которую необходимо найти.

ИмяОбъекта (текст) – имя объекта в диалоге, содержащего список закладок. Если параметр не указан, функция ищет указанную закладку в диалоге.

Возвращает

Да (логическое значение) – если указанная закладка найдена.

Нет (логическое значение) – если указанная закладка не найдена.



Примеры

Необходимо при выводе диалога сделать закладку «Модули» активной, если она существует:

```
оДиалог = Диалог ("О программе") ;  
Если (оДиалог . НайтиЗакладку ("Модули" )  
    оДиалог . АктивироватьЗакладку ("Модули" ) ;  
оДиалог . Выполнить ( ) ;
```

См. также:

- [СпрятатьЗакладку](#)
- [ПоказатьЗакладку](#)
- [Диалог](#)
- [Выполнить](#)

АктивироватьВвод

Устанавливает активное поле ввода, т.е. в диалоге курсор позиционируется в это поле ввода, а само поле выделяется желтым цветом.

Синтаксис

```
Окно . АктивироватьВвод ( [ИмяПоля] )
```

Параметры

ИмяПоля (текст) – имя поля в диалоге, которое должно быть активным.

Возвращает

Да (логическое значение) – если активное поле установлено.

Нет (логическое значение) – если активное поле не установлено.

Примеры

При открытии диалогового окна активируем поле ввода «НазвВалюты» для выбора валюты:

```
оДиалог = Диалог ("Конфигурация системы") ;  
оДиалог . АктивироватьВвод ("НазвВалюты" ) ;  
оДиалог . Открыть ;
```



См. также:

- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [СпрятатьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)
- [ЕстьВвод](#)

АктивныйВвод

Возвращает имя активного поля ввода в диалоге.

Синтаксис

`Окно.АктивныйВвод ()`

Параметры

Не указываются.

Возвращает

Текст, имя активного поля ввода в диалоге.

Примеры

Необходимо при открытии диалога узнать имя активного поля:

```
оДиалог = Диалог("Конфигурация системы");  
Сообщить(оДиалог.АктивныйВвод ( ) );
```

См. также:

- [АктивироватьВвод](#)
- [ЕстьВвод](#)
- [Диалог](#)
- [Выполнить](#)



ЗапретитьВвод

Запрещает ввод данных в указанное поле или во все поля диалога, т.е. поле ввода становится серым.

Синтаксис

```
Окно.ЗапретитьВвод ( [ИмяПоля1] [ , ИмяПоля2 ] . . . )
```

Параметры

ИмяПоля1, ИмяПоля2, ... (текст) – имена полей ввода диалога, в которые нужно запретить ввод данных. Если параметр не указан, то функция запрещает ввод данных во все поля диалога.

Возвращает

Да (логическое значение) – если ввод в поле запрещен.

Нет (логическое значение) – если ввод в поле не запрещен (если указано несуществующее в объекте поле).

Примеры

Необходимо запретить ввод данных в поле «Серия» диалога «Паспорт»:

```
оДиалог = Диалог ("Паспорт") ;  
оДиалог.ЗапретитьВвод ("Серия") ;
```

См. также:

- [ЗапретитьФлаги](#)
- [АктивироватьВвод](#)
- [РазрешитьВвод](#)
- [СпрятатьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)

ЗапретитьВводРасш

Расширенная версия функции **ЗапретитьВвод**, позволяет указывать приоритет. Запрещает ввод данных в указанное поле или во все поля диалога. Функция обратная функции "РазрешитьВводРасш".

Синтаксис

```
Окно.ЗапретитьВводРасш (Приоритет [ , ИмяПоля1 ] [ ,  
ИмяПоля2 ] . . . )
```

Параметры

Приоритет (текст) - уровень приоритета функции. Возможные значения "приоритетный" и "обычный".

ИмяПоля1, ИмяПоля2, ... (текст) – имена полей ввода диалога, в которые нужно запретить ввод данных. Если параметр не указан, то функция запрещает ввод данных во все поля диалога.

Возвращает

Да (логическое значение) – если ввод в поле запрещен.

Нет (логическое значение) – если ввод в поле не запрещен (если указано несуществующее в объекте поле).

Примеры

Необходимо запретить изменять проводки в закрытом документе:

```
функция НаДиалогОткрыть ( )  
{  
    перем Результат = Следфункция ( ) ;  
    ЗапретитьВводРасш ( "приоритетный" ,  
"ДокументПроводки" ) ;  
    Вернуть Результат ;  
}
```

См. также:

- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [РазрешитьВводРасш](#)
- [СпрятатьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)

РазрешитьВвод

Разрешает ввод данных в указанное поле или во все поля диалога. Функция обратная функции «ЗапретитьВвод».

Синтаксис

```
Окно.РазрешитьВвод ( [ИмяПоля1] [ , ИмяПоля2 ] . . . )
```

Параметры

ИмяПоля1, ИмяПоля2, ... (текст) – имена полей ввода диалога, в которых разрешен ввод данных. Если параметр не указан, то функция разрешает ввод данных во всех полях диалога.

Возвращает

Да (логическое значение) – если ввод в поле разрешен.

Нет (логическое значение) – если ввод в поле не разрешен.

Примеры

Необходимо разрешить ввод данных в поле «Серия», в остальные поля ввод информации запрещен:

```
оДиалог = Диалог ("Паспорт" );  
оДиалог.ЗапретитьВвод ( ) ;  
оДиалог.РазрешитьВвод ("Серия" ) ;  
оДиалог.Выполнить ( ) ;
```

См. также:

- [ЗапретитьФлаги](#)
- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [СпрятатьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)

РазрешитьВводРасш

Расширенная версия функции **РазрешитьВвод**, позволяет указывать приоритет. Разрешает ввод данных в указанное поле или во все поля диалога. Функция обратная функции "ЗапретитьВводРасш".

Синтаксис

```
Окно.РазрешитьВводРасш(Приоритет [ , ИмяПоля1 ] [ ,  
ИмяПоля2 ] . . . )
```

Параметры

Приоритет (текст) - уровень приоритета функции. Возможные значения "приоритетный" и "обычный".

ИмяПоля1, ИмяПоля2, ... (текст) – имена полей ввода диалога, в которых разрешен ввод данных. Если параметр не указан, то функция разрешает ввод данных во всех полях диалога.

Возвращает

Да (логическое значение) – если ввод в поле разрешен.

Нет (логическое значение) – если ввод в поле не разрешен.

Примеры

Необходимо разрешить изменять список внешних документов закрытого документа:

```
функция НаДиалогОткрыть (  
{  
    перем pРезультат = СледФункция ();  
    РазрешитьВводРасш ("приоритетный", "Внешние  
документы_");  
    Вернуть pРезультат;  
}
```

См. также:

- [РазрешитьВвод](#)
- [ЗапретитьВвод](#)
- [ЗапретитьВводРасш](#)



- [СпрятатьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)

СпрятатьВвод

Скрывает указанное поле ввода в диалоге.

Синтаксис

`Окно . СпрятатьВвод (ИмяПоля)`

Параметры

ИмяПоля (текст) – имя поля ввода в диалоге, которое необходимо скрыть.

Возвращает

Да (логическое значение) – если указанное поле ввода в диалоге скрыто.

Нет (логическое значение) – если если указанное поле ввода в диалоге не скрыто.

Примеры

Необходимо спрятать в диалоге поле «ДатНач»:

```
оДиалог = Диалог("Интервал дат")
оДиалог . СпрятатьВвод ("ДатНач") ;
оДиалог . Выполнить ( ) ;
```

См. также:

- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [ПоказатьВвод](#)
- [ОбновитьВвод](#)

ПоказатьВвод

Функция обратная функции «СпрятатьВвод», отображает в диалоге скрытое поле ввода.

Синтаксис

`Окно . ПоказатьВвод (ИмяПоля)`

Параметры

ИмяПоля (текст) – имя поля ввода в диалоге, которое необходимо показать.

Возвращает

Да (логическое значение) – если скрытое поле ввода показано в диалоге.

Нет (логическое значение) – если скрытое поле ввода не показано в диалоге.

Примеры

Необходимо показать в диалоге поле ввода «ДатНач»:

```
оДиалог = Диалог ("Интервал дат")
оДиалог . ПоказатьВвод ("ДатНач") ;
оДиалог . Выполнить ( ) ;
```

См. также:

- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [СпрятатьВвод](#)
- [ОбновитьВвод](#)
- [ПоказатьПоле](#)

СостояниеВвода

Возвращает состояние указанного поля ввода в диалоге.

Синтаксис



Окно . СостояниеВвода (ИмяПоля)

Параметры

ИмяПоля (текст) – имя поля ввода в диалоге, состояние которого необходимо узнать.

Возвращает

Объект, содержащий в себе три переменные логического типа:

- «**Активен**» - установлен курсор в данном поле или нет;
- «**Показан**» - отображается данное поле в диалоге или нет;
- «**Разрешён**» - разрешён ввод данных в этом поле или нет.

Если указано несуществующее поле в диалоге, то функция возвратит пустой объект.

Примеры

В конфигурации рабочего места необходимо установить дату окончания рабочего периода «05.05.07», предварительно проверив, разрешён ли ввод данных в это поле:

```
оДиалог = Диалог ("Конфигурация рабочего места") ;  
оСостВвода = оДиалог . СостояниеВвода ("ДатКнц") ;  
Если (оСостВвода . Разрешен==Да)  
    оДиалог . ДатКнц="05 . 05 . 07" ;  
оДиалог . Выполнить ;
```

См. также:

- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)

ПометитьНеверныйВвод

Помечает в диалоге поле с неверным вводом данных, т.е выделяет его красным контуром.

Синтаксис

Окно . ПометитьНеверныйВвод (ИмяПоля)



Параметры

ИмяПоля (текст) – имя поля в диалоге с неверным вводом данных.

Возвращает

Да (логическое значение) – если поле с неверным вводом в диалоге помечено.

Нет (логическое значение) – если поле с неверным вводом в диалоге не помечено.

Примеры

Если в диалоге в поле "КодПодр" указан код, содержащий более 3 символов, необходимо пометить этот ввод, как неверный:

```
оДиалог = Диалог ("Паспорт" );
оДиалог . АктивироватьВвод ("КодПодр" );
оДиалог . Выполнить ( ) ;
пКод = оДиалог . КодПодр ;
Если (Размер (пКод) > 3)
    оДиалог . ПометитьНеверныйВвод ("КодПодр" ) ;
оДиалог . Выполнить ( ) ;
```

См. также:

- [РазметитьНеверныйВвод](#)
- [ЕстьВвод](#)

РазметитьНеверныйВвод

Снимает в диалоге пометку с поля с неверным вводом данных, т.е. убирает красную рамку вокруг этого поля.

Синтаксис

```
Окно . РазметитьНеверныйВвод (ИмяПоля)
```

Параметры

ИмяПоля (текст) – имя поля в диалоге с неверным вводом данных.

Возвращает



Да (логическое значение) – если снята пометка с поля с неверным вводом в диалоге.

Нет (логическое значение) – если не снята пометка с поля с неверным вводом в диалоге.

Примеры

Необходимо убрать пометку с поля "КодПодр" с неверным вводом в диалоге:

```
оДиалог = Диалог ("Паспорт" );  
оДиалог . АктивироватьВвод ("КодПодр" );  
оДиалог . Выполнить ( ) ;  
пКод = оДиалог . КодПодр ;  
Если (Размер (пКод) > 3)  
    оДиалог . ПометитьНеверныйВвод ("КодПодр" ) ;  
Сообщить (оДиалог . РазметитьНеверныйВвод ("КодПодр" ) ) ;  
оДиалог . Выполнить ( ) ;
```

См. также:

- [ПометитьНеверныйВвод](#)
- [ЕстьВвод](#)

ОбновитьВвод

Перерисовывает указанное или все поля ввода в диалоге.

Синтаксис

```
Окно . ОбновитьВвод ( [ИмяПоля] )
```

Параметры

ИмяПоля (текст) – имя поля ввода в диалоге, которое нужно обновить. Если параметр не указан, то функция обновляет все поля в диалоге.

Возвращает

Да (логическое значение) – если поле ввода обновилось.

Нет (логическое значение) – если поле ввода не обновилось.

Примеры

```
функция пОбновитьВвод (оФорма)
```



```
{
    оформа.Имя = "Введите текст";
    оформа.ОбновитьВвод(оформа.Имя);
}; # при вызове функции поле ввода Имя обновится, и
запишутся новые данные.
```

См. также:

- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [СпрятатьВвод](#)
- [ПоказатьВвод](#)

ЕстьВвод

Проверяет наличие в окне поля ввода с указанным именем.

Синтаксис

Окно.ЕстьВвод(ИмяПоляВвода)

Параметры

ИмяПоляВвода (текст) – название поля ввода, наличие которого надо проверить.

Возвращает

Да (логическое значение) – если на окне есть указанное поле.

Нет (логическое значение) – если поля с указанным именем не нашлось.

Пример

Необходимо узнать является ли поле с именем «выражение1» в окне калькулятора полем ввода:

```
оДиалог = Диалог("Калькулятор");
Сообщить(оДиалог.ЕстьВвод("выражение1")); # вернет
«Да»
```



См. также:

- [ПометитьНеверныйВвод](#)
- [РазметитьНеверныйВвод](#)
- [ОбновитьВвод](#)

СохранитьПоля

Сохраняет данные в полях ввода диалога, аналогично нажатию кнопки «Сохранить» в диалоге.

Синтаксис

Окно . СохранитьПоля ()

Параметры

Не указываются.

Возвращает

Да (логическое значение) – данные сохранены.

Нет (логическое значение) – данные не сохранены (например, не заполнено обязательное для заполнения поле).

Примеры

Необходимо, чтобы при открытии диалогового окна была проставлена указанная рабочая дата:

```
оДиалог = Диалог("Конфигурация системы");  
оДиалог.ТекДата = "13.03.05";  
оДиалог.СохранитьПоля();
```

См. также:

- [ИзменилосьПоле](#)
- [Окно](#)
- [Диалог](#)

ИзменилосьПоле

Сообщает программе об изменении поля ввода в диалоге.

Синтаксис

`Окно.ИзменилосьПоле (ИмяПоля)`

Параметры

ИмяПоля (текст) – имя поля ввода в диалоге, которое изменяется.

Возвращает

Да (логическое значение) - если поле изменилось.

Нет (логическое значение) – если поле не изменилось.

Примеры

Необходимо узнать, изменилось ли поле «ДатКнц»:

```
одиалог = Диалог ("Интервал дат");  
одиалог.ДатКнц = "13.03.05";  
Сообщить (одиалог.ИзменилосьПоле ("ДатКнц"));
```

См. также:

- [СохранитьПоля](#)
- [Окно](#)
- [Диалог](#)

УстановитьСписокПунктов

Устанавливает для поля в диалоге выпадающий список указанных значений.

Синтаксис

`Окно.УстановитьСписокПунктов (ИмяПоля, ИмяМассива)`

Параметры

ИмяПоля (текст) – имя поля, для которого мы создаем список пунктов.

ИмяМассива (массив) – список пунктов.



Возвращает

Да (логическое значение) – если список значений установлен для указанного поля.

Нет (логическое значение) – если список значений не установлен для указанного поля.

Примеры

Установим в поле для ввода штрих-кода диалога "Ввод строки" выпадающий список возможных значений номеров номенклатуры:

```
оД = Диалог ("Ввод строки") ;  
мМ = Массив ("0901", "05", "У301", "У306") ;  
оД.УстановитьСписокПунктов ("Строка", мМ) ;  
оД.Выполнить () ;
```

См. также:

- [УстановитьТекст](#)
- [УстановитьПодсказку](#)

ПолучитьСписокПунктов

Возвращает список установленных для указанного поля в диалоге значений.

Синтаксис

Окно . ПолучитьСписокПунктов (ИмяПоля)

Параметры

ИмяПоля (текст) – имя поля, для которого необходимо получить список возможных значений.

Возвращает

Массив, список установленных для указанного поля в диалоге значений.

Примеры

Узнаем список значений в выпадающем списке поля «Используемое подключение» в диалоге "Конфигурация рабочего места":

```
оД = Диалог ("Конфигурация рабочего места") ;
```



```
оД.Модем=="Использовать модем";  
мСписок=оД.ПолучитьСписокПунктов("ИмяСоединения");
```

```
оД = Диалог("Ввод строки");  
мМ = Массив("0901", "05", "У301", "У306");  
оД.УстановитьСписокПунктов("Строка", мМ);  
мСписок=оД.ПолучитьСписокПунктов("Строка");  
ДляВсех(Переменных(мСписок, мЭлемент))  
    Сообщить(мСписок[мЭлемент])
```

См. также:

- [УстановитьТекст](#)
- [УстановитьПодсказку](#)

УстановитьТекст

Устанавливает новый текст или картинку в поле ввода.

Синтаксис

Окно.УстановитьТекст(ИмяПоля, НовТекстИлиКартинка)

Параметры

ИмяПоля (текст) – имя поля, в которое нужно вставить новый текст. Поле, в котором нужно заменить картинку, должно быть типа «Картинка».

НовТекстИлиКартинка (текст) – строка с новым текстом или имя файла с картинкой. Необходимо указывать полный путь к файлу, иначе программа будет искать этот файл в каталоге «Модули». Все картинки к программе по умолчанию находятся в этом каталоге.

Возвращает

Строка с прежним текстом или именем файла прежней картинки.

Примеры

Необходимо поменять старое название кнопки «Сохранить» на новое «Ок» в диалоге «Конфигурация системы»:

```
оДиалог = Диалог("Конфигурация системы");  
Сообщить(оДиалог.УстановитьТекст("Сохранить",
```



```
"Ок" ) ) ;  
оДиалог.Выполнить ( ) ; # в результате название кнопки  
поменяется на «Ок», а функция возвратит прежнее  
название «Сохранить» .
```

Нужно поменять картинку в диалоге:

```
оДиалог=Диалог ("Пример с картинкой" ) ;  
оДиалог.Выполнить ;  
Сообщить (оДиалог.УстановитьТекст ("Картинка" ,  
"advice.gif" ) ) ;  
оДиалог.Выполнить ;
```

См. также:

- [УстановитьСписокПунктов](#)
- [УстановитьПодсказку](#)

УстановитьПодсказку

Устанавливает всплывающую подсказку для поля в диалоге, т.е. при наведении указателем мышки на поле в диалоге будет появляться поясняющий текст.

Синтаксис

`Окно.УстановитьПодсказку (ИмяПоля , ТекстПодсказки)`

Параметры

ИмяПоля (текст) – имя поля в диалоге, для которого устанавливается всплывающая подсказка.

ТекстПодсказки (текст) – строка с текстом всплывающей подсказки.

Возвращает

Ничего не возвращает.

См. также:

- [УстановитьСписокПунктов](#)
- [УстановитьТекст](#)

ЗапретитьФлаги

Запрещает указанные флаги в диалоге, т.е. выделяет их серым цветом.

Синтаксис

```
Окно.ЗапретитьФлаги (ИмяПоляФлагов, ИмяФлага1, ....)  
Окно.ЗапретитьФлаги (ИмяПоляФлагов, НомерФлага1,  
....)
```

Параметры

ИмяПоляФлагов (текст) – имя поля типа «Флаги» в диалоге.

ИмяФлага1,... (текст) – названия поля флагов, которые нужно запретить в диалога.

НомерФлага1,... (целое) – порядковые номера флагов в диалоге.

Возвращает

Да (логическое значение) – если флаг запрещен.

Нет (логическое значение) – если флаг не запрещен.

Примеры

В диалоге "Конфигурация задачи 'Товары и материалы'" запретим второе предупреждение:

```
оДиалог = Диалог ("Конфигурация задачи 'Товары и  
материалы' ");  
оДиалог.ЗапретитьФлаги ("Флаг1", 2);
```

См. также:

- [РазрешитьФлаги](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [Диалог](#)

РазрешитьФлаги

Разрешает запрещенные флаги в диалоге. Функция обратная функции «ЗапретитьФлаги».



Синтаксис

`Окно.РазрешитьФлаги(ИмяПоляФлагов, ИмяФлага1,)`
`Окно.РазрешитьФлаги(ИмяПоляФлагов, НомерФлага1,)`

Параметры

ИмяПоляФлагов (текст) – имя поля типа «Флаги».

ИмяФлага1,... (текст) – названия флагов, которые нужно разрешить в поле диалога.

НомерФлага1,... (целое) – порядковые номера флагов в диалоге.

Возвращает

Да (логическое значение) – если флаг разрешен.

Нет (логическое значение) – если флаг не разрешен.

Примеры

В диалоге "Конфигурация задачи 'Товары и материалы'" разрешить запрещенное второе предупреждение:

```
оДиалог = Диалог("Конфигурация задачи 'Товары и материалы'");  
оДиалог.ЗапретитьФлаги("Флаг1", 2);  
оДиалог.РазрешитьФлаги("Флаг1", 2);  
оДиалог.Выполнить();
```

См. также:

- [ЗапретитьФлаги](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [Диалог](#)

Координаты

Позволяет узнать координаты и размеры элемента окна или диалога.

Синтаксис

`Окно.Координаты(ИмяЭлемента)`

Параметры

ИмяЭлемента (текст) – имя элемента окна или диалога (например, поля или выборки), координаты которого необходимо узнать.

Возвращает

Объект со следующим набором полей: **Лево** (расстояние в пикселях от левого края окна), **Верх** (расстояние в пикселях от верхнего края окна), **Право** (расстояние в пикселях от правого края окна), **Низ** (расстояние в пикселях от нижнего края окна), **Ширина** (ширина элемента в пикселях), **Высота** (высота элемента в пикселях), **ТекПозГор** (позиция элемента по горизонтали), **ТекПозВерт** (позиция элемента по вертикали).

Если указывается несуществующий элемент, то функция вернет пустой объект.

См. также:

- [Окно](#)
- [Диалог](#)
- [УстановитьКоординаты](#)

УстановитьКоординаты

Позволяет изменить координаты и размеры элемента окна или диалога.

Синтаксис

`Окно.УстановитьКоординаты(ИмяЭлемента, Лево, Верх[,
Ширина, Высота])`

Параметры

ИмяЭлемента (текст) – имя элемента окна или диалога, положение которого необходимо изменить.

Лево, Верх, Ширина, Высота (целое) – новые координаты и размеры элемента.

Возвращает



Да (логическое значение) – если были установлены новые координаты.

Нет (логическое значение) – если не удалось установить указанные координаты.

См. также:

- [Окно](#)
- [Диалог](#)
- [Координаты](#)

Страницы в диалого-помощнике

ПоказатьСтраницу

Переходит к указанной странице в диалого-помощнике (диалог, у которого есть страницы и кнопки для перехода между страницами, например, «Назад», «Далее», «Готово»).

Синтаксис

Диалог-помощник . ПоказатьСтраницу (ИмяСтраницы)

Параметры

ИмяСтраницы (текст) – имя страницы, которую нужно показать. Список страниц можно посмотреть в описании диалого-помощника в ресурсе.

Комментарии

Функция покажет страницу, при условии, что указанный диалого-помощник открыт.

Возвращает

Ничего не возвращает.

Примеры

В мастере создания налогоплательщика необходимо показать страницу «Общие сведения об организации»:

`oДиалог=Диалог ("Мастер создания налогоплательщика") ;`

оДиалог.ПоказатьСтраницу ("Общие сведения об организации") ;

оДиалог.Выполнить ;

См. также:

- [СброситьПоследниеСтраницы](#)
- [СделатьСтраницуПоследней](#)
- [ТекущаяСтраница](#)

СброситьПоследниеСтраницы

Убирает признак последней страницы (кнопку «Готово») со страницы и переходит к начальной (стартовой) странице в диалоге-помощнике (диалог, у которого есть страницы и кнопки для перехода между страницами, например, «Назад», «Далее», «Готово»).

Синтаксис

Диалог-помощник.СброситьПоследниеСтраницы ()

Параметры

Не указываются.

Возвращает

Ничего не возвращает.

См. также:

- [ПоказатьСтраницу](#)
- [СделатьСтраницуПоследней](#)
- [ТекущаяСтраница](#)

СделатьСтраницуПоследней

Добавляет на указанную страницу в диалоге-помощнике признак последней страницы (кнопку «Готово»).



Синтаксис

Диалог-помощник . СделайтеСтраницуПоследней (ИмяСтраницы)

Параметры

ИмяСтраницы (текст) – имя страницы, которую нужно сделать завершающей в прохождении диалога-помощника. Список страниц можно посмотреть в описании диалога-помощника в ресурсе.

Возвращает

Да (логическое значение) – если удалось сделать страницу последней.

Нет (логическое значение) – если не удалось сделать страницу последней.

Примеры

В мастере создания налогоплательщика необходимо сделать страницу «Общие сведения об организации» завершающей:

```
оДиалог=Диалог ("Мастер создания налогоплательщика") ;  
оДиалог.ПоказатьСтраницу ("Общие сведения об организации") ;  
оДиалог.Выполнить ;  
оДиалог.СделайтеСтраницуПоследней ("Общие сведения об организации") ;  
оДиалог.Выполнить ;
```

См. также:

- [СброситьПоследниеСтраницы](#)
- [СделатьСтраницуПоследней](#)
- [ТекущаяСтраница](#)

ТекущаяСтраница

Возвращает название текущей страницы в диалоге-помощнике.

Синтаксис

Диалог-помощник . ТекущаяСтраница ()

Параметры

Не указываются.

Возвращает

Текст, имя страницы, которая на данном шаге прохождения диалога-помощника является текущей.

Примеры

В мастере создания налогоплательщика необходимо сделать страницу «Общие сведения об организации» завершающей:

```
оДиалог=Диалог("Мастер создания налогоплательщика");  
оДиалог.ПоказатьСтраницу("Общие сведения об  
организации");  
оДиалог.Выполнить;  
оДиалог.СделатьСтраницуПоследней("Общие сведения об  
организации");  
оДиалог.Выполнить;
```

См. также:

- [СброситьПоследниеСтраницы](#)
- [СделатьСтраницуПоследней](#)
- [ТекущаяСтраница](#)

Команды для окон и диалогов

ВыполнитьКоманду – для окна

Выполняет указанную команду в окне. Команды можно посмотреть, открыв любое меню в файле описания ресурсов.

Синтаксис

```
Окно.ВыполнитьКоманду(ИмяКоманды[, ИмяПоля])
```

Параметры

ИмяКоманды (текст) – имя команды, которую необходимо выполнить.



ИмяПоля (текст) – имя поля, для которого задается команда. Если параметр не указывается, то выполняется команда для текущего поля.

Возвращает

Да (логическое значение) – если объект принял команду к выполнению. Но это совсем не значит, что команда будет выполнена.

Нет (логическое значение) – если команда не выполнена.

Примеры

При вызове указанного обработчика, документ с отчетом будет закрыт, т.е. недоступен для дальнейшего редактирования:

```
функция НаОтчетЗакреть ()
{
    ЗаписатьОбъект (ПараметрыОтчета, оИнф) ;
    пРезультат = Документ.ВыполнитьКоманду
        ("CloseDoc") ;
    Вернуть (СледФункция) ;
}
```

См. также:

- [ВыполнитьКоманду](#)

ВыполнитьКоманду

Выполняет указанную команду в программе. Команды можно посмотреть, открыв любое меню в файле описания ресурсов.

Синтаксис

ВыполнитьКоманду (**ИмяКоманды**)

Параметры

ИмяКоманды (текст) – имя команды, которую необходимо выполнить.

Возвращает

Да (логическое значение) – если объект принял команду к выполнению. Но это совсем не значит, что команда будет выполнена.

Нет (логическое значение) – если команда не будет выполнена.

Примеры

Откроем окно для выбора таблицы базы данных:

```
ВыполнитьКоманду ("OpenMainDBTable");
```

См. также:

- [ВыполнитьКоманду – для окна](#)
- [ПоместитьВОчередьКоманд](#)

ПоместитьВОчередьКоманд

Помещает команды в очередь на выполнение в программе.

Синтаксис

```
ПоместитьВОчередьКоманд (ИмяКоманды1 [ , ИмяКоманды2 ] [ ,  
ИмяКоманды3 ] . . . )
```

Параметры

ИмяКоманды1, ИмяКоманды2, ИмяКоманды3... (текст) – имя команды, помещаемой в очередь на выполнение. Команды вызываются для текущего окна в программе и, если они к нему применимы, то выполняются.

Возвращает

Да (логическое значение) – если команда помещена в очередь.

Нет (логическое значение) – если команда не помещена в очередь.

Примеры

Необходимо поочередно выполнить следующие команды: добавить новую запись в таблице, затем открыть окно для выбора таблицы из базы данных и затем открыть окно с параметрами выборки:

```
ПоместитьВОчередьКоманд ("InsRec" , "OpenMainDBTable" ,  
"SelParam"); # в итоге будут выполнены первые две  
команды. Последняя выполнена не будет, т.к. после  
выполнения первых двух команд текущим окном будет  
окно выбора таблицы из базы данных, а для этого окна  
не применима команда отбора элементов.
```

**См. также:**

- [ВыполнитьКоманду](#)

ПоместитьНаЭкран

Выводит информацию на экран указанного монитора.

Синтаксис

ПоместитьНаЭкран (**НомерЭкрана**)

Параметры

НомерЭкрана (целое) – номер монитора, на который необходимо вывести информацию.

Комментарий

Функция .применяется при работе СБиС++ на нескольких мониторах.

Возвращает

Да (логическое значение) – удалось вывести информацию на экран указанного монитора.

Нет (логическое значение) – не удалось вывести информацию на экран указанного монитора.

Примеры

Необходимо на втором мониторе отобразить окно настроек рабочего места:

```
оДиалог = Диалог("Конфигурация рабочего места");  
оДиалог.ПоместитьНаЭкран(2);  
оДиалог.Выполнить();
```

ПослатьСообщение

Посылает сообщение текущему окну и его объектам.

Синтаксис

Окно.ПослатьСообщение (**Имя** [, **Объект**])

Параметры

Имя (текст) – имя сообщения.

Объект (объект) – объект с данными, которому посылается сообщение. Если не параметр не указывается, то посылается сообщение текущему объекту.

Возвращает

Да (логическое значение) – сообщение отправлено.

Нет (логическое значение) – сообщение не отправлено.

Примеры

```
Функция НаИзменениеПоля (ИмяПоля)
{
    переменная пИмя;
    оИнф [ИмяПоля] = Table4 [ИмяПоля] ;
    оИнф = РассчитатьФормаОтчетности (оИнф) ;
    ДляВсех (Полей (Table4, пИмя))
        Table4 [пИмя] = оИнф [пИмя] ;
    Table3.ПослатьСообщение ("ОбновитьДанные") ;
    Вернуть (1) ;
}; # при изменении данных в Table4 посылается
сообщение для обновления данных в Table3.
```

См. также:

- [ПослатьСообщениеВсем](#)

ПослатьСообщениеВсем

Рассылка оповещений в программе.

Синтаксис

```
ПослатьСообщениеВсем (Имя [ , Объект ] )
```

Параметры

Имя (текст) – имя сообщения.



Объект (объект) – объект с данными, который посылается обработчику сообщения. Если параметр не указан, посылается пустой объект.

Возвращает

Да (логическое значение) – сообщение отправлено.

Нет (логическое значение) – сообщение не отправлено.

Примеры

Посылаем всем открытым окнам в программе сообщение "Обновить данные":

```
ПослатьСообщениеВсем ("Обновить данные") ;
```

См. также:

- [ПослатьСообщение](#)

ПраваНаКоманду

Проверяет наличие прав у текущего пользователя на указанную команду.

Синтаксис

```
ПраваНаКоманду (Окно , Выборка , Папка , Команда)
```

Параметры

Окно (текст) – имя окна, в котором необходимо проверить наличие прав на команду.

Выборка (текст) – имя выборки окна.

Папка (текст) – имя папки выборки, для которой будут проверяться права на указанную команду. Если указать «», то будут проверяться права на команду в корне.

Команда (текст) – имя команды. Название команды прописано в ресурсах.

Возвращает

Да (логическое значение) – у пользователя есть права на данную команду.

Нет (логическое значение) – у пользователя нет прав на команду.

Пример

Необходимо узнать наличие прав у пользователя на закрытие документов в корне реестра «Расходные накладные»:

```
Сообщить (ПраваНаКоманду ( "Расходные накладные" ,  
"Расходные накладные" , "" , "DocClose" ) );
```

СменитьФоновоеОкно

Устанавливает новое фоновое окно для текущей задачи (в ресурсах ничего не меняется, меняется только на момент выполнения).

Синтаксис

```
СменитьФоновоеОкно (НазваниеОкна)
```

Параметры

НазваниеОкна (текст) – название фонового окна, описанного в ресурсах программы, например, «фнс_окно», «стат_окно», и т.д.

Комментарии

Функция используется в основном в ЭО, в интерфейсе которой присутствуют фоновые окна, например, окно формирования бухгалтерской и налоговой отчетности, отчетности в ПФ и другой регламентированной отчетности.

Возвращает

Ничего не возвращает.

Пример

Фрагмент использования функции при выполнении оценки риска налоговой проверки. Если анализируемая отчетность не сформирована, то программа предлагает перейти в окно «ФНС» и ввести необходимые отчеты вручную:

```
функция КнопкаВвестиОтчетыВручную ()  
{  
    Сообщить ("Сейчас данное окно будет закрыто. "  
        + "Вам будет открыт реестр отчетов.  
        Введите все отчеты за период, который хотите
```

```
проанализировать . "
    + "После этого снова нажмите кнопку
\"Оценка риска налоговой проверки\" . " ) ;
{
    ВыполнитьКоманду ("Cancel") ;
    СменитьФоновоеОкно ("фнс_окно") ;
}
```

Таблицы в окнах и диалогах

Запись – для окна/диалога

Возвращает запись указанной таблицы в окне или диалоге.

Синтаксис

```
Окно.Запись ( [ИмяТаблицы] )
```

Параметры

ИмяТаблицы (текст) - имя таблицы с записями (браузера). Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Объект – запись указанной таблицы в окне или диалоге.

Примеры

Выполним выбор записи в окне «Виды цен»:

```
oОкно = Окно ("Виды цен") ;
oОкно.ВыполнитьВыбор () ;
oНовЗапись = oОкно.Запись () ;
```

Скопируем текущую запись документа – основания в расходной накладной, при этом сумма у новой записи будет нулевая:

```
oЗапись = Запись ("ДокСледствие") ;
oНовЗапись = СкопироватьОбъект (oЗапись) ;
oНовЗапись.Очистить () ;
```

```
Связать (оНовЗапись . ДокОснование ,  
оЗапись . ДокОснование) ;  
оНовЗапись . Добавить ( ) ;
```

См. также:

- [Окно](#)

УстановитьКурсор

Устанавливает курсор на указанной записи в таблице в окне или диалоге.

Синтаксис

```
Окно . УстановитьКурсор (Запись)  
Окно . УстановитьКурсор ( [НомерЗаписи [ , ИмяТаблицы] ] )
```

Параметры

Запись (объект) – запись, на которой нужно установить курсор.

НомерЗаписи (целое) – порядковый номер записи (начиная с 1), на которой устанавливается курсор.

ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Целое, номер записи, на которой установлен курсор.

Пример

```
оОкно = Окно ("Сотрудники") ;  
оОкно . УстановитьКурсор (3) ; # курсор установлен на 3  
записи реестра "Сотрудники" .  
оОкно . ИзменитьЗапись ( ) ; # изменим эту запись .
```

При сохранении приходной накладной, если в таблице наименований присутствует услуга, то будет выдано сообщение об ошибке и курсор будет установлен на этой записи. Документ при этом сохранен не будет:

```
перем пНомерЗаписи = 1;  
ДляВсех (Наименований ())  
{  
    Если (Наим. Тип=="Услуга")  
    {  
  
        УстановитьКурсор (пНомерЗаписи, "ДокументПриходы");  
        Ошибка ("В документе обнаружена услуга");  
    }  
    пНомерЗаписи ++;  
}
```

См. также:

- [Окно](#)
- [Диалог](#)
- [Запись – для окна/диалога](#)

ДобавитьЗапись

Добавляет запись в таблице в окне или диалоге (аналогично нажатию клавиши <Insert>), т.е открывает диалоговое окно для создания новой записи.

Синтаксис

```
Окно.ДобавитьЗапись ([ИмяТаблицы] [, Запись])
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями, в которую нужно добавить новую запись. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Запись (объект) – запись, которую необходимо добавить в таблицу. Если параметр не указан, то функция добавляет новую запись в таблицу.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – если запись добавлена.

Нет (логическое значение) – если запись не была добавлена.

Примеры

Добавим новую запись в справочнике «Корректирующие документы»:

```
oОкно = Окно ("Корректирующие документы") ;  
oОкно.ДобавитьЗапись ( ) ;
```

См. также:

- [ИзменитьЗапись](#)
- [ОтметитьЗапись](#)

ИзменитьЗапись

Изменяет запись в таблице в окне или диалоге (аналогично нажатию <F3> на записи), т.е. открывает диалоговое окно записи для редактирования.

Синтаксис

```
Окно.ИзменитьЗапись ( [ИмяТаблицы] [ , Запись ] )
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями (браузера) в окне или диалоге, в которой нужно изменить запись. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Запись (объект) – запись в таблице, которую необходимо изменить. Если параметр не указан, функция берет текущую запись в таблице для изменения.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – если нажали кнопку «Сохранить».

Нет (логическое значение) – если действие отменили.

Примеры

Изменим выбранную запись в окне «Организации»:

```
oОкно = Окно ("Организации" );
если (oОкно . ВыполнитьВыбор ( ) )
{
    перем oЗ = oОкно . Запись ( ) ;
    oОкно . ИзменитьЗапись ( ) ;
}
```

Нужно перебрать наименования накладной и изменить те, у которых сумма цен не проставлена:

```
ДляВсех (Наименований ( ) )
    Если (СуммаЦен=="")
        ИзменитьЗапись (Наим) ;
```

См. также:

- [ДобавитьЗапись](#)
- [ОтметитьЗапись](#)

ОтметитьЗапись

Выделяет в таблице с записями в окне или диалоге записи (аналогично нажатию клавиши <Пробел>).

Синтаксис

```
Окно . ОтметитьЗапись (Запись )
Окно . ОтметитьЗапись ( [ИмяТаблицы , ] НомерЗаписи )
```

Параметры

Запись (объект) – запись в таблице в окне или диалоге, которую нужно отметить.

ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге, в которой нужно отметить запись. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

НомерЗаписи (число) – порядковый номер записи в таблице в окне или диалоге.



Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – запись отмечена.

Нет (логическое значение) – запись не отмечена.

Примеры

Выделим текущую запись в окне «Организации»:

```
oОкно = НайтиОкно ("Организации") ;
```

```
oЗапись = oОкно.Запись ( ) ;
```

```
oОкно.ОтметитьЗапись (oЗапись) ;
```

Выделим вторую запись в окне:

```
oОкно = НайтиОкно ("Организации") ;
```

```
oОкно.ОтметитьЗапись (2) ;
```

См. также:

- [СнятьОтметкуЗаписей](#)
- [ЕстьОтмеченныеЗаписи](#)
- [ДляВсех\(ОтмеченныхЗаписей\)](#)
- [ДляВсех\(ОтмеченныхИлиВсехЗаписей\)](#)

РазметитьЗапись

Снимает отметку указанной записи в таблице окна или диалога. Функция обратная функции «ОтметитьЗапись».

Синтаксис

```
Окно.РазметитьЗапись (Запись)
```

```
Окно.РазметитьЗапись ( [ИмяТаблицы, ]НомерЗаписи)
```

Параметры

Запись (объект) – запись в таблице окна или диалога, с которой нужно снять отметку.



ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге, в которой нужно разметить запись. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

НомерЗаписи (число) – порядковый номер записи в таблице в окне или диалоге, с которой нужно снять отметку.

Комментарий

Диалог обязательно должен содержать таблицу с записями. Функция работает так же, как и функция **СнятьОтметкуЗаписей**, с той лишь разницей, что последняя снимает отметку не у конкретной записи, а у всех записей таблицы.

Возвращает

Да (логическое значение) – отметка с записи снята.

Нет (логическое значение) – не удалось снять отметку с записи.

Примеры

Снимем отметку с 6-ой записи в окне «Организации»:

```
oОкно = НайтиОкно ("Организации") ;  
oОкно. РазметитьЗапись ("Организации" , 6) ;
```

См. также:

- [ОтметитьЗапись](#)
- [ЕстьОтмеченныеЗаписи](#)
- [ДляВсех\(ОтмеченныхЗаписей\)](#)
- [ДляВсех\(ОтмеченныхИлиВсехЗаписей\)](#)
- [СнятьОтметкуЗаписей](#)

СнятьОтметкуЗаписей

Снимает отметку записей в таблице в окне или диалоге. Функция обратная функции «ОтметитьЗапись».

Синтаксис

```
Окно.СнятьОтметкуЗаписей ( [ИмяТаблицы] )
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге, в которой нужно снять отметку с записей. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – отметка записей снята.

Нет (логическое значение) – отметка записей не снята.

Примеры

Снимем отметку записей в окне «Организации»:

```
oОкно = НайтиОкно ("Организации") ;  
oОкно . СнятьОтметкуЗаписей ( ) ;
```

См. также:

- [ОтметитьЗапись](#)
- [РазметитьЗапись](#)
- [ЕстьОтмеченныеЗаписи](#)
- [ДляВсех\(ОтмеченныхЗаписей\)](#)
- [ДляВсех\(ОтмеченныхИлиВсехЗаписей\)](#)

ЕстьОтмеченныеЗаписи

Проверяет существование отмеченных записей в таблице в окне или диалоге.

Синтаксис

```
Окно . ЕстьОтмеченныеЗаписи ( [ИмяТаблицы] )
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге, в которой происходит проверка существования отмеченных записей. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – если есть отмеченные записи в окне.

Нет (логическое значение) – если нет отмеченных записей в окне.

Примеры

Проверим наличие отмеченных записей в окне:

```
оОкно = НайтиОкно ("Организации") ;  
оОкно.ЕстьОтмеченныеЗаписи ( ) ;
```

См. также:

- [ОтметитьЗапись](#)
- [СнятьОтметкуЗаписей](#)
- [ДляВсех\(ОтмеченныхЗаписей\)](#)
- [ДляВсех\(ОтмеченныхИлиВсехЗаписей\)](#)

ДляВсех(ОтмеченныхЗаписей)

Перебирает отмеченные записи в таблице браузера.

Синтаксис

```
ДляВсех (Окно.ОтмеченныхЗаписей ( [ИмяТаблицы] ) )
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями в окне или диалоге, в которой будут перебираться отмеченные записи. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Браузер должен содержать таблицу с записями. Если отмеченных записей в таблице нет, то цикл будет выполнен только для текущей записи. Стоит отметить, что у схематического браузера не существует текущей записи, поэтому в случае отсутствия выделенных записей цикл не будет работать ни разу.

Возвращает

Ничего не возвращает.

Примеры

Переберем в справочнике «Организации» отмеченные записи и узнаем их название:

```
oОкно = НайтиОкно ("Организации") ;  
ДляВсех (oОкно . ОтмеченныхЗаписей ( ) )  
    Сообщить (Название) ;
```

Увеличим сумму себестоимости отмеченных наименований в приходной накладной на 10%:

```
ДляВсех (ОтмеченныхЗаписей ("ДокументПриходы" ) )  
    Запись . СуммаСебест = Запись . СуммаСебест +  
    Запись . СуммаСебест * 0 . 1 ;
```

См. также:

- [ОтметитьЗапись](#)
- [СнятьОтметкуЗаписей](#)
- [ДляВсех\(ОтмеченныхИлиВсехЗаписей\)](#)
- [ЕстьОтмеченныеЗаписи](#)

ДляВсех(ОтмеченныхИлиВсехЗаписей)

Перебирает отмеченные записи в таблице окна или диалога, либо все записи таблицы, если нет отмеченных.

Синтаксис

```
ДляВсех (Окно . ОтмеченныхИлиВсехЗаписей ( [ИмяТаблицы] ) )
```



Параметры

ИмяТаблицы (текст) – имя таблицы с записями в открытом окне или диалоге, в которой перебираются записи. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями.

Комментарий

Окно или диалог должен содержать таблицу с записями.

Возвращает

Ничего не возвращает.

Примеры

Найдем в справочнике организаций запись с указанным ИНН и сообщим название этой организации:

```
oОкно = НайтиОкно ("Организации") ;  
перем пИНН = "7604006689" ;  
ДляВсех (oОкно . ОтмеченныхИлиВсехЗаписей ( )  
    Если (ИНН == пИНН)  
        Сообщить (Название) ;
```

См. также:

- [ОтметитьЗапись](#)
- [СнятьОтметкуЗаписей](#)
- [ЕстьОтмеченныеЗаписи](#)
- [ДляВсех\(ОтмеченныхЗаписей\)](#)

ВставитьПоле

Добавляет поле в таблице в окне или диалоге. Аналог команды «Вставить столбец» (<Shift+Insert>). Функция обратная функции «УдалитьПоле».

Синтаксис

```
Окно . ВставитьПоле (ИмяПоля)  
Окно . ВставитьПоле (ИмяТаблицы , ИмяПоля [ ,  
    НазваниеПоля [ , ВставлятьсяПосле ] ] )
```

Параметры

ИмяТаблицы (текст) – имя таблицы с записями, в которую нужно добавить поле. Параметр имеет смысл применять, если окно или диалог содержит более одной таблицы с записями. Если параметр не указан, то поле вставляется в первую попавшуюся таблицу в окне.

ИмяПоле (текст) – имя поля, которое должно быть добавлено в таблицу. Имя поля указывается таким, каким оно значится в базе данных.

НазваниеПоля (текст) – пользовательское название поля, которое добавляется в таблицу. Если параметр не указан, то в этом случае название поля будет совпадать с именем этого поля в базе данных.

ВставлятьПосле (текст) – имя поля, после которого вставляется указанное поле. Если у параметра указать «-», то поле добавляется перед указанным полем. Если параметр не указан, то поле добавляется после всех отображаемых полей в таблице.

Комментарий

Диалог должен содержать таблицу с записями.

Возвращает

Да (логическое значение) – если поле добавлено в таблицу.

Нет (логическое значение) – если поле не добавлено в таблицу.

Примеры

Добавим поле «ОКПО» в справочнике организаций:

```
oОкно=Окно ("Организации") ;  
oОкно.ВставитьПоле ("ОКПО") ;  
oОкно.Открыть () ;
```

Поместим поле «ОКПО» перед полем «Телефон»:

```
oОкно=Окно ("Организации") ;  
Сообщить (oОкно.ВставитьПоле ("Организации", "ОКПО",  
"ОКПО Организации", "-Телефон")) ;  
oОкно.Открыть () ;
```

См. также:

- [СпрятатьПоле](#)

- [УдалитьПоле](#)

УдалитьПоле

Удаляет указанное поле из таблицы в окне или диалоге. Аналог команды «Удалить столбец» (<Shift+Delete>). Функция обратная функции «ВставитьПоле».

Синтаксис

`Окно.УдалитьПоле (ИмяПоля [, ИмяТаблицы])`

Параметры

ИмяПоля (текст) – имя поля в таблице, которое необходимо удалить.

ИмяТаблицы (текст) – имя таблицы в диалоге, поле в которой необходимо удалить.

Возвращает

Да (логическое значение) – если поле в таблице удалено.

Нет (логическое значение) – если не удалось удалить поле в таблице.

Примеры

Необходимо удалить столбец «ИНН» в окне организаций:

```
oОкно = Окно ("Организации")
oОкно.УдалитьПоле ("ИНН") ;
oОкно.Открыть ( ) ;
```

См. также:

- [ПоказатьПоле](#)
- [ВставитьПоле](#)

СпрятатьПоле

Скрывает указанное поле в таблице окна или диалога. Функция обратная функции «ПоказатьПоле».

Синтаксис

`Окно.СпрятатьПоле (ИмяПоля [, ИмяТаблицы])`

Параметры

ИмяПоля (текст) – имя поля в таблице, которое необходимо скрыть.

ИмяТаблицы (текст) – имя таблицы в диалоге, поле в которой необходимо скрыть.

Возвращает

Да (логическое значение) – если поле в таблице скрыто.

Нет (логическое значение) – если не удалось скрыть поле в таблице.

Примеры

Необходимо скрыть столбец «ИНН» в окне организаций:

```
oОкно = Окно ("Организации")
oОкно. СпрятатьПоле ("ИНН") ;
oОкно. Открыть ( ) ;
```

См. также:

- [ПоказатьПоле](#)
- [ВставитьПоле](#)

ПоказатьПоле

Отображает скрытое поле в таблице окна или диалога. Функция обратная функции «СпрятатьПоле».

Синтаксис

```
Окно. ПоказатьПоле (ИмяПоля [ , ИмяТаблицы ] )
```

Параметры

ИмяПоля (текст) – имя поля в таблице, которое необходимо отобразить.

ИмяТаблицы (текст) – имя таблицы в диалоге, поле в которой необходимо отобразить.

Комментарии

Функция отображает скрытое поле в конце таблицы, если поле было скрыто функцией «УдалитьПоле», или в том месте, где и было поле изначально, если поле было скрыто функцией «СпрятатьПоле».

Возвращает

Да (логическое значение) – если скрытое поле в таблице отображено.

Нет (логическое значение) – если не удалось отобразить скрытое поле в таблице.

Примеры

Необходимо отобразить столбец «ОКПО» в окне организаций:


```
oОкно = Окно ("Организации")
oОкно . ПоказатьПоле ("ОКПО") ;
oОкно . Открыть ( ) ;
```

См. также:

- [АктивироватьВвод](#)
- [ЗапретитьВвод](#)
- [РазрешитьВвод](#)
- [СпрятатьВвод](#)
- [СпрятатьПоле](#)
- [ОбновитьВвод](#)

Установка параметров в отчетах

ВыбратьЭлемент



Служит для добавления варианта детализации данных в отчете по вертикали (по аналогии с нажатием кнопки «Добавить детализацию»  в диалоге отчета).

Синтаксис

```
Диалог . ВыбратьЭлемент (ВариантДетализации [ , Да | Нет ] )
```

Параметры

Вариант Детализации (текст) – вариант детализации данных в отчете. Список возможных вариантов можно узнать по кнопке <In> в блоке «Детализация по горизонтали».

Да|Нет (логическое) – если указано значение «Да», то добавляется детализация по иерархии (то есть с указанием папок, у детализации появится значок ). По умолчанию, предполагается значение «Нет», то есть добавление детализации без иерархии (в отчет будут выводиться только содержимое папок, без указания самих папок, значок  будет отсутствовать).

Возвращает

Да (логическое значение) – если удалось установить детализацию.

Нет (логическое значение) – если не удалось установить детализацию.

Примеры

В отчете «Остатки» добавим вариант детализации выводимых данных по поставщикам с указанием папок:

```
oДиалог=Диалог ("Остатки" ) ;  
oДиалог . ОчиститьЭлементы ( ) ;  
oДиалог . Выполнить ( ) ;  
oДиалог . ВыбратьЭлемент ("Поставщик" , Да ) ;  
oДиалог . Выполнить ( ) ;
```

ОчиститьЭлементы

Инициализирует поля в диалоге отчета: сбрасывает всю вертикальную детализацию и все установленные фильтры.

Синтаксис

```
Диалог . ОчиститьЭлементы ( )
```

Параметры

Не указываются.

Возвращает



Да (логическое значение) – если удалось очистить параметры в отчете.

Нет (логическое значение) – если не удалось очистить параметры в отчете.

Примеры

Пример приводится в описании функции «ВыбратьЭлемент».

ОбновитьПараметры

Обновляет параметры в отчете после предварительных изменений параметров.

Синтаксис

`Диалог.ОбновитьПараметры ()`

Параметры

Не указываются.

Возвращает

Ничего не возвращает.

УстановитьФильтр

Устанавливает значение фильтра (в группе «Фильтрация») по указанному названию или адресу записи (или нескольких записей) с возможностью детализации данных по иерархии.

Синтаксис

`Диалог.УстановитьФильтр (ИмяФильтра , ИмяЗаписи)`
`Диалог.УстановитьФильтр (ИмяФильтра , АдресЗаписи [,`
`Да | Нет])`
`Диалог.УстановитьФильтр (ИмяФильтра , МассивАдресов)`

Параметры

ИмяФильтра (текст) – имя фильтра, в который нужно поместить указанную запись (или несколько записей). Все возможные варианты фильтрации указаны в отчете в блоке «Фильтрация».

ИмяЗаписи (текст) – имя записи, которое нужно поместить в фильтр. Например, «МОЛ» можно установить по его названию.

АдресЗаписи (целое) – адрес записи в базе данных, которую нужно поместить в фильтр. Например, вариант вызова функции УстановитьФильтр("МОЛ", 1) установит в фильтр «МОЛ» запись из таблицы лиц с позицией «1».

Да|Нет (логическое) – если указано значение «Да» (по умолчанию), то при фильтрации будет учитываться и содержимое папки, на которую указывает фильтр. Если указано значение «Нет», то при фильтрации не будет учитываться содержимое папки. Параметр обрабатывает по аналогии с флагом «Включая содержимое папки» в окне настройки параметров отчета.

МассивАдресов (массив) – вместо адреса одной записи может передаваться массив адресов, то есть для фильтра можно указать несколько значений, по которым будут отбираться данные для построения отчета.

Возвращает

Да (логическое значение) – если удалось установить значение фильтра.

Нет (логическое значение) – если не удалось установить значение фильтра.

Примеры

Чтобы установить конкретного поставщика, по которому нужно выполнить фильтрацию данных, можно воспользоваться следующим кодом, который выполниться, например, при нажатии на кнопку в отчете:

```
оЗапись=Лицо ("7605015333", "Организации");  
оЗапись.АдресЗаписи ();  
УстановитьФильтр ("Поставщик", оЗапись.АдресЗаписи (),  
Нет);
```

Фрагмент использования функции при построении взаиморасчетов с поставщиками:

```
функция ПоказатьРасчетыЛица (пАдресЛица, лНапр,  
пДата)  
{  
    перем пОкно = Окно ("Взаиморасчеты");  
    перем пЗапись = пОкно.Запись ("Взаиморасчеты");  
    перем пПарам = пЗапись.ПараметрыВыборки ();
```



```
перем тЛ = Таблица ("Лица" );
тЛ. Загрузить (пАдресЛица) ;
перем пСчет = НайтиСчет (тЛ. СЧЕТ_РАЗДЕЛА) ;
пПарам. ОчиститьЭлементы ( ) ;
пПарам. УстановитьФильтр ("Счета" ,
пСчет. АдресЗаписи ( ) ) ;
пПарам. УстановитьФильтр ("Первые аналитики" ,
пАдресЛица) ;
Если (лНапр)
    пПарам. ВыбратьЭлемент ("Третьи аналитики" ) ;
    пПарам. ДатНач = Мин (пДата, Месяц (ТекДата ( ) ) < 7 ?
Дата (1, 7, Год (ТекДата ( ) ) -2) : Дата (1, 1,
Год (ТекДата ( ) ) -1) ) ;
    пПарам. ДатКнц = ТекДата ( ) ;
    пПарам. ФормаОтображения = "Взаиморасчеты" ;
    пПарам. СортировкаДокументов = "Датам" ;
    пПарам. Валюта = "рублях" ;
    пПарам. ТипыСчетов = "+Взаиморасчетные" ;
    пПарам. СворачиватьДвойные = "-Сворачивать двойные
обороты" ;
    пПарам. ОбновитьПараметры ( ) ;
    пЗапись. УстановитьПараметрыВыборки (пПарам) ;
    пОкно. Выполнить ( ) ;
}
```