



**ВСТРОЕННЫЙ ЯЗЫК
СИСТЕМЫ «1С:
ПРЕДПРИЯТИЕ»**

Встроенный язык является важной частью технологической платформы «1С:Предприятия 8», поскольку позволяет разработчику описывать собственные алгоритмы функционирования прикладного решения.

Встроенный язык имеет много общих черт с другими языками, такими как Pascal, JavaScript, Basic, что облегчает его освоение начинающими разработчиками. Однако он не является прямым аналогом какого-либо из перечисленных языков.

Наиболее значимые особенности встроенного языка:

- предварительная компиляция — перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код;
- кэширование скомпилированных модулей в памяти;
- мягкая типизация — тип переменной определяется типом значения, которое она содержит, и может изменяться в процессе работы;
- отсутствие программного описания объектов конфигурации — разработчик может использовать либо встроенные в платформу объекты, либо объекты, созданные системой в результате визуального конструирования прикладного решения.

Событийная ориентированность встроенного языка

Назначение встроенного языка в системе 1С:
Предприятие определяется идеологией создания
прикладных решений. Прикладные решения в 1С:
Предприятия 8 не кодируются целиком. Большая часть
прикладного решения создается разработчиком путем
визуального конструирования — создания новых
объектов конфигурации, задания их свойств, форм
представления, взаимосвязей и пр.

Встроенный язык используется лишь для того, чтобы
определить поведение объектов прикладного решения,
отличное от типового, и создать собственные алгоритмы
обработки данных.

По этой причине модули, содержащие текст
на встроенном языке, используются системой
в конкретных, заранее известных ситуациях, которые
могут возникнуть в процессе работы прикладного
решения. Такие ситуации называются **событиями**.
События могут быть связаны с функционированием
объектов прикладного решения или с самим прикладным
решением, как таковым.

Например,
с функционированием
объекта прикладного
решения **Справочник**
связан ряд событий, среди
которых есть событие
ПередЗаписью.

Это событие возникает
непосредственно перед тем,
как данные элемента
справочника должны быть
записаны в базу данных.

Синтакс-помощник

Содержание Индекс Поиск

- Общее описание встроенного языка
- Глобальный контекст
- Общие объекты
- Универсальные коллекции значений
- Интерфейс (управляемый)
- Интерфейс (обычный)
- Прикладные объекты**
- Константы
- Критерии отбора
- Справочники**
- СправочникИМенеджер
- СправочникИМенеджер.<Имя справочника>
- СправочникСсылка.<Имя справочника>
- СправочникОбъект.<Имя справочника>**
- Свойства
- Методы
- События**
- ОбработкаЗаполнения
- ОбработкаПроверкиЗаполнения
- ПередЗаписью**
- ПередУдалением
- ПриЗаписи
- ПриКопировании

СправочникОбъект.<Имя справочника>
(CatalogObject.<Имя справочника>)
ПередЗаписью (BeforeWrite)

Синтаксис:
ПередЗаписью(<Отказ>)

Параметры:
<Отказ>

Тип: Булево. Признак отказа от записи элемента. Если в теле процедуры-обработчика установить данному параметру значение Истина, запись элемента выполнена не будет. Значение по умолчанию: Ложь

Описание:
Возникает перед выполнением записи элемента справочника. Процедура-обработчик вызывается после начала транзакции записи, но до начала записи элемента справочника.

Разработчик, используя встроенный язык, может описать алгоритм, который, например, будет проверять корректность данных, введенных пользователем. Разместив этот алгоритм в соответствующем модуле, разработчик обеспечит то, что каждый раз, как пользователь будет выполнять запись элемента справочника, система будет выполнять созданный разработчиком алгоритм и проверять, не забыл ли пользователь заполнить обязательные реквизиты справочника.

Таким образом можно сказать, что встроенный язык является *скриптовым* языком для программирования *бизнес-логики*, а использование модулей на встроенном языке является событийно-зависимым, т. е. выполнение модулей происходит при возникновении определенных событий в процессе функционирования прикладного решения.

Переменные

Для объявления переменных используется ключевое слово "Перем". Существует также неявное объявление переменных при первом присвоении значения переменной.

```
Перем Значение1;  
Значение2 = 4;
```

Переменная *Значение1* объявлена явно, а переменная *Значение2* объявлена неявно. При неявном объявлении переменной система определяет ее тип по присваиваемому ей значению. При повторном присвоении переменной значения ее тип может измениться.

В именах переменных допускается использование символов и цифр, а также символа подчеркивания. При этом имя переменной должно начинаться только с символа. Использование пробелов в именах переменных не допускается.

Операторы

Операторы имеют вид стандартного обращения к процедуре, за исключением оператора присваивания ($A = B;$) и синтаксических конструкций встроеного языка (например, таких как Для, Пока, Если).

Между собой операторы обязательно следует разделять символом ";" (точка с запятой).

Конец строки не является признаком конца оператора, т.е. операторы могут свободно переходить через строки и продолжаться на другой строке. Можно располагать произвольное число операторов в одной строке, разделяя их символом ";".

Операторы языка в программном модуле можно подразделить на две категории:

операторы объявления переменных
и исполняемые операторы.

Операторы объявления переменных создают имена переменных, которыми манипулируют исполняемые операторы.

Одним из операторов являются циклы. Во встроенном языке различают следующие виды циклов:

Пока <условие> цикл.

Для каждого <имя переменной> из <имя коллекции> цикл.

Для <имя переменной> = <начальное значение> по <конечное значение> цикл.

Для досрочного прекращения цикла используется оператор "Прервать".

Пример использования цикла "для каждого":

```
// Создание массива
Массив = Новый Массив(10);

// Заполнение массива
Для Сч=0 по 9 Цикл
    Массив[Сч]=Сч;
КонецЦикла;

// Индикация массива
Для Каждого Элемент из Массив Цикл
    Сообщить(Элемент);
КонецЦикла;
```

Рассмотрим следующий вид операторов - "условия". Условия оформляются следующим образом:

```
Если <ключевое слово > (<логическое выражение>) <ключевое слово> (<логическое выражение>) :  
:  
ИначеЕсли <ключевое слово> (<логическое выражение>) <ключевое слово> (<логическое выражение>) :  
:  
Иначе  
КонецЕсли;
```

Встроенные функции

К встроенным функциям языка относятся функции работы со значениями типа дата, строка, число, а также функции преобразования значений и функции форматирования

