

Типы данных и операторы

Переменные

Динамическая типизация.

```
a = -1;
a = "Один";
Сообщить(a);
// "Один"
```

ТипЗнч(элмас) - вывести тип переменной.

Можно явно определить имя переменной, ключевое слово Экспорт позволяет обращаться к переменной через контекст модуля. Без экспорта только для эстетики кода.

```
Перем <Имя переменной 1> [Экспорт]
```

Типы данных и преобразования типов

Примитивные типы данных

Тип	Описание
Null	Исключительно для определения отсутствующего значения при работе с базой данных.
Неопределено (Undefined)	<div>Пустое значение, не принадлежащее ни к одному другому типу.</div> <div>a = Неопределено; б = ?(а = Неопределено, 0, 1);//б = 0</div> <div>Прямое преобразование в булево: нельзя При сравнении любой тип не равен Неопределено</div> <div>a = 0; б = ?(а = Неопределено, 0, 1);//б = 1</div>

Тип	Описание
Число	Определены основные арифметические операции. Максимальная разрядность 38 знаков. Разделитель точка. 32 знака. Прямое преобразование в булево: любое ненулевое Истина, 0 Ложь Преобразование в строку: Строка()
Строка	Формат Unicode произвольной длины. В двойных кавычках. Многостроковый режим через Прямое преобразование в булево: нельзя <div>a = "";</div> <div>b = ?(a, 0, 1); //вызовет ошибку</div> <div>Преобразование в число:</div> <div>Ч1 = Число(СтрокаЧ1);//вызовет исключение при невозможности</div> <div>Преобразование в дату:</div> <div>Дата("20211231123456"); //YYYYMMDDHHMMSS</div> <div>Дата("2021", "12", "24", "12", "34", "56");</div>
Дата	Строка цифр, заключенная в одинарные кавычки вида: 'ГГГГММДдччммсс' Прямое преобразование в булево: нельзя Контроль заполненности: Перем.ЗначениеЗаполнено() Пустая дата - Дата(1,1,1)
Булево	Значения данного типа имеют два значения Истина и Ложь
Тип	Используются для идентификации типов значений. Это необходимо для определения и сравнения типов.

Универсальные коллекции значений

Тип	Описание
-----	----------

<p>Массив</p>	<p>Аналог списка. Конструктор:</p> <div data-bbox="815 147 1485 409"> <pre> м1 = Новый Массив; //массив нулевой длины м2 = Новый Массив(<ФиксированныйМассив>); //создание массива из фиксированного массива м3 = Новый Массив(10, 3); //мас. из 10 эл-тов, каждый из которых - мас. из 3 эл-тов </pre> </div> <p>Методы:</p> <p>ВГраница - индекс верхней границы</p> <p>Вставить(Индекс, Значение) - Добавляет Значение в указанный индекс со смещением. Если указать больше границы, то будут созданы несуществующие значения с типом Неопределено и последним - Значение.</p> <p>Добавить - Добавляет элемент в конец массива.</p> <p>Количество</p> <p>Найти - Если найден - индекс, иначе неопределено.</p> <p>Очистить - Удаляет все значения из массива.</p> <p>Удалить - удаляет указанный индекс</p>
<p>Структура</p>	<p>Аналог словаря. Ключи только строковые.</p> <p>Конструктор:</p> <div data-bbox="815 925 1485 1088"> <pre> ст1 = Новый Структура; //пустая ст1 = Новый Структура("к1, к2", 1, 2) //структура с ключами к1, к2 и знач. 1 и 2 </pre> </div> <p>Доступ к элементу:</p> <div data-bbox="815 1167 1485 1283"> <pre> а["к1"]; а.к1; </pre> </div> <p>Перебор значений:</p> <div data-bbox="815 1361 1485 1624"> <pre> а = Новый Структура("к1, к2", 1, 2); Для каждого элмас из а Цикл Сообщить(элмас.Ключ + " " + элмас.Значение); КонецЦикла; </pre> </div> <p>Вставить(ключ, значение)</p> <p>Количество()</p> <p>Очистить() - удаляет все элементы</p> <p>Свойство(Ключ, [НайденноеЗначение]) - безопасный поиск по ключу</p> <p>Удалить(Ключ)</p>
<p>Соответствие</p>	<p>Как структура, только ключ может быть любого типа.</p>

Список значений	Как массив + строковое описание (Представление), Пометка и Картинка
-----------------	--

Таблица значений

Двумерная таблица.

```
тз = Новый ТаблицаЗначений;//Создание
таблицы
тз.Колонки.Добавить("Наименование", Новый
ОписаниеТипов("Строка"));
//Имя — идентификатор колонки
//Заголовок — представление колонки в
диалогах
//ТипЗначения — тип, может быть
произвольного типа;
//Ширина — ширина колонки в диалогах;
```

```
найдКолонка =
тз.Колонки.Найти("Наименование");//поиск
колонки
Если найдКолонка = Неопределено Тогда
    Сообщить("Колонка не найдена!");
КонецЕсли;
```

```
//перебор колонок
Для каждого Колонка Из тз.Колонки Цикл
    Сообщить(Колонка.Имя);
КонецЦикла;
```

```
Вставить() — Вставляет новую колонку в
указанную позицию коллекции
Добавить() — Добавляет новую колонку в конец
коллекции
Количество() — Возвращает количество колонок в
коллекции
Найти() — Ищет колонку в коллекции по имени
Очистить() — Удаляет все колонки из коллекции
Сдвинуть() — Сдвигает колонку влево или вправо
Удалить() — Удаляет колонку из коллекции
```

```
СтрокаТЧ = тз.Добавить();
СтрокаТЧ.Наименование = "Стул деревянный";
тз = СтрокаТЧ.Владелец();
тз.Удалить(тз.Индекс(СтрокаТЧ));
```

```
Для Каждого СтрокаТЧ Из тз Цикл
    ИндСтроки = тз.Индекс(СтрокаТЧ);
КонецЦикла;
```

Фиксированный (массив, соответствие, структура)	Аналогичный объект только для чтения.
Хранилище значений	Недоступен напрямую в управляемой форме. Нужно городить временное хранилище.

Объекты

Респект и уважуха автору AlexO за разъяснение сущности взаимодействия в 1С [ссылка на форум](#)

В 1С нет настоящих "объектов" ООП. Эти товарищи обозвали словом "объект" ссылку на словарь значений. В 1С введено внутреннее понятие КоллекцияЗначений - и это не аналог контейнера объектов из ООП (который и сам определяет поведение входящих объектов, и дает доступ напрямую к ним - к их свойствам, методам, данным, событиям и т.д.), а набор ссылок на другие объекты, и из коллекции, если не получен "вложенный" элемент-объект (например, через метод НабораЗаписей НаборЗаписей.Прочитать()), нельзя напрямую получить свойства и методы элементов коллекции, а только - получить "объекты" коллекции, и уже обходом или обращением к элементу коллекции - работать со свойствами и методами "вложенных" объектов. Объект РегистрСведений не содержит объект РегистрСведенийНаборЗаписей, а НаборЗаписей не содержит объекты РегистрСведенийЗапись. Для работы с каждым вложенным уровнем нужно заново получать объекты этого нового уровня вложенности. Собственно, вся канитель "не могу получить данные объекта там-то", "не могу получить доступ к процедуре тут-то", "не видна переменная экспортная такая-то" и прочие невообразимые и множественные ограничения платформы - именно из-за наборов не связанных напрямую друг с другом "объектов", которых нужно каждый раз "получать", извлекать данные, и которым нужно каждый раз указывать - что мы от них хотим.

Тип конфигурации ...Объект (например СправочникОбъект) -

Поскольку переписывать лень, далее используется слово Объект в понимании 1С.

Внешних модулей нет. Использование процедур в разных проектах - только копирование/вставка. Придется смириться.

Обращение к свойствам через точку или <Объект>["имя свойства"]

```
Менеджер = Справочники["Менеджеры"];
```

```
Менеджер = Справочники.Менеджеры;
```

Обращение к методам - через точку. Дополнение:

Категорический запрет на использование в запросах "двойного разыменования"
(Объект.Свойство<содержащее СсылкуНаДругойОбъект>.СвойствоДругогоОбъекта) -
только явное соединение таблиц через СОЕДИНЕНИЕ...

Условные операторы

?(<Логическое выражение>, <Выражение 1>, <Выражение 2>)

```
а = Истина;
```

```
б = ?(а = Ложь, 0, 1); // б = 1
```

Если <Логическое выражение> Тогда

// Операторы

[ИначеЕсли <Логическое выражение> Тогда]

// Операторы

[Иначе]

// Операторы

КонецЕсли;

Разное

```
Выполнить(<Строка>);
```

```
Выполнить("Сообщить(а)");
```

Исполняет строковое представление команды.

Revision #15

Created 2 January 2025 14:35:01 by Admin

Updated 10 February 2025 09:52:22 by Admin