

#T2

Основы программирования. Базовые операторы

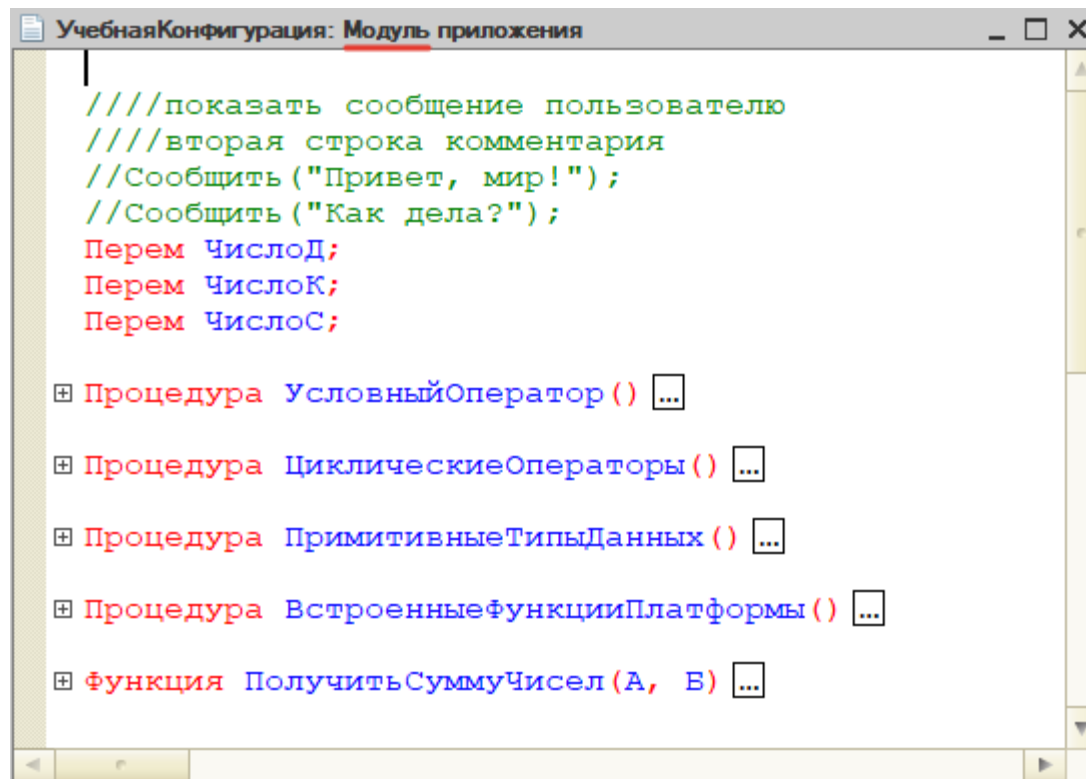
Программист 1С: с нуля до профи



Базовые ПОНЯТИЯ

Программный модуль

Программный модуль - Контейнер программного кода

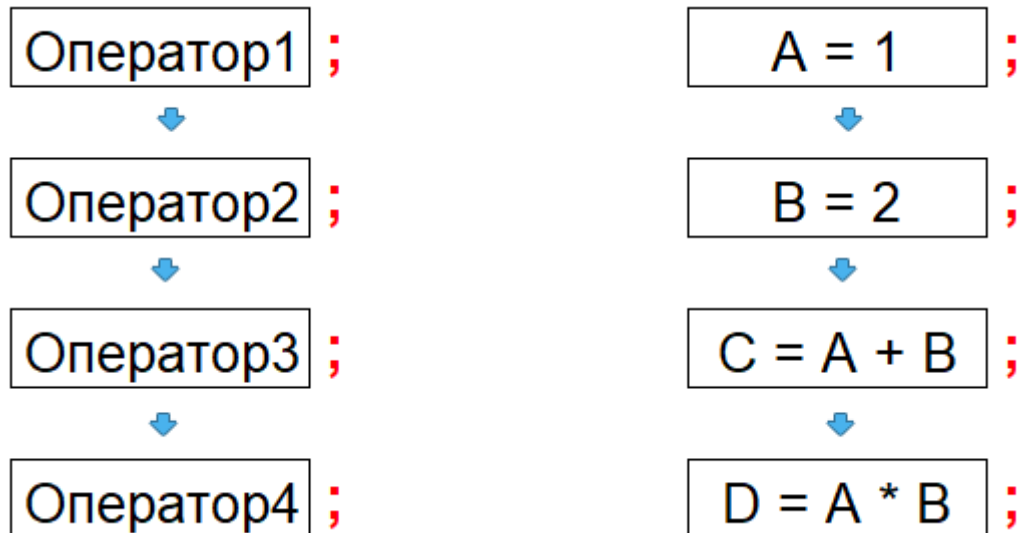


```
УчебнаяКонфигурация: Модуль приложения
|
|
|   ////показать сообщение пользователю
|   ////вторая строка комментария
|   //Сообщить ("Привет, мир!");
|   //Сообщить ("Как дела?");
|   Перем ЧислоД;
|   Перем ЧислоК;
|   Перем ЧислоС;
|
|   ▣ Процедура УсловныйОператор () ...
|
|   ▣ Процедура ЦиклическиеОператоры () ...
|
|   ▣ Процедура ПрimitivesкиеТипыДанных () ...
|
|   ▣ Процедура ВстроенныеФункцииПлатформы () ...
|
|   ▣ функция ПолучитьСуммуЧисел (А, В) ...
```

Инструкция (оператор)

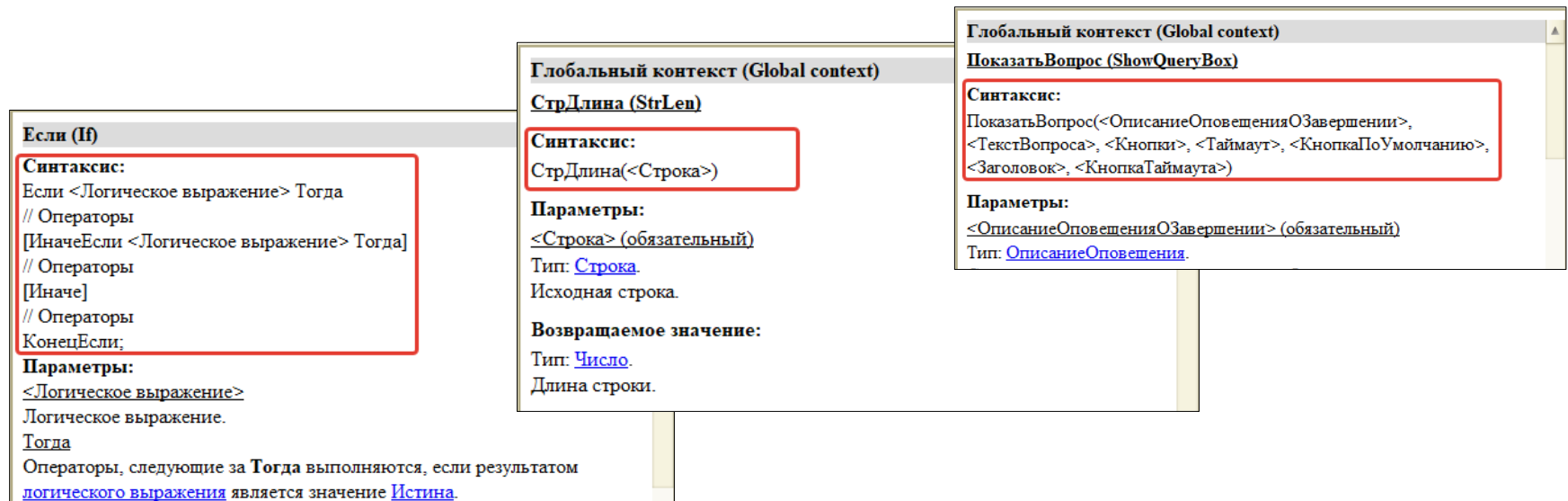
- **Инструкция** или **оператор** – наименьшая автономная часть языка программирования. Программный код состоит из операторов.

Разделитель операторов - ;



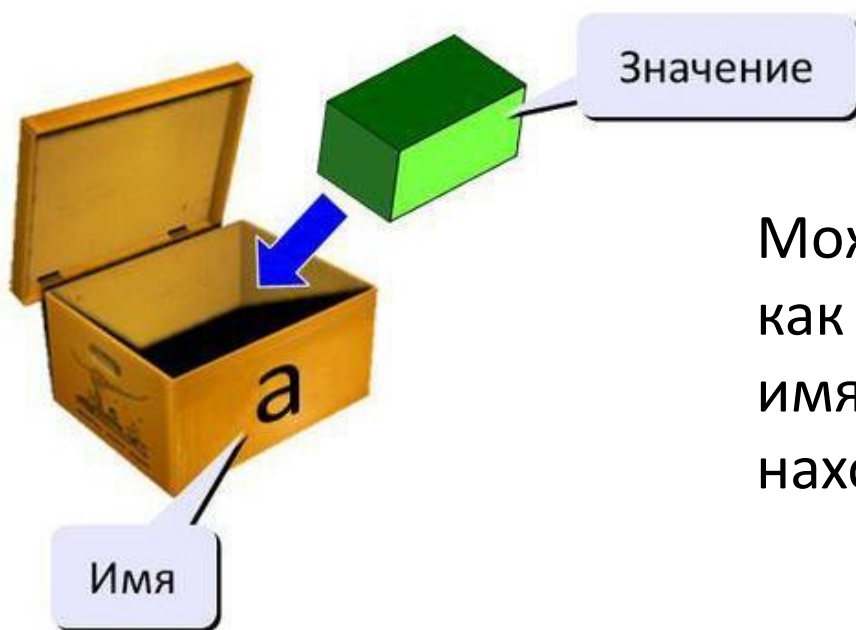
Синтаксис

- **Синтаксис** – правила описания конструкций (операторов, функций и т.д.) во встроенном языке программирования.



Переменная

- **Переменная** – именованный контейнер для хранения данных.



Можно представить переменную как ящик, на котором написано имя переменной, а внутри находится значение.

Как правильно называть переменные

Главное правило:

Имя переменной

- Должно начинаться либо с буквы либо с символа «_» (нижнее подчеркивание)
- Не должно содержать пробелов и специальных СИМВОЛОВ

Неправильно:

- 1Сентября
- Статья расходов
- Гл.Бухгалтер

Правильно:

- ПервоеСентября
- СтатьяРасходов
- ГлБухгалтер



Как правильно называть переменные

Дополнительные рекомендации:

- Первая буква имени переменной должна быть большой
- Если имя переменной состоит из нескольких слов, то каждое слово лучше начинать с большой буквы

Некрасиво:

- валютаучета
- подразделение
- физическоелицо

Красиво:

- ВалютаУчета
- Подразделение
- ФизическоеЛицо

Способы объявления переменных

Как описать переменную в модуле

Вариант 1 (явный способ):

Перем `ИмяПеременной` ;

Пример

Перем `КодВалюты` ;

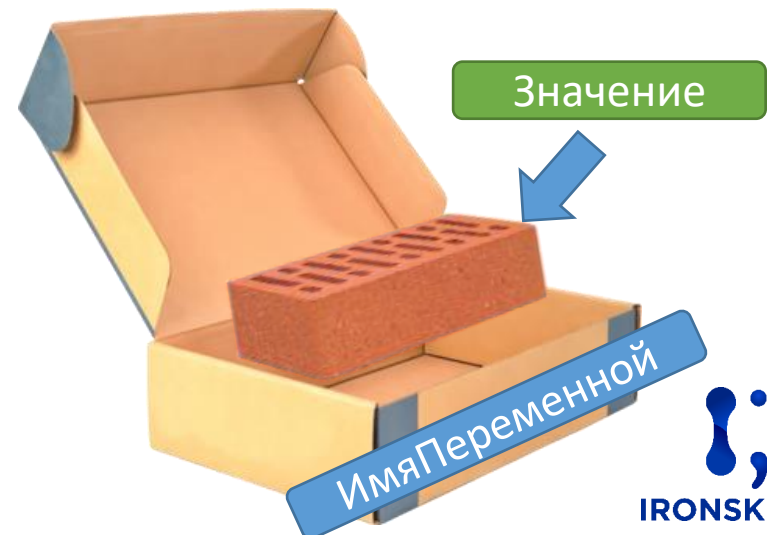


Вариант 2 (неявный способ):

`ИмяПеременной` = `Значение` ;

Пример

`КодВалюты` = `933` ;



Оператор присваивания

↓

<Назначение> = <Источник>;

Как читать:

Присвоить значение <Источник> переменной, обозначенной как <Назначение>

Примеры:

```
//1. присвоить значение способа отгрузки "Самовывоз" переменной СпособыОтгрузки  
СпособыОтгрузки = Перечисления.СпособыОтгрузки.Самовывоз;
```

```
//2. присвоить значение даты "10.09.2021" переменной "ДатаДоставки"  
ДатаДоставки = '20210910';
```

```
//3. присвоить значение 5 переменной "РасстояниеДоПункта"  
РасстояниеДоПункта = 5;
```



Комментарии в модуле

```
// Формирует новую таблицу контактной информации.  
//  
// Параметры:  
// КолонкаОбъект - Булево - Если Истина, то таблица будет содержать колонку "Объект".  
// Это нужно, если в таблице необходимо хранить контактной информацию для нескольких объектов.  
//  
// Возвращаемое значение:  
// ТаблицаЗначений - с колонками:  
// * Объект - ЛюбаяСсылка - владелец контактной информации.  
// * Вид - СправочникСсылка.ВидыКонтактнойИнформации - вид контактной информации.  
// * Тип - ПеречислениеСсылка.ТипыКонтактнойИнформации - тип контактной информации.  
// * Значение - Строка - JSON, соответствующий структуре контактной информации.  
// * ЗначенияПолей - Строка - XML, соответствующий XDTO пакету КонтактнаяИнформация или Адрес.  
// * Представление - Строка - представление контактной информации.  
// * Дата - Дата - дата, с которой действует запись контактной информации.  
// * ИдентификаторСтрокиТабличнойЧасти - Число - идентификатор строки этой табличной части  
//
```

Функция НоваяКонтактнаяИнформация(КолонкаОбъект = Истина) Экспорт

КонтактнаяИнформация = Новый ТаблицаЗначений;

ОписаниеТиповСтрока1500 = Новый ОписаниеТипов("Строка", , Новый КвалификаторыСтроки(1500));

Если КолонкаОбъект Тогда

КонтактнаяИнформация.Колонки.Добавить("Объект");

КонецЕсли;

КонтактнаяИнформация.Колонки.Добавить("Представление",

КонтактнаяИнформация.Колонки.Добавить("ЗначенияПолей",

КонтактнаяИнформация.Колонки.Добавить("Значение",

КонтактнаяИнформация.Колонки.Добавить("Вид",

КонтактнаяИнформация.Колонки.Добавить("Тип",

КонтактнаяИнформация.Колонки.Добавить("Дата",

КонтактнаяИнформация.Колонки.Добавить("ИдентификаторСтрокиТабличнойЧасти", Новый ОписаниеТипов("Число"));

ОписаниеТиповСтрока1500);

Новый ОписаниеТипов("Строка"));

Новый ОписаниеТипов("Строка"));

Новый ОписаниеТипов("СправочникСсылка.ВидыКонт

Новый ОписаниеТипов("ПеречислениеСсылка.ТипыКо

Новый ОписаниеТипов("Дата"));

Новый ОписаниеТипов("Число"));

Возврат КонтактнаяИнформация;

КонецФункции

Тип данных.

Примитивные типы

Тип данных

- **Тип** – это множество значений и операций над ЭТИМИ значениями.

Типизация в 1С **неявная** – тип переменной определяется значением, которое ей присвоено.

$$\boxed{2} + \boxed{2} = \boxed{}$$

$$\boxed{2} + \boxed{2} = \boxed{}$$

Примитивные типы данных

- Число
- Строка
- Дата
- Булево
- Неопределено
- Тип
- NULL

Правила преобразования ТИПОВ

2 + **"2"** =
число строка

"2" + **2** =
строка число

2 + **"A"** =
число строка

'20190903' + **10** =
дата число

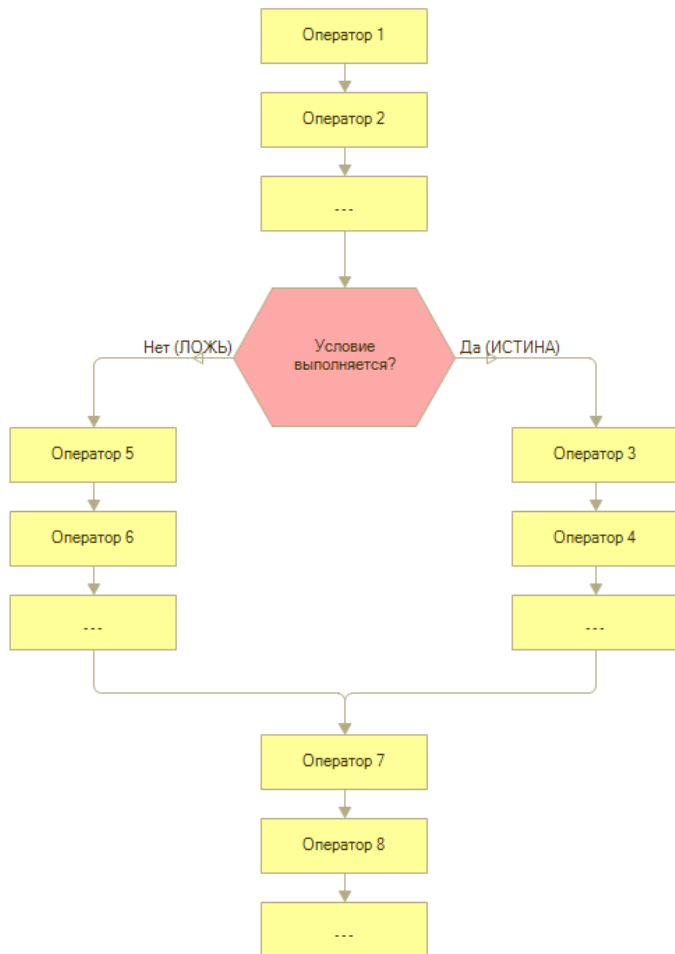
Условный оператор

Условный оператор

- **Предназначение** – организовать ветвление в программном коде.

Другими словами, при выполнении определенного условия перейти к выполнению одних операторов, а при невыполнении условия перейти к выполнению других операторов.

Условный оператор



Синтаксис

Если <Условие> Тогда

...

Иначе

...

КонецЕсли;

Краткая форма

? (<Условие>, <ВыражениеЕслиУсловиеИСТИНА>, <ВыражениеЕслиУсловиеЛожь>)

Операции сравнения

=, >, >=, <, <=, <>

Логические операции

И, ИЛИ, НЕ, ()

Логическая операция « \Rightarrow »

Какое сообщение увидит пользователь на экране в результате выполнения кода?

```
ЧислоА = 7;  
Числов = 5;  
ЧислоА = Числов = 5;  
Сообщить (ЧислоА);
```

Условный оператор. Логические операции

Логическая операция "И"

Условие1		Условие2	Результат
Истина	И	Истина	Истина
Истина	И	Ложь	Ложь
Ложь	И	Истина	Ложь
Ложь	И	Ложь	Ложь

Логическая операция "ИЛИ"

Условие1		Условие2	Результат
Истина	ИЛИ	Истина	Истина
Истина	ИЛИ	Ложь	Истина
Ложь	ИЛИ	Истина	Истина
Ложь	ИЛИ	Ложь	Ложь

Логическая операция "НЕ"

	Условие	Результат
НЕ	Истина	Ложь
НЕ	Ложь	Истина

Приоритеты логических операций

Логическая операция	Приоритет
()	1
НЕ	2
И	3
или	4

Циклический оператор

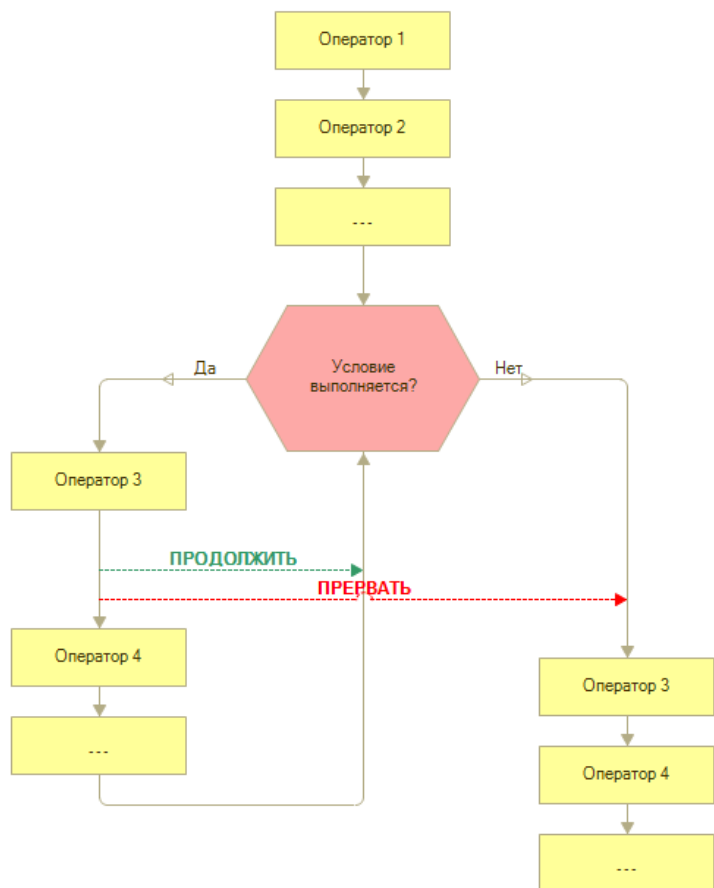
Циклический оператор (цикл)

- **Предназначение** – организовать «защивание» в программном коде.

Другими словами - организовать многократное выполнение определенных операторов.

- **Итерация цикла** – однократное выполнение операторов внутри цикла.

Циклический оператор



Цикл по условию

Пока <Условие> Цикл
...
КонецЦикла;

Цикл по счетчику

Для Счетчик=<НачЗначение> По <КонЗначение> Цикл
...
КонецЦикла;

Цикл по коллекции

Для каждого ЭлементКоллекции Из <Коллекция> Цикл
...
КонецЦикла;

Циклический оператор (цикл). Виды циклов

- **Цикл по условию**

Выполнять операторы до тех пор, пока условие – Истина. Условие может быть произвольным

Пока <Условие> Цикл

...

КонецЦикла;

- **Цикл по счетчику**

Выполнять операторы до тех пор, пока условие – Истина.
Условие всегда одинаковое: Счетчик <= КонечноеЗначение

Для Счетчик=<НачЗначение> По <КонЗначение> Цикл

...

КонецЦикла;

- **Цикл по коллекции**

Выполнять операторы до тех пор, пока в коллекции есть следующий элемент.

Для каждого ЭлементКоллекции Из <Коллекция> Цикл

...

КонецЦикла;

Пропуск итерации и принудительное прекращение

- Как пропустить одну итерацию цикла:
Продолжить;
- Как завершить (прервать) цикл:
Прервать;

Конец 2-го занятия

Подписывайтесь на нас в социальных сетях 😊



<https://www.youtube.com/channel/UC124tAuP11GlaGpK3YGHzw>



<https://www.facebook.com/ironskillsby/>



https://www.instagram.com/ironskills_/