

Лабораторная работа № 8

**ВСТРОЕННЫЙ ЯЗЫК 1С.
ОПЕРАТОРЫ.
УНИВЕРСАЛЬНЫЕ
КОЛЛЕКЦИИ ЗНАЧЕНИЙ.**



1. Условный оператор в языке 1С

В данном разделе мы познакомились со сложными логическими выражениями, которые образуются из простых при помощи логических операций И, Или, Не.

Мы разберемся, зачем эти логические выражения вообще нужны и научимся применять их на практике. А понадобятся они нам при изучении новой команды языка программирования 1С, которая называется Условная команда.

Простая условная команда:

Если коротко, то правила и суть этой команды можно представить так:

Если(логическое выражение)Тогда

Различные команды;

КонецЕсли

Более подробно:

1. Частью условной команды являются следующие три слова:

Если, Тогда и КонецЕсли.

2. Между словами Если и Тогда всегда находится некоторое логическое выражение, которое принимает значение Истина или Ложь.

3. Между словами Тогда и КонецЕсли может находиться любое количество команд компьютеру (например, Сообщить, ВвестиЧисло, ОткрытьЗначение и другие).

4. И, наконец, главное: команды компьютеру, заключенные между словами Тогда и КонецЕсли выполняются только в том случае, если логическое выражение, заключенное между Если и Тогда принимает значение Истина.



Расширенная форма условной команды

Если(логическое выражение)**Тогда**

Различные команды;

// Эти команды выполняются,

// если (логическое выражение)=Истина

Иначе

Различные команды;

// Эти команды выполняются,

// если (логическое выражение)=Ложь

КонецЕсли

Пример сложной конструкции "Если" в 1с

Если А>Б **Тогда**

// блок операторов

ИначеЕсли Б>С **Тогда**

// блок операторов

Иначе

// блок операторов

КонецЕсли;

Конструкция тернарной команды "?" в 1с

?(*<Логическое выражение>*,*<выражение 1>*,*<выражение 2>*);

Тернарная команда содержит три параметра. Первый параметр – логическое выражение. На втором и третьем местах — выражения любых типов. Выполнение данной команды начинается с вычисления логического выражения. Если результат вычисления — "Истина", то далее вычисляется выражение 1, в противном случае — выражение 2).



2. Циклы в языке 1С

Цикл — это специальная команда компьютеру, которая позволяет повторять выполнение других команд нужное количество раз.

В языке 1С бывает два вида циклов: Цикл **Для** и Цикл **Пока**.

Цикл Для

Конструкция цикла выглядит вот так:

```
Для Имя = НачальноеЧисло По КонечноеЧисло Цикл  
    Команда1();  
    Команда2();  
    ...  
КонецЦикла;
```

Команды, заключенные между словами **Цикл** и **КонецЦикла**, выполняются столько раз, сколько нужно шагов, чтобы *НачальноеЧисло* стало больше *КонечногоЧисла*.

При этом *Имя* программист придумывает сам, и с этим именем связывается текущее значение шага.

Конструкция цикла **Пока** выглядит так:

```
Пока ЛогическоеВыражение Цикл  
    Команда1();  
    Команда2();  
    ...  
КонецЦикла;
```

Команды, заключенные между словами **Цикл** и **КонецЦикла** выполняются пока *ЛогическоеВыражение* равно Истине.

В качестве логического выражения может быть любое условие. Пока это условие верно (то есть равно Истина), выполнение команд в цикле повторяется снова и снова, как только условие перестало быть верным, команды перестают выполняться.



3. Массив в языке 1С

Описание:

Предназначен для доступа к элементам массива, его методам и конструктору.

Элементы коллекции: Произвольный

Для объекта доступен обход коллекции посредством оператора *Для каждого ... Из ... Цикл*. При обходе выбираются значения элементов массива.

Возможно обращение к значению элемента посредством оператора [...]. В качестве аргумента передается индекс значения (**нумерация с 0**).

Методы:

- ВГраница() – Получает наибольший индекс элемента массива.
- Вставить(<Индекс>, <Значение>) – Вставляет значение в массив по указанному индексу.
- Добавить(<Значение>) – Добавляет элемент в конец массива.
- Количество() – Получает количество элементов в массиве.
- Найти(<Значение>) – Выполняет поиск элемента в массиве. Если элемент найден, возвращается его индекс. Если элемент не найден, возвращается Неопределено.
- Очистить() – Удаляет все значения из массива.
- Получить(<Индекс>) – Получает значение по индексу. Работает аналогично оператору [].
- Удалить(<Индекс>) – Удаляет значение из массива по указанному индексу.
- Установить(<Индекс>, <Значение>) – Устанавливает значение по индексу. Работает аналогично оператору [].

Конструкторы:

- Новый Массив(<ФиксированныйМассив>)
- На основании фиксированного массива. – Новый Массив(<КоличествоЭлементов1>, ..., <КоличествоЭлементовN>)

По количеству элементов. Создает массив из указанного количества элементов. Если задано несколько параметров, то будет создан массив, элементами которого являются массивы (и так далее в зависимости от количества параметров). Фактически конструктор позволяет создать массивы массивов, которые могут являться аналогом многомерного массива.

```
/// Как создать пустой массив (с последующим добавлением  
/// элементов)
```

```
Процедура Пример2()
```

```
    // в массиве 0 элементов
```

```
    Массив = Новый Массив;
```

```
    // добавляем последовательно три элемента
```

```
    Массив.Добавить(100); // (100)
```

```
    Массив.Добавить(300); // (100, 300)
```

```
    Массив.Добавить(500); // (100, 300, 500)
```

```
    // выводим на печать
```

```
    Для Каждого ЭлементМассива из Массив Цикл
```

```
        Сообщить(ЭлементМассива); // 100 300 500
```

```
    КонецЦикла;
```

```
КонецПроцедуры
```

```
/// Как обойти все элементы массива по индексу
```

```
Процедура Пример3()
```

```
    // инициализируем массив: (100, 300, 500)
```

```
    Массив = Новый Массив;
```

```
    Массив.Добавить(100);
```

```
    Массив.Добавить(300);
```

```
    Массив.Добавить(500);
```

```
    // пробегаемся от первого (с индексом 0) до последнего
```

```
    // элемента (с индексом Количество - 1).
```

```
    Для Индекс = 0 По Массив.Количество() - 1 Цикл
```

```
        Сообщить(Массив[Индекс]);
```

```
    КонецЦикла;
```

```
КонецПроцедуры
```

```
/// Массив может содержать элементы различных типов:  
/// числа, строки и т.д.
```

```
Процедура Пример4()
```

```
    Массив = Новый Массив(4);  
    Массив[0] = "HELP"; // ("HELP")  
    Массив[1] = "ME"; // ("HELP", "ME")  
    Массив[2] = "1С"; // ("HELP", "ME", "1С")  
    Массив[3] = Формат(2013, "ЧГ=0"); // ("HELP", "ME", "1С", 2016)
```

```
    Для Каждого ЭлементМассива из Массив Цикл  
        Сообщить(ЭлементМассива); // HELP ME 1С 2016  
    КонецЦикла;  
КонецПроцедуры
```

```
/// Вставка, удаление и очистка массива
```

```
Процедура Пример10()
```

```
    // инициализация пустого массива
```

```
    Массив = Новый Массив;
```

```
    // вставка трех элементов; каждый элемент
```

```
        // вставляется в начало
```

```
    Массив.Вставить(0, 10); // (10)
```

```
    Массив.Вставить(0, 100); // (100, 10)
```

```
    Массив.Вставить(0, 1000); // (1000, 100, 10)
```

```
    // определение последнего индекса
```

```
    Сообщить(Массив.ВГраница()); // 2
```

```
    // вывод на печать
```

```
    Для Индекс = 0 по Массив.ВГраница() Цикл
```

```
        Сообщить(Массив[Индекс]); // 1000 100 10
```

```
    КонецЦикла;
```

```
    // удаление элемента со значением 100
```

```
    // для этого сначала находим индекс элемента
```

```
    // если не находим возвращаем Неопределено
```

```
    Индекс = Массив.Найти(100); // 1
```

```
    // и удаляем элемент по найденному индексу
```

```
    Массив.Удалить(Индекс); // (100, 10)
```

```
    Сообщить(Массив.ВГраница()); // 1
```

```
    // удаление всех элементов из массива
```

```
    Массив.Очистить();
```

```
КонецПроцедуры
```

4. Структура в языке 1С

Описание:

Представляет собой коллекцию пар КлючИЗначение. При этом ключ может быть только строковым и должен удовлетворять требованиям, предъявляемым к именованию переменных встроенного языка.

К значениям структуры можно обращаться как к свойствам объекта. При этом ключ используется как имя свойства. Структура используется обычно для хранения небольшого количества значений, каждое из которых имеет некоторое имя.

Элементы коллекции: *КлючИЗначение*

Для объекта доступен обход коллекции посредством оператора *Для каждого ... Из ... Цикл*. При обходе выбираются элементы коллекции.

Свойства:

<Имя ключа>

В качестве свойств структура предоставляет элементы. Имя свойства определяется ключом, а значение свойства определяется значением элемента.

Методы:

- Вставить(<Ключ>, <Значение>) – Устанавливает значение элемента структуры по ключу. Если элемент с переданным значением ключа существует, то его значение заменяется, в противном случае добавляется новый элемент.
- Количество() – Получает количество элементов структуры.
- Очистить() – Удаляет все элементы структуры.
- Свойство(<Ключ>, <НайденноеЗначение>) – Получает значение элемента по указанному имени, а также проверяет, имеется ли указанное свойство. Возвращает Истина — ключ найден, Ложь — в противном случае.
- Удалить(<Ключ>) – Удаляет элемент структуры с заданным ключом.

Конструкторы:

- Новый Структура(<ФиксированнаяСтруктура>) – Создаваемая структура заполняется свойствами из исходной фиксированной структуры.
- Новый Структура(<Ключ>, <Значения>) – Создает структуру с ключами, значения которых передаются в параметре конструктора. Если в первом параметре заданы ключи элементов структуры, то в следующих параметрах могут быть указаны значения этих элементов в том порядке, в котором они расположены в строке в первом параметре.

Процедура Пример()

```

/// Как создать структуру (способ 1)
ЛичныеДанные = Новый Структура;
    // пара ключ (имя переменной) - значение (произв. типа)
    ЛичныеДанные.Вставить("Фамилия", "Иванов");
    ЛичныеДанные.Вставить("Имя", "Иван");
    ЛичныеДанные.Вставить("ДатаРождения", '19800802');
    Сообщить(ЛичныеДанные.Имя + " " + ЛичныеДанные.Фамилия +
" родился " + ЛичныеДанные.ДатаРождения);
/// Как создать структуру (способ 2)
Время = Новый Структура("Часы,Минуты,Секунды", 12, 45, 33);
Сообщить("Время " + Время.Часы + ":" +
Время.Минуты + ":" + Время.Секунды);
/// Как изменить значение поля созданной структуры
/// (способ 1)
ЛичныеДанные.Вставить("Фамилия", "Сидоров");
/// Как изменить значение поля созданной структуры
/// (способ 2)
Время.Часы = 13;
/// Как узнать количество элементов структуры
Сообщить(Время.Количество());
/// Как обойти все элементы структуры
Для Каждого Элемент Из ЛичныеДанные Цикл
    Сообщить(Элемент.Ключ + " " + Элемент.Значение);
КонецЦикла;
/// Как удалить элемент структуры
ЛичныеДанные.Удалить("ДатаРождения");
/// Как очистить все элементы структуры
ЛичныеДанные.Очистить();
КонецПроцедуры

```

Задание на лабораторную работу

Выполнить 2 задания по варианту, используя объект дерева конфигурации «Обработка» и ее форму для ввода/вывода требуемой информации.


Задание 1: порядковый номер студента в списке группы (можно уточнить у преподавателя).

Задание 2: порядковый номер студента в группе +10.

Единственное ограничение: **использование модальных окон запрещено!**

Задание 1:

Вариант	Описание задачи
1	<p>Дано натуральное число. Проверить, является ли число палиндромом, т.е. одинаково записывается слева направо и справа налево. Если на конце числа 0, то считаем, что оно не является палиндромом.</p> <p>Пример:</p> <ul style="list-style-type: none"> 1221 (число) – Истина (булево или строка) 1220 (число) – Ложь (булево или строка)
2	<p>Написать функцию перевода натурального числа из десятичной системы в двоичную. Вход и выход – строки.</p> <p>Пример:</p> <ul style="list-style-type: none"> 31 – 11111 17 – 10001
3	<p>Написать функции перевода натурального числа из двоичной системы в десятичную. Вход и выход – строки</p> <p>Пример:</p> <ul style="list-style-type: none"> 11111 – 31 10001 – 17

4	<p>Написать функцию, вычисляющую N-й бит натурального числа. Сначала указывается номер бита, потом через пробел число. Самый младший бит имеет номер 0.</p> <p>Пример;</p> <ul style="list-style-type: none"> • 3 31 – 1 • 2 17 – 0
5	<p>Задано число. Необходимо вывести на экран таблицу умножения на это число. Пример:</p> <div data-bbox="543 625 1172 1151" data-label="Form"> <p>ТУВход: <input type="text" value="5,00"/> </p> <p>ТУВывод:</p> <pre> 5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45 5 * 10 = 50 </pre> </div>
6	<p>Назовем натуральное число палиндромом, если его запись читается одинаково как с начала, так и с конца (пример: 4884, 393, 1, 22). Найти все меньшие 100 натуральные числа, которые при возведении в квадрат дают палиндром.</p>
7	<p>Натуральное число называется совершенным, если оно равно сумме всех своих делителей за исключением самого себя. Например. $6 = 1 + 2 + 3$. Дано натуральное число n. Получить все совершенные числа, меньшие n.</p>
8	<p>Дано натуральное число n. Среди чисел $1, 2, \dots, n$ найти все такие, запись которых совпадает с последними цифрами записи их квадрата.</p>

	<p>Пример:</p> <ul style="list-style-type: none"> • 6 и 36 • 5 и 25
9	<p>Ввести массив, состоящий из 15-ти элементов (двузначные целые числа). Изменить разрядность цифр, образующих элементы исходного массива и, таким образом, сформировать новый массив. Например, исходный массив: 25 71 84..., новый массив: 52 17 48....</p>
10	<p>Ввести два массива действительных чисел, состоящих из 7 и 9 элементов. Сформировать третий массив из упорядоченных по убыванию значений обоих массивов.</p>
11	<p>Выполнить свёртку массива, оставив только уникальные значения. Массив содержит только натуральные числа. На входе из строки получить массив, свернуть его и на выходе отобразить в виде строки.</p> <p>Пример:</p> <ul style="list-style-type: none"> • 1, 1, 2, 3, 4, 4, 5 – 1, 2, 3, 4, 5 • 1, 2, 3, 1, 2, 3, 3, 3, 3 – 1, 2, 3
12	<p>Выполнить сортировку массива по возрастанию. Массив содержит только натуральные числа. На входе из строки получить массив, отсортировать его и на выходе отобразить в виде строки.</p> <p>Пример:</p> <ul style="list-style-type: none"> • 1, 2, 3, 4, 3, 2 – 1, 2, 2, 3, 3, 4 • 1, 2, 3, 4, 5, 6 – 1, 2, 3, 4, 5, 6
13	<p>Написать программу поиска суммы последовательности отрицательных чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.</p>
14	<p>Написать программу поиска произведения последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному отрицательному числу. Контрольный пример: 1,2,3,4,5,-2.</p>



15	В массиве хранятся цены на 10 видов мороженого. С помощью генератора случайных чисел заполнить массив целыми значениями, лежащими в произвольном диапазоне. Определить порядковый номер самого дорогого мороженого.
*	В программе генерируется случайное целое число от 0 до 100. Пользователь должен его отгадать не более чем за 10 попыток. После каждой неудачной попытки должно сообщаться больше или меньше введенное пользователем число, чем то, что загадано. Если за 10 попыток число не отгадано, то вывести загаданное число.

Содержание отчета

1. Цель работы.
2. Описание варианта задания.
3. Пошаговое описание процесса выполнения варианта задания.
4. Выводы