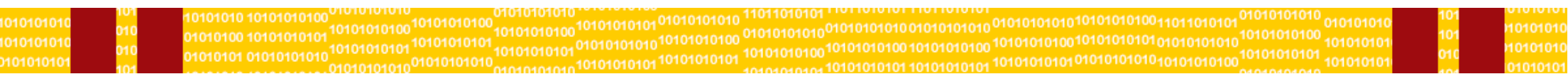


# Встроенный язык 1С

## Лекция 5

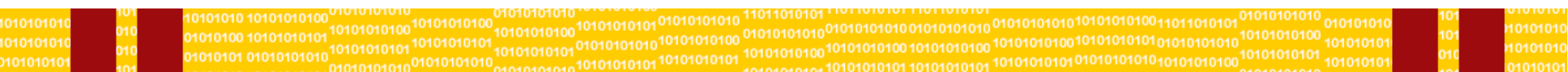
1С

# 1 Операторы



# Операторы

- Оператор присваивания
- Оператор доступа
- Условные операторы
- Операторы цикла
- Оператор приведения типа



# Оператор присваивания

- **Форма записи:**

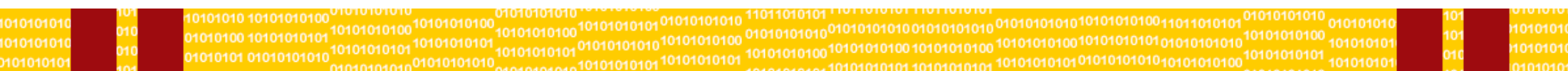
*<Имя\_переменной> = <Выражение>*

- **Пример использования:**

*Сч = 1;*

*СуммаСкидки = БазоваяСумма \* ПроцентСкидки / 100;*

*МинПериод = МИН(Дата1, Дата2);*



# Оператор доступа

- **Форма записи:**

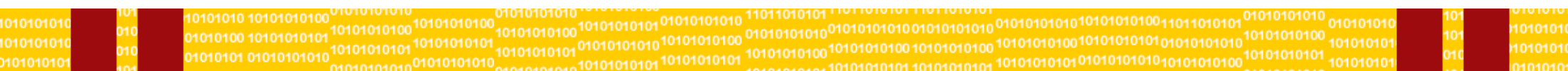
*<ИмяПеременной/ИмяОбщегоМодуля>.<ИмяАтрибута/ИмяМетода>*

- **Пример использования:**

*НазваниеПредмета = ТекПредмет.Наименование;*

*КурсВалюты = ВалютныеОперации.ПолучитьКурс(Валюта);*

*КоличествоЗаписей = Выборка.Количество();*



# Условный оператор

- **Форма записи:**

*Если <Условие> Тогда*

*<Операторы>*

*{ИначеЕсли <Условие> Тогда}*

*<Операторы>*

*{Иначе }*

*<Операторы>*

*КонецЕсли;*

*<ИмяПеременной> = ?(<Условие>;  
<Выражение1>; <Выражение2>)*

- **Пример использования:**

*Если Сумма > 1000 Тогда*

*Скидка = 10;*

*ИначеЕсли Сумма > 2000 Тогда*

*Скидка = 20;*

*Иначе*

*Скидка = 0*

*КонецЕсли;*

*Скидка = ?(Сумма > 1000; 10; 0)*

# Составные условия

**Пример 1.** Пусть у нас есть три числовые переменные *Число1*, *Число2* и *Число3*. Разделить *Число1* либо на *Число2*, либо на *Число3*. Делить на 0 **нельзя**, делитель будем выбирать исходя из этого условия.

```
Если Число2 <> 0 Тогда
    Частное = Число1 / Число2;
    Сообщить (Частное);
ИначеЕсли Число3 <> 0 Тогда
    Частное = Число1 / Число3;
    Сообщить (Частное);
Иначе
    Сообщить ("Делители равны 0, операция невозможна");
КонецЕсли;
```

# Составные условия

Если Не Число2 = 0 Тогда

//...

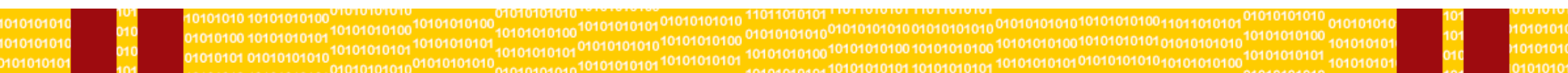
ИначеЕсли Не Число3 = 0 Тогда

//.....

Иначе

//.....

КонецЕсли;





# Условия 1С со сложными логическими выражениями

Пример 2. Разделить *Число1* на сумму переменных *Число2* и *Число3*. Чтобы не получить ошибку деления, произвести проверку на 0.

```
Если Число2 = 0 И Число3 = 0 Тогда  
    Сообщить ("Нельзя делить на 0");
```

```
Иначе
```

```
    Частное = Число1 / (Число2 + Число3);  
    Сообщить (Частное);
```

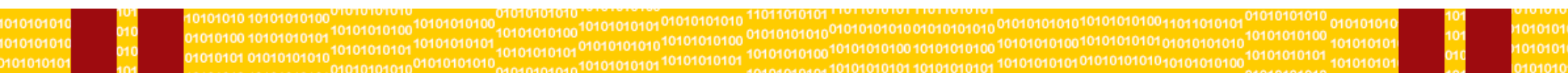
```
КонецЕсли;
```



# Условия 1С со сложными логическими выражениями

**Пример 3.** Разделить *Число1* на произведение переменных *Число2* и *Число3*. Чтобы не получить ошибку деления, произвести проверку на 0.

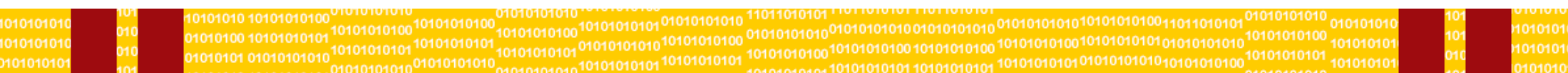
```
Если Число2 = 0 Или Число3 = 0 Тогда  
    Сообщить ("Нельзя делить на 0");  
Иначе  
  
    Частное = Число1 / (Число2 * Число3);  
    Сообщить (Частное);  
  
КонецЕсли;
```



# Циклы в 1С

В 1с существует три вида циклов

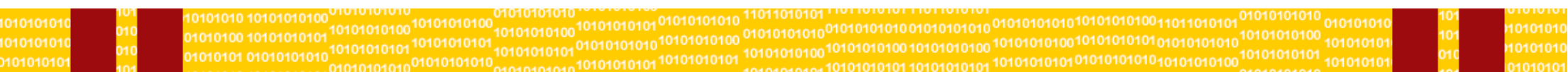
| Вид цикла   | Описание   |
|-------------|--|
| Пока        | Цикл осуществляющий повторения, пока выполняется условие.            |
| Для         | Цикл осуществляющий заданное количество повторений.                  |
| Для каждого | Цикл для обхода коллекций. Обходит каждую строку заданной коллекции. |



# Цикл Пока

Синтаксис цикла выглядит так:

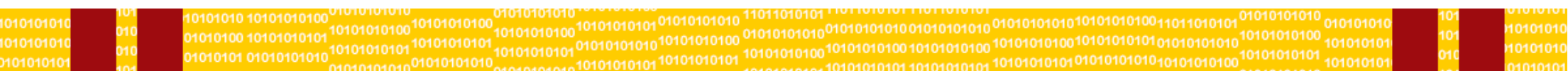
```
Пока <Логическое выражение> Цикл
    // Операторы
КонецЦикла;
```



# Цикл Пока

Пример 4. При помощи сообщения вывести пользователю цифры от 1 до 10.

```
Цифра = 1;  
Пока Цифра <= 10 Цикл  
  
    Сообщить (Цифра) ;  
    Цифра = Цифра + 1;  
  
КонецЦикла;
```



# Цикл Пока

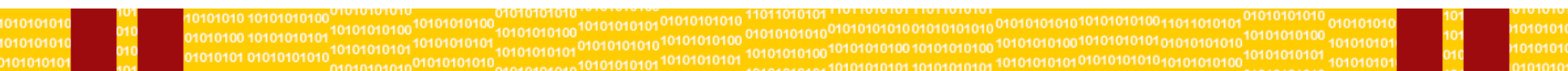
Пример 5. А теперь только не четные, в интервале от 1 до 100, в обратном порядке.

```
Цифра = 100;  
Пока Цифра >= 1 Цикл
```

```
    Если Цифра % 2 = 1 Тогда  
        Сообщить (Цифра) ;  
    КонечЕсли;
```

```
    Цифра = Цифра - 1;
```

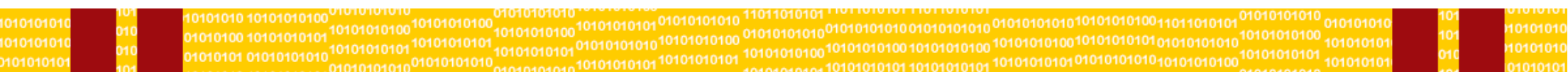
```
КонецЦикла;
```



# Цикл Для

Синтаксис цикла выглядит так:

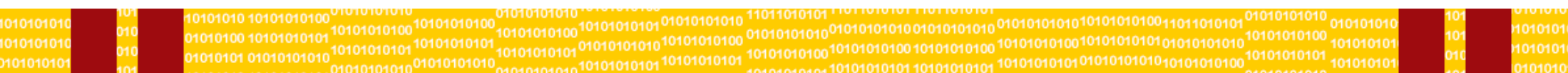
```
Для <Счетчик> = <Выражение1> По <Выражение2> Цикл  
    // Операторы  
КонецЦикла;
```



# Цикл Для

Пример 6. Вывести посимвольно строку «Привет мир!».

```
Текст = "Привет мир!";  
ДлинаСтроки = СтрДлина(Текст);  
  
Для НомерСимвола = 1 По ДлинаСтроки Цикл  
  
    Символ = Сред(Текст, НомерСимвола, 1);  
    Сообщить(Символ);  
  
КонецЦикла;
```

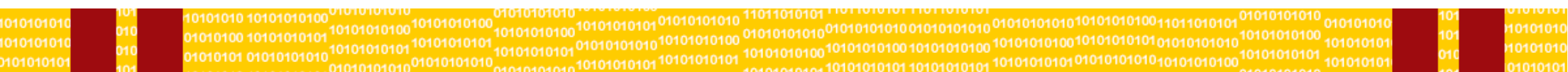




# Цикл Для Каждого

Синтаксис цикла выглядит так

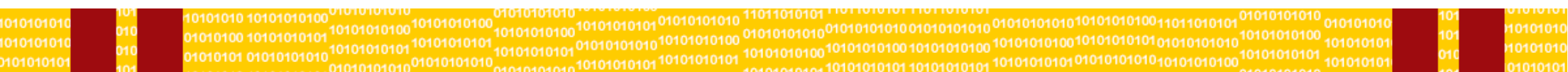
```
Для каждого <Переменная1> Из <Переменная2> Цикл  
    // Операторы  
КонецЦикла;
```



# Специальные операторы циклов

Циклы 1с могут использовать специальные операторы *Продолжить* и *Прервать*.

Они предназначены для предварительного завершения итерации и всего цикла соответственно.



# Специальные операторы циклов

Пример 7. Выведем числа от 1 до 10 циклом *Пока*.

```
Цифра = 1;
```

```
Пока Истина Цикл
```

```
    Если Цифра > 10 Тогда
```

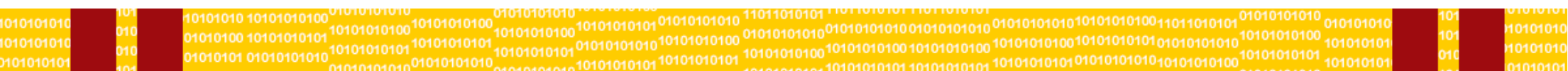
```
        Прервать;
```

```
    КонечЕсли;
```

```
        Сообщить (Цифра) ;
```

```
        Цифра = Цифра + 1;
```

```
    КонечЦикла;
```



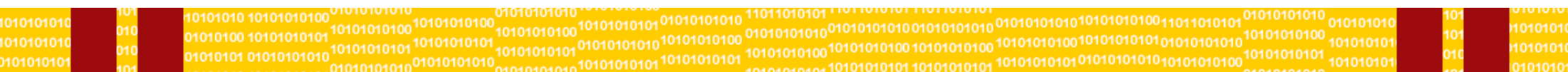
# Оператор приведения типа

- **Форма записи:**

*<ИмяТипа>(<Выражение>)*

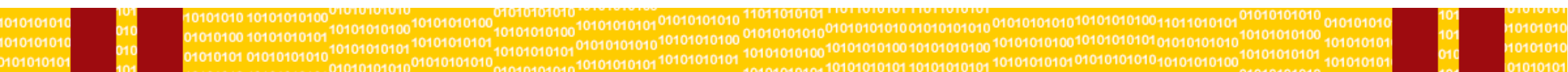
- **Пример использования:**

*НачДата = Дата(«05.03.2016»);*



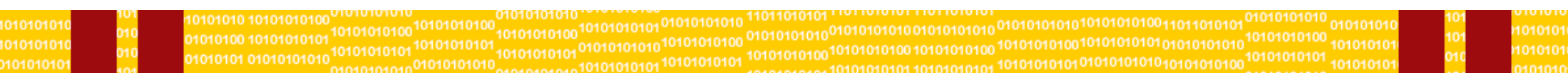
2

# Универсальные коллекции значений. Простые коллекции.



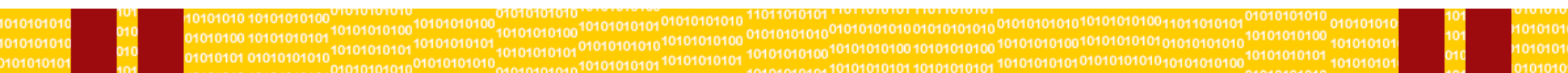
# Общее для коллекций

- Коллекция может создаваться в результате работы какой-либо функции (функция возвращает в качестве значения универсальную коллекцию).
- Почти любую универсальную коллекцию можно создать с помощью конструктора (исключением являются табличные части, которые выступают в качестве объектов конфигурации).
- Для универсальных коллекций существуют такие общие понятия, как **индекс** и **номер**. Каждый элемент коллекции имеет **индекс**. При этом **индекс** начинается с нуля.



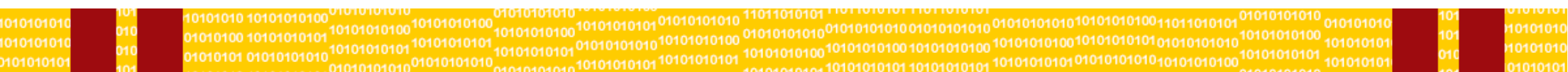
# Общее для коллекций

- Для всех коллекций используется обход элементов коллекции. Обход возможен двумя способами:
  - циклом *Для*
  - циклом *Для каждого из*.
- Для большинства универсальных коллекций применимы методы: *Количество, Индекс, Добавить, Вставить, Удалить и Найти*.



# Массив

- Хранит последовательность элементов любого типа прикладного решения.
- Доступ к элементам по индексу и через итератор  
*Для каждого.*
- Все элементы упорядочены по индексу.



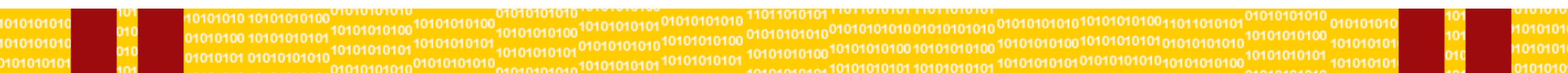


# Работа с массивами 1С

- Для добавления нового элемента в массив используется метод *Добавить(<Значение>)*.

```
НашМассив = Новый Массив(); //массив без фиксированного кол-ва значений  
ГСЧ = Новый ГенераторСлучайныхЧисел(); //будем заполнять случайными числами
```

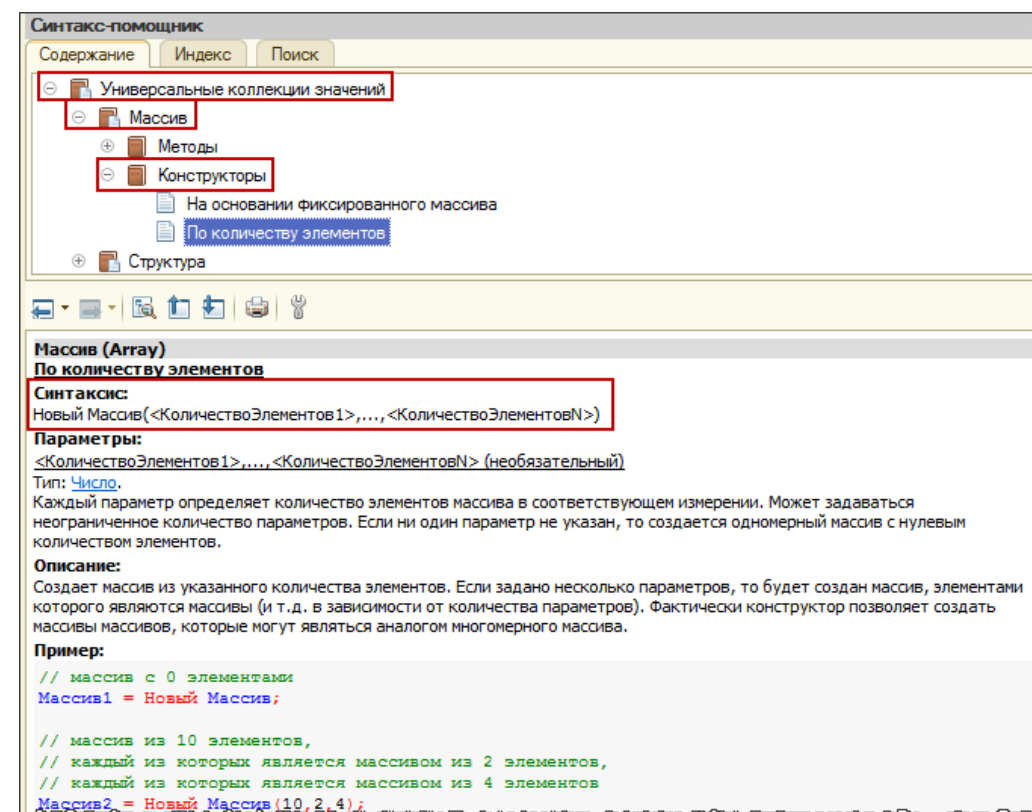
```
НашМассив.Добавить ( ГСЧ.СлучайноеЧисло (0, 1000) );  
НашМассив.Добавить ( ГСЧ.СлучайноеЧисло (0, 1000) );
```



# Работа с массивами 1С

```
НашМассив = новый Массив();
```

```
Для Счетчик = 0 по 99 Цикл  
    НашМассив.Добавить(Счетчик);  
    Сообщить(НашМассив[Счетчик]);  
КонецЦикла;
```



# Работа с массивами 1С

**Пример 9.** Необходимо ввести 5 чисел, определить сумму и вывести результат. *(без массива)*

```
Число1=0;  
Число2=0;  
Число3=0;  
Число4=0;  
Число5=0;  
ВвестиЧисло (Число1) ;  
ВвестиЧисло (Число2) ;  
ВвестиЧисло (Число3) ;  
ВвестиЧисло (Число4) ;  
ВвестиЧисло (Число5) ;  
Сумма5 = Число1 + Число2 + Число3 + Число4 + Число5;  
Сообщить (Сумма5) ;
```

# Работа с массивами 1С

**Пример 9.** Необходимо ввести 5 чисел, определить сумму и вывести результат. (с массивом)

```
Числа = Новый Массив(5);  
Сумма5 = 0;
```

```
Для Шаг=0 По 4 Цикл  
    ВвестиЧисло(Числа[Шаг]);  
    Сумма5 = Сумма5 + Числа[Шаг];
```

```
КонецЦикла;  
Сообщить(Сумма5);
```

# Работа с массивами 1С

**Пример 10.** Получить сумму элементов числового массива

```
НашМассив = Новый Массив;  
НашМассив.Добавить (100);  
НашМассив.Добавить (-5);  
НашМассив.Добавить (3.4);  
НашМассив.Добавить (0);  
НашМассив.Добавить (3);
```

```
Сумма5 = 0;
```

```
Для Каждого ЭлементМассива Из НашМассив Цикл  
    Сумма5 = Сумма5 + ЭлементМассива;  
КонецЦикла;
```

```
Сообщить (Сумма5);
```

# Работа с массивами 1С

**Пример 10.** Получить сумму элементов числового массива

```
НашМассив = Новый Массив;  
НашМассив.Добавить (100);  
НашМассив.Добавить (-5);  
НашМассив.Добавить (3.4);  
НашМассив.Добавить (0);  
НашМассив.Добавить (3);
```

```
Сумма5 = 0;
```

```
Для Каждого ЭлементМассива Из НашМассив Цикл  
    Сумма5 = Сумма5 + ЭлементМассива;  
КонецЦикла;
```

```
Сообщить (Сумма5);
```

# Работа с массивами 1С

- Метод, который позволяет добавлять значения в коллекцию – **метод Вставить.**

**Синтаксис:** Вставить(<Индекс>,<Значение>)

Для Число=0 По 10 Цикл

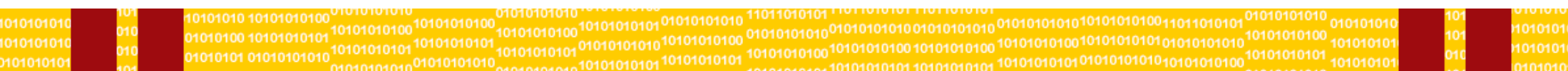
НашМассив.Вставить(0,Число);

КонецЦикла;

Метод, который позволяет удалять значения – **метод Удалить.**

**Синтаксис:** Удалить(<Индекс>)

**Пример использования:** НашМассив.Удалить(5);



# Дополнительные методы для Массива

- Метод **ВГраница()** возвращает количество элементов минус один.

```
// Использование цикла Для
КоличествоВМассиве = НашМассив.Количество();
Для Индекс=0 По КоличествоВМассиве-1 Цикл

    Сообщить(НашМассив[Индекс]);

КонецЦикла;
```

Переменную **КоличествоВМассиве** можно было определить:

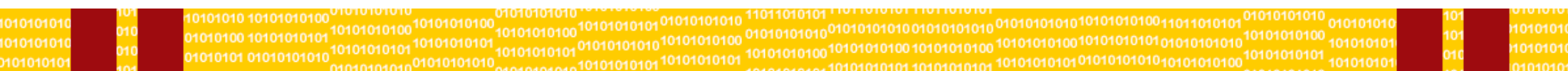
**КоличествоВМассиве = НашМассив.ВГраница();**

Тогда при описании самого цикла отнимать от данной переменной единицу не следует.



# Дополнительные методы для Массива

- Метод **Установить** позволяет присвоить значение элементу Массива по индексу.
  - Синтаксис: Установить(<Индекс>,<Значение>)
  - Пример: НашМассив.Установить (2,8);
  - Альтернативный вариант: НашМассив[2] = 8;
- Метод **Получить** позволяет прочитать значение по индексу, не обращаясь к использованию квадратных скобок.
  - Синтаксис: Получить(<Индекс>)
  - Пример: НашаПеременная = НашМассив.Получить(2);
  - Альтернативный вариант: НашаПеременная = НашМассив[2];



# Основные конструкции

*НашМассив = Новый Массив;  
НашМассив2 = Новый Массив(10)  
НашМассив.Добавить(34);  
ПозицияЭлемента = НашМассив.Найти(34);  
НашМассив.Удалить(1);  
НашМассив.Вставить(1,"Еще элемент");  
НашМассив.ВГраница();  
Значение = НашМассив.Получить(3);*

*Для Сч = 1 ПО НашМассив.Количество() Цикл  
    Сообщить(НашМассив[Сч]);  
КонецЦикла;*

*Для Каждого Элемент Из НашМассив Цикл  
    Сообщить(НашМассив[Сч]);  
КонецЦикла;*

# Структура

- Каждый элемент состоит из пары: **Ключ + Значение**.
- Ключ – строковый идентификатор, Значение любого типа.
- Доступ к элементам по ключу и через итератор  
*Для Каждого*.

*Пример:* НашаСтруктура.Код  
НашаСтруктура.Имя

# Работа со структурами

**Пример 11.** Необходимо хранить информацию о человеке - ФИО

```
Человек = Новый Массив (3) ;  
Человек [0] = "Иванов" ;  
Человек [1] = "Иван" ;  
Человек [2] = "Петрович" ;
```

```
Сообщить (Человек [2]) ;
```

# Работа со структурами

**Пример 11.** Необходимо хранить информацию о человеке - ФИО

```
Человек = Новый Структура ("Фамилия, Имя, Отчество") ;
```

```
Человек.Фамилия = "Иванов";
```

```
Человек.Имя = "Иван";
```

```
Человек.Отчество = "Петрович";
```

```
Сообщить (Человек.Отчество) ;
```

**Пример с содержанием структуры:**

```
НашаСтруктура = Новый Структура ("Код, Имя", 133, "Вася");
```

# Работа со структурами

**Пример с методом Вставить:**

НашаСтруктура.Вставить(“ЧленовСемьи”,3);

**Пример с методом Свойство:**

НашаСтруктура.Свойство ( “ЧленовСемьи” ) ;

**Обращение к свойствам Структуры по индексу:**

Сообщить(НашаСтруктура[“ЧленовСемьи”]);

# Основные конструкции

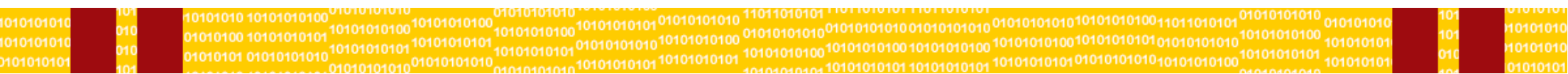
*НашаСтруктура = Новый Структура;  
НашаСтруктура = Новый Структура("Валюта,Курс");*

*НашаСтруктура.Валюта = "Руб."; НашаСтруктура.Курс = 1;  
НашаСтруктура.Вставить("Кратность",1);*

*Если НашаСтруктура.Свойство("Кратность") Тогда  
    Сообщить(НашаСтруктура.Курс \* НашаСтруктура.Кратность);  
КонецЕсли;*

*Для Каждого Элемент Из НашаСтруктура Цикл  
    Сообщить(Элемент.Ключ + « : » + Элемент.Значение);  
КонецЦикла;*

# 3 Еще примеры





# Массив в строку функцией СтрСоединить

СтрСоединить(<МассивСтрок>, <Разделитель>)

**Пример 12.** Массив содержащий элементы :

- «Обучение 1С»;
- «Разработка 1С»;
- «Программирование 1С».
- Преобразовать в строку. В качестве разделителя использовать: «;».

```
МассивСтрок = Новый Массив;  
МассивСтрок.Добавить ("Обучение 1С");  
МассивСтрок.Добавить ("Разработка 1С");  
МассивСтрок.Добавить ("Программирование 1С");  
  
СтрокаИзМассива = СтрСоединить (МассивСтрок, ";");
```

# Строка в массив подстрок

СтрРазделить(<Строка>, <Разделитель>, <ВключатьПустые>)

Строку из предыдущего примера, преобразовать в массив. В качестве разделителя использовать: «;»

```
МассивСтрок = СтрРазделить (СтрокаИзМассива, ";" );
```



**Надеюсь, Вы не  
слишком устали?**

**Спасибо за внимание!**