Выражения, операции, операнды и операторы

Руководитель:
ст. препод. Леонов Ю.А.
Выполнил:
Студент гр. 10-САПР:
Бордодымова И.Г.,

Содержание

1. Выражения
   1.1. Понятие выражения
   1.2. Состав выражения
2. Операции
   2.1. Понятие операции
   2.2. Приоритеты операций
   2.3. Виды операций
      2.3.1. Арифметические
      2.3.2. Отношения
      2.3.3. Поразрядно – логические
      2.3.4. Логические
      2.3.5. Сдвиговые
      2.3. 6. Строковые
      2.3. 7. Над множествами
      2.3. 8. Взятие адреса

Содержание

3. Операнды
   3.1. Понятие операндов
   3.2. Состав операндов
4. Операторы
   4.1. Понятие оператора
   4.2. Классификация операторов
      4.2.1. Простые
         - Присваивание
         - Безусловный переход
         - Групповой
         - Процедуры
         - Присоединение
      4.2.2. Структурные
         - Условие
         - Цикл

Выражения

Выражение – это совокупность констант, переменных и функций, объединённая знаками арифметических операций и скобками таким образом, чтобы данное выражение имело арифметический смысл.

Пример:

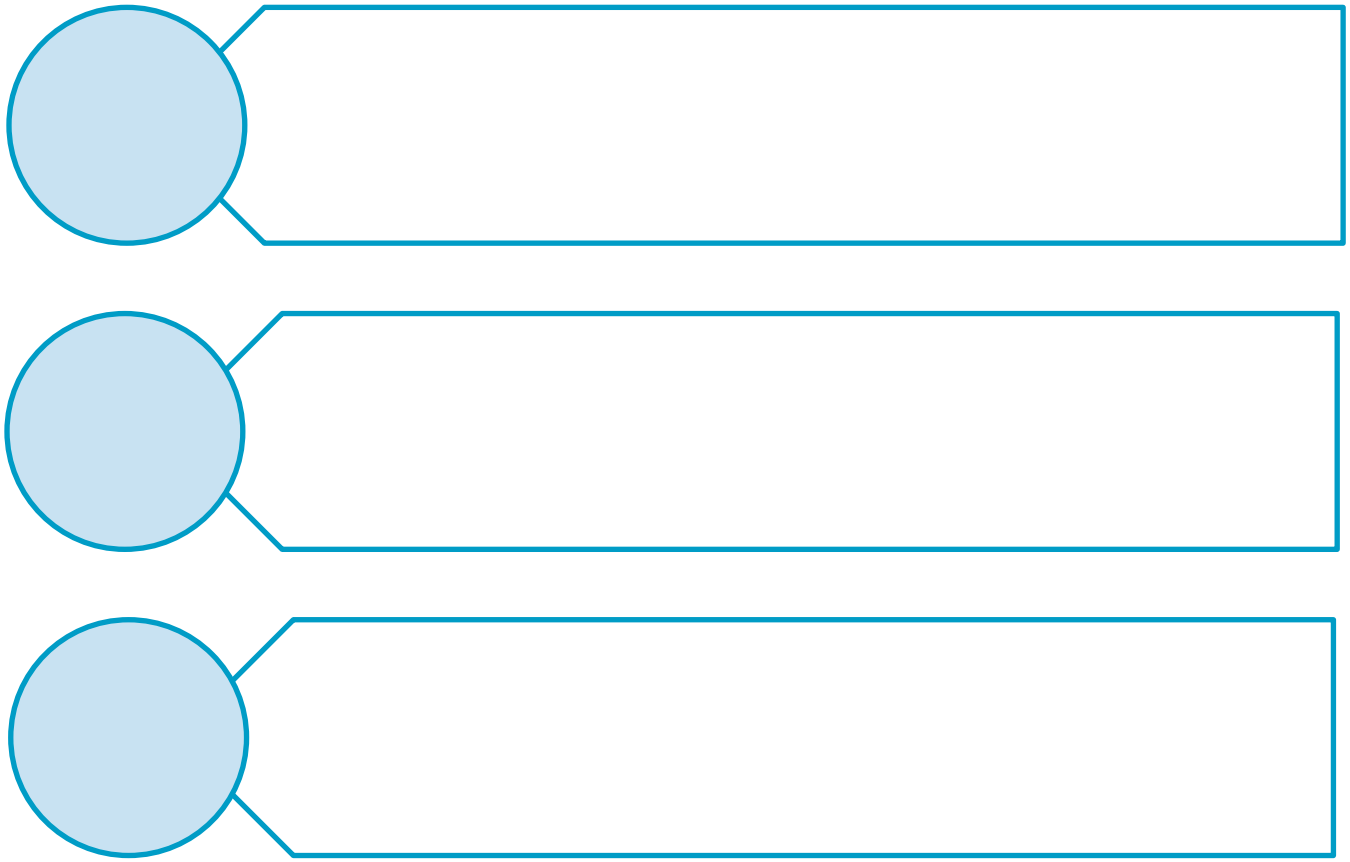
(x + y - 10)

17 div 5

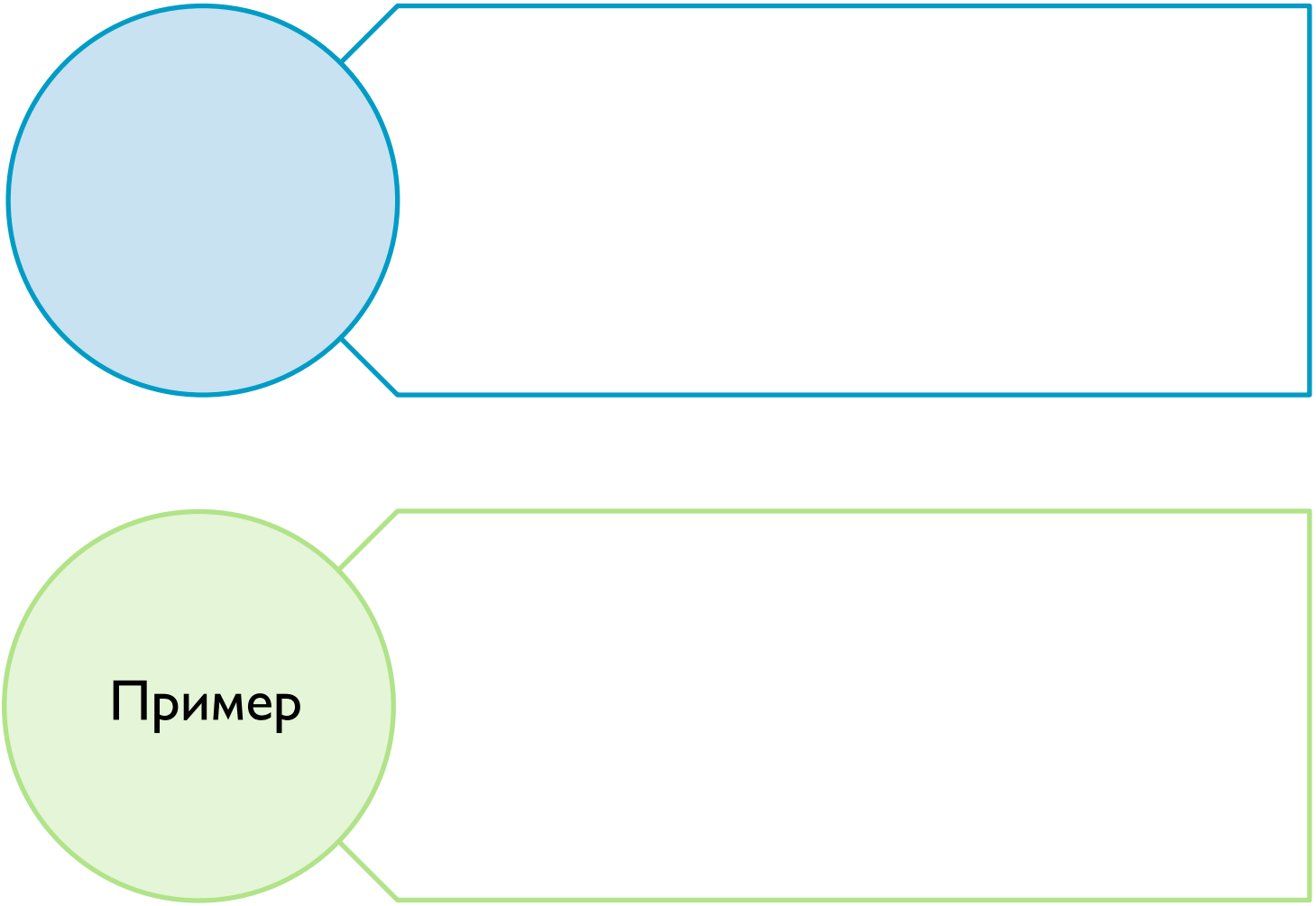
A or B

Выражения

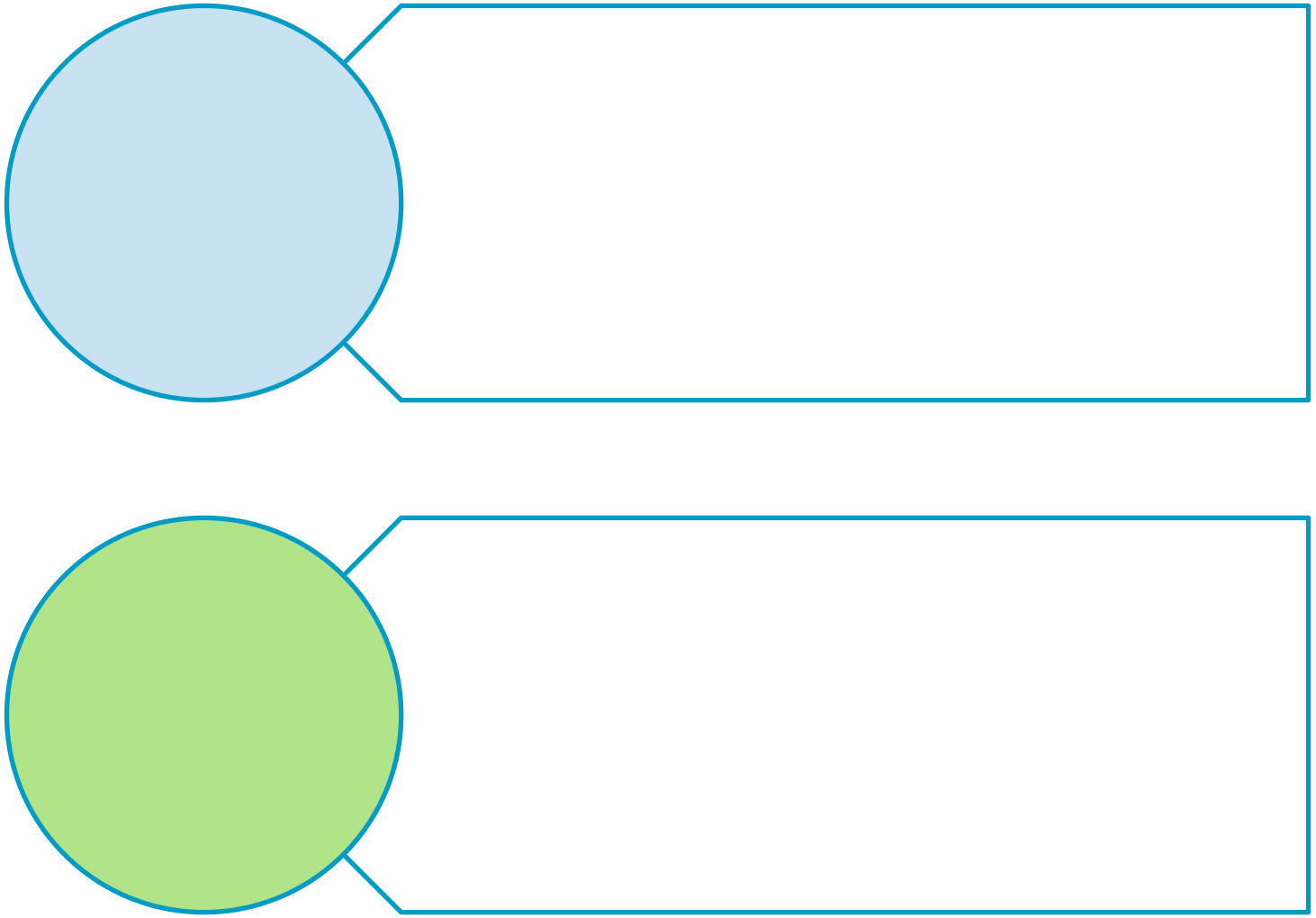
Выражение состоит из



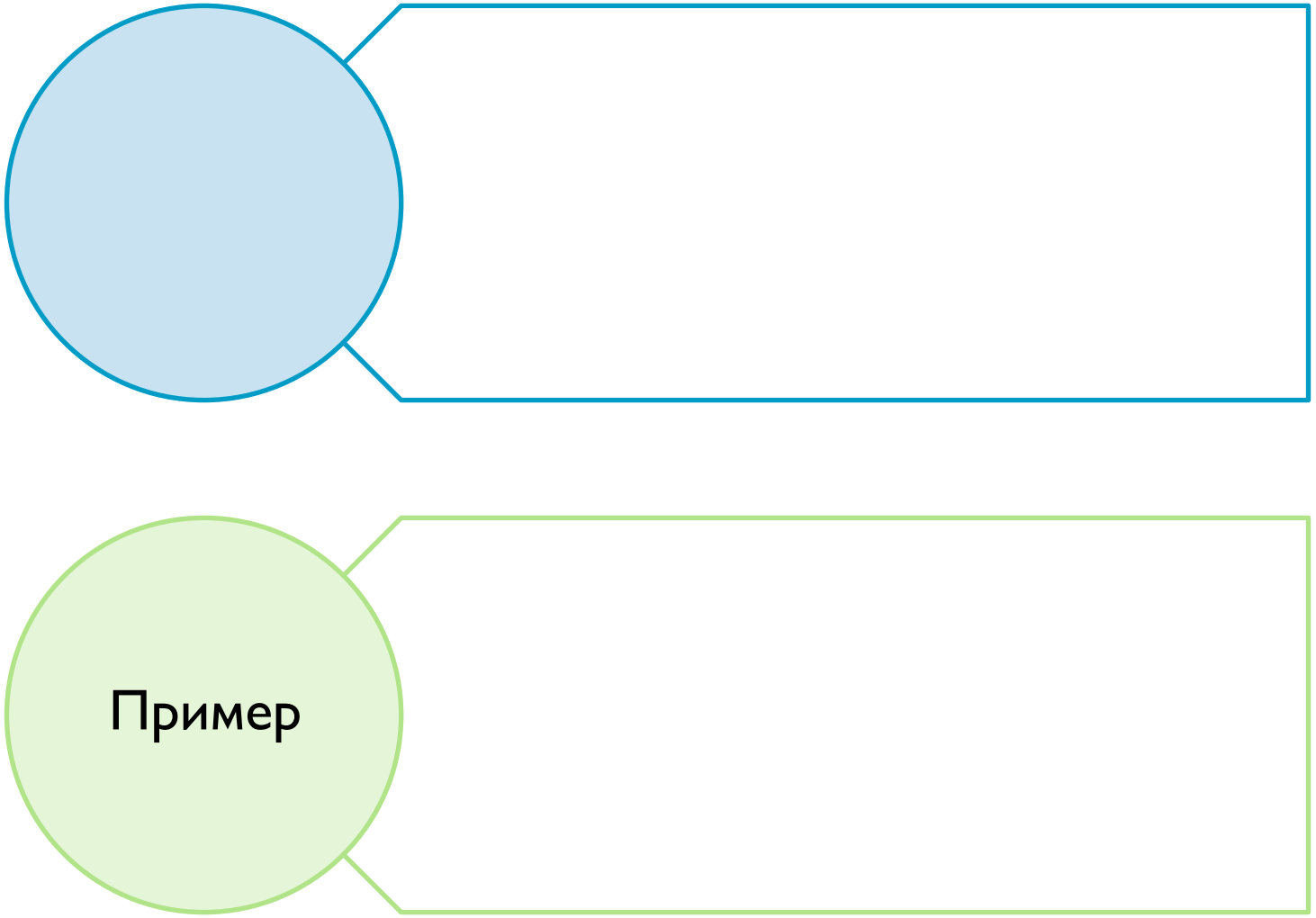
Выражения



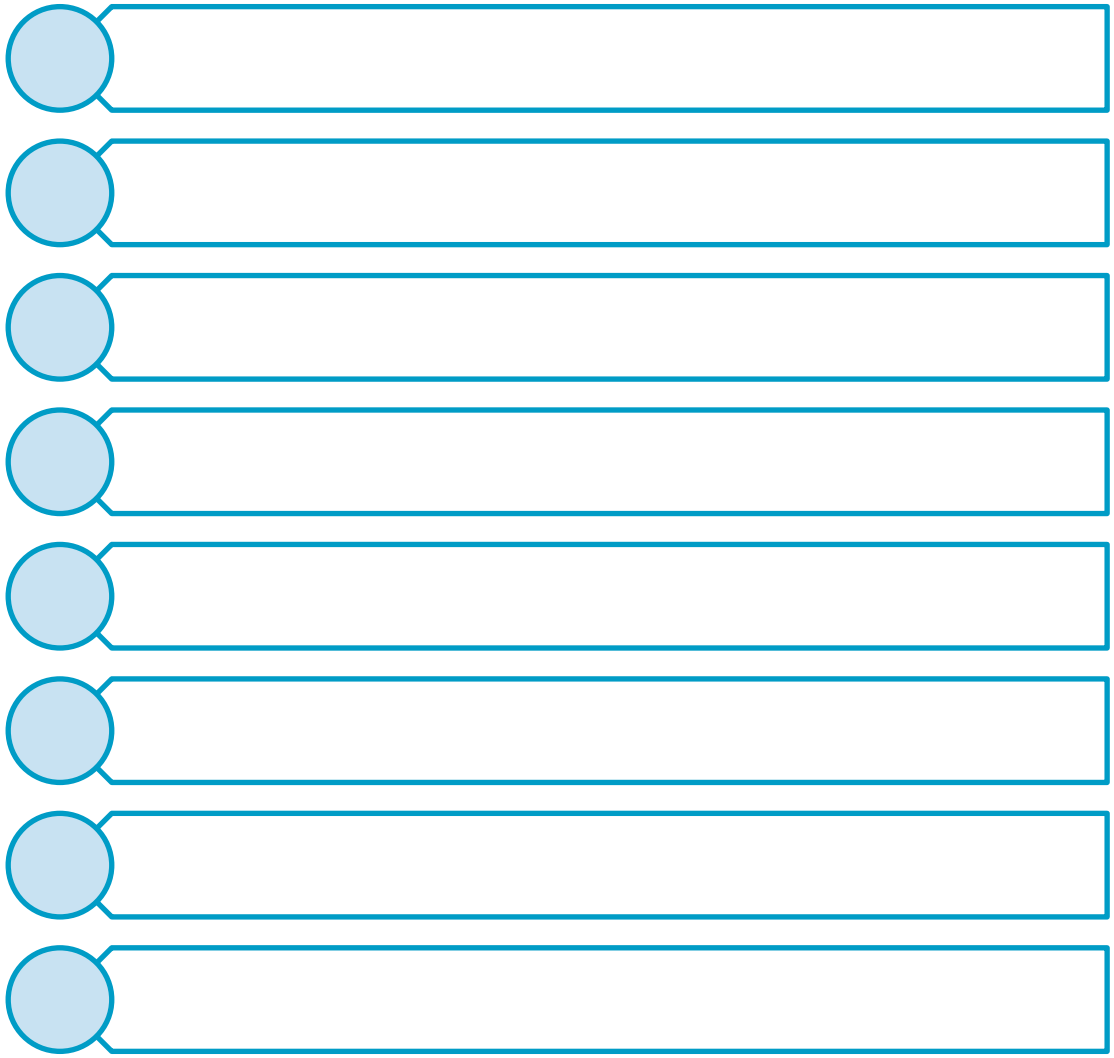
Выражения



Операции



Виды операций



Арифметические операции

Операция	Действие	Типы операндов	Типы результата
+	Сложение	Целый, вещественный	Целый, вещественный
-	Вычитание	Целый, вещественный	Целый, вещественный
*	Умножение	Целый, вещественный	Целый, вещественный
/	Деление	Целый, вещественный	Вещественный
Div	Целочисленное деление	Целый	Целый
Mod	Целочисленный остаток	Целый	Целый

Примеры арифметических выражений

Выражение	Результат	Операция
2 + 3	5	Плюс
4 - 1	3	Минус
2 * 3	6	Умножить
10 / 5	2	Разделить
17 div 5	3	Целочисленное деление
17 mod 5	2	Целочисленный остаток

Операции отношения

Операция	Действие	Типы операндов	Типы результата
=	Равно	Логический	Логический
<>	Не равно	Логический	Логический
<	Меньше	Логический	Логический
>	Больше	Логический	Логический
<=	Меньше либо равно	Логический	Логический
>=	Больше либо равно	Логический	Логический

Примеры выражений отношения

Выражение	Результат
a = b	Истина, если a равно b Ложь, если a не равно b
a <> b	Истина, если a не равно b Ложь, если a равно b
a < b	Истина, если a меньше b Ложь, если a больше, либо равно b
a <= b	Истина, если a меньше, либо равно b Ложь, если a больше b
a > b	Истина, если a больше b Ложь, если a меньше, либо равно b
a >= b	Истина, если a больше, либо равно b Ложь, если a меньше b

Логические операции

Операция	Действие	Тип операнда	Тип результата
Not	Отрицание	Булевский	Булевский
And	И	Булевский	Булевский
Or	Или	Булевский	Булевский
Xor	Исключающее или	Булевский	Булевский

Примеры простых логических выражений

Операция «Not»

A	not A
True	False
False	True

Операция «And»

A	B	A and B
True	True	True
True	False	True
False	True	True
False	False	False

Примеры простых логических выражений

Операция «Or»

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

Операция «Xor»

A	B	A xor B
True	True	False
True	False	True
False	True	True
False	False	True

Поразрядные логические операции

Операция	Действие	Тип операнда	Тип результата
Not	Побитовое отрицание	Целочисленный	Целочисленный
And	И	Целочисленный	Целочисленный
Or	Или	Целочисленный	Целочисленный
Xor	Исключающее или	Целочисленный	Целочисленный

Сдвиговые операции

Операция	Действие	Тип операндов	Тип результата
Shl	Бинарный сдвиг влево	Целочисленный	Целочисленный
Shr	Бинарный сдвиг вправо	Целочисленный	Целочисленный

Пример: 5 shl 1 = 10

5_{10}	0	0	0	0	0	0	0	1	0	1
Результат 10_{10}	0	0	0	0	0	0	1	0	1	0

Операции со строками

Операция	Действие	Типы операнда	Тип результата
+	Конкатенация (склеивание)	Строковый, символьный	Строковый

Пример:

Выражение	Результат
'Object'+' Pascal'	'Object Pascal'
'ABC'+'D'	'ABCD'
'X'+'Z'	'XZ'

Операции над множествами

Операция	Действие	Типы операнда	Тип результата
+	Объединение	«Базовый тип», множество	Множество
-	Разность	«Базовый тип», множество	Множество
*	Пересечение	«Базовый тип», множество	Множество
in	Проверка вхождения во множество	«Базовый тип», множество	Булевский
=	Равно	Булевский	Булевский
<>	Не равно	Булевский	Булевский
<=	Меньше либо равно	Булевский	Булевский
>=	Больше либо равно	Булевский	Булевский

Операции над множествами

Примеры:

[1, 2, 3, 4] + [3, 4, 5, 6] => [1, 2, 3, 4, 5, 6]

[1, 2, 3, 4] - [3, 4, 5, 6] => [1, 2]

[1, 2, 3, 4] * [3, 4, 5, 6] => [3, 4]

[a, b, c, d] = [a, b, c, d] => True

[a, b, c, d] = [b, c, a, d] => True

[a, b, c, d] <> [a, b, c, d]=> False

[1, 2, 3, 4] >= [1, 2, 3 ,4] => False

[1, 2, 3, 4] <= [1, 2, 3, 4] => False

[1, 2] in [1, 2, 3, 4] => True

Операция взятия адреса

Операция	Действие	Тип операнда	Тип результата
@	Взятие адреса	Идентификатор константы, переменной, процедуры, функции	Адрес

Операция @ возвращает адрес переданного операнда.

Пример:

```
Var
   X: byte;
   P: pointer;
Begin
   P:=@X; { адрес переменной X присваиваем в переменную P }
End.
```

Приоритет операций

Приоритетом называется очерёдность выполнения операции в выражения.

Операции одного приоритета выполняются по порядку, слева направо; порядок выполнения операций можно менять с помощью круглых скобок.

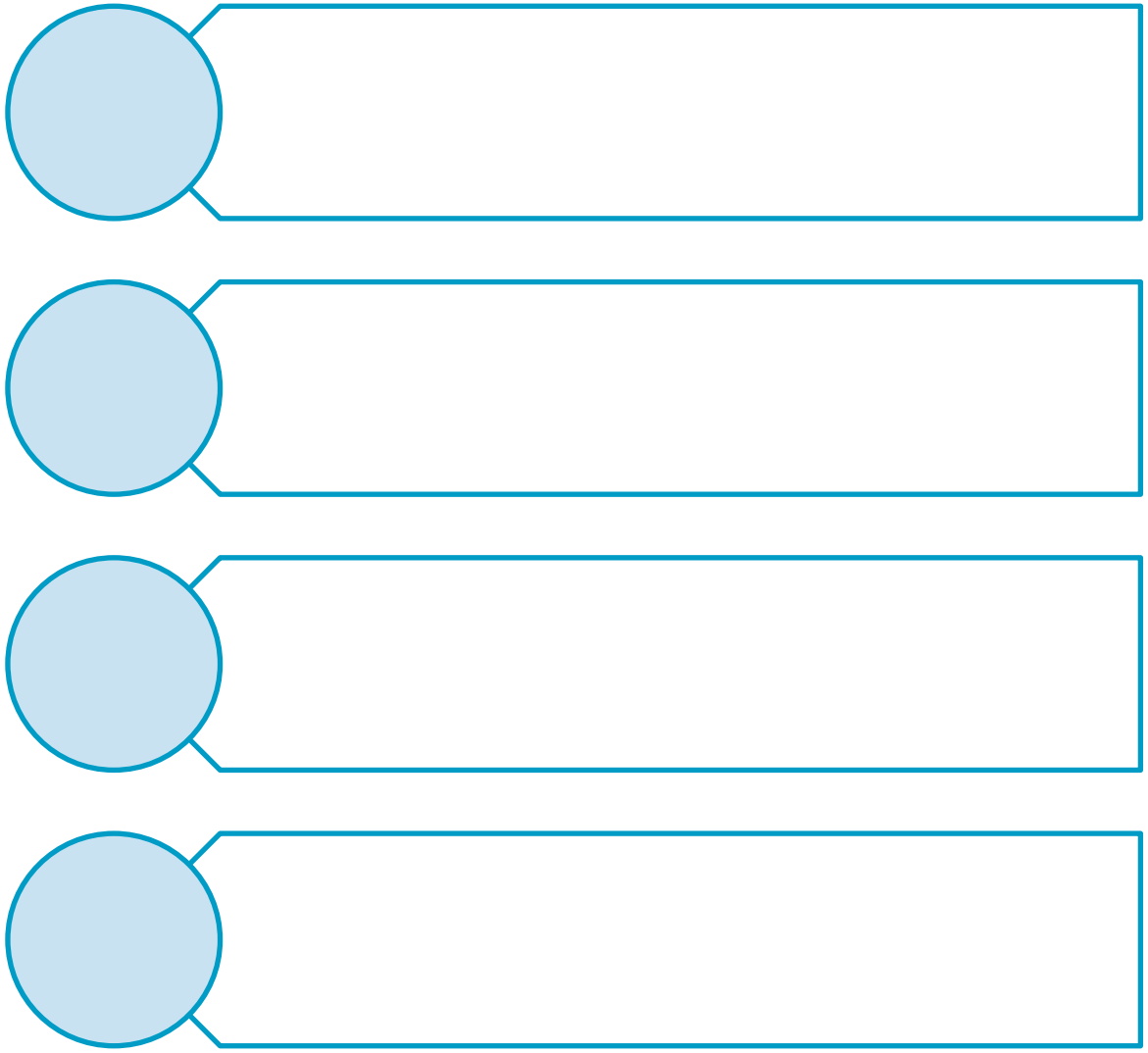
Пример:

(1) (2) (5) (3) (4)

(x + y) * 5 - (z + 8) / 7

1	2	3	4	5
(x + y)	(x + y) * 5	z + 8	(z + 8) / 7	(x + y) * 5 - (z + 8) / 7

Приоритеты



Операнды

Операнд – величина, представляющая собой элемент, реализуемый в ходе выполнения программы вычислений.

Пример:

В выражении $(x + y - 10)$,

где x, y и 10 – операнды.

Операнды

Операнды могут быть представлены с помощью:

Константы и переменные

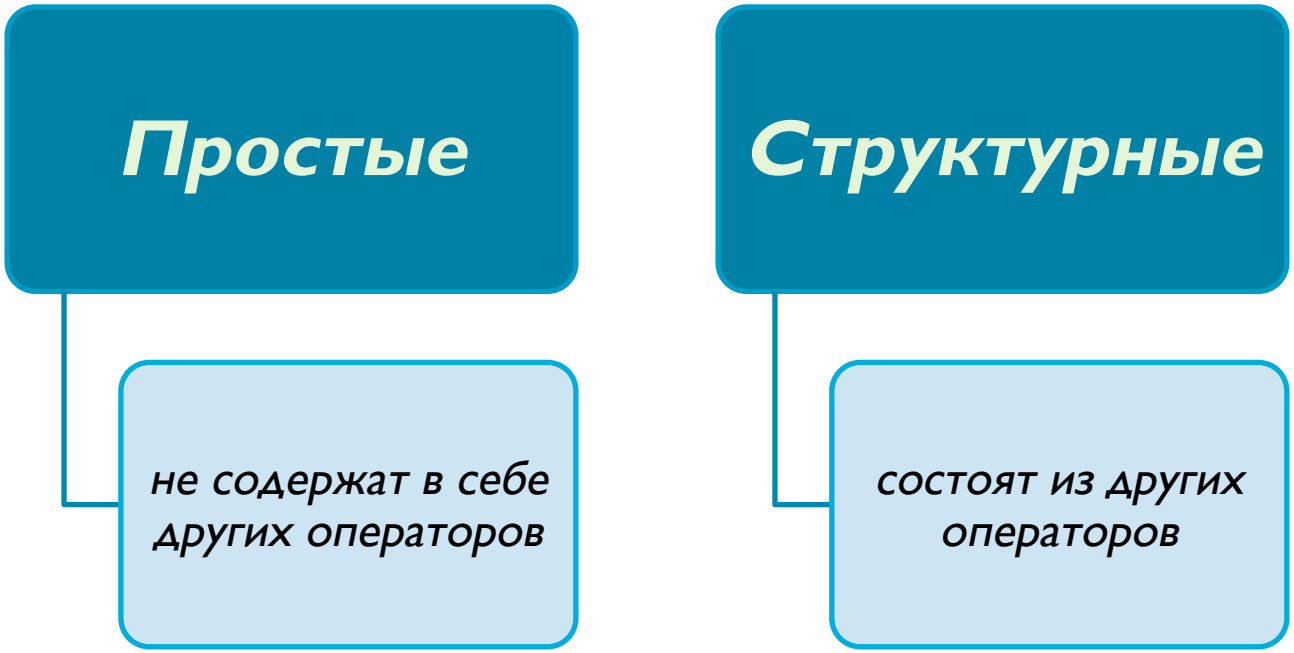
Все переменные и константы, используемые в программе, должны быть перечислены в разделе описания переменных и констант.

Пример:

```
Var
    Day: word;
    Money: real;
Const
    MyBirthYear =1992;
    MySalary = ‘invisible’;
```

Операторы

Операторы описывают действия, которые должны выполняться.



Операторы

Простые операторы
Присваивания
Процедуры
Безусловного перехода
Групповой
Присоединения

Оператор присваивания

Выполнение оператора присваивания приводит к вычислению выражения, значение которого помещается в переменную.

Синтаксис:

<переменная> : = <выражение> ;

Оператор присваивания

Пример:

```
x := 5+3;
{ выражение: 5+3; результат: x = 8 }
i := sqr(7);
{ функция sqr вычислит квадрат; результат i = 49 }
r := 'Turbo' + 'Pascal';
{ выполняется операция «конкатинация» результат присваивается в переменную r }
```

Оператор процедуры

Оператор процедуры состоит из идентификатора, в котором указаны фактические параметры.

Синтаксис:

<имя_процедуры> (<список формальных параметров>);

<имя_функции> (<список формальных параметров>);

Выполнение данного оператора приводит к запуску действий, описанных в теле процедуры.

Оператор процедуры

Пример:

Написать процедуру, которая выводит на экран строку, состоящую из звездочек.

```
Procedure StarLine (len: integer);
Var
  I : integer;
Begin
  For I :=1 to len do
  Write ( '*' );
End.
```

Оператор перехода

Оператор перехода прерывает естественный порядок выполнения программы и указывает, что дальнейшее выполнение должно продолжаться, начиная с оператора, помеченного меткой

Синтаксис:

Goto <имя_метки>;

Групповой оператор

Групповой оператор используется для объединения операторов в единую группу.

Синтаксис:

```
Begin
    <оператор_1;>
    <оператор_2;>
    …
    <оператор_n;>
End;
```

Используется совместно со структурными операторами.

Оператор присоединения

Оператор *with* используют для краткого обращения к полям записи.

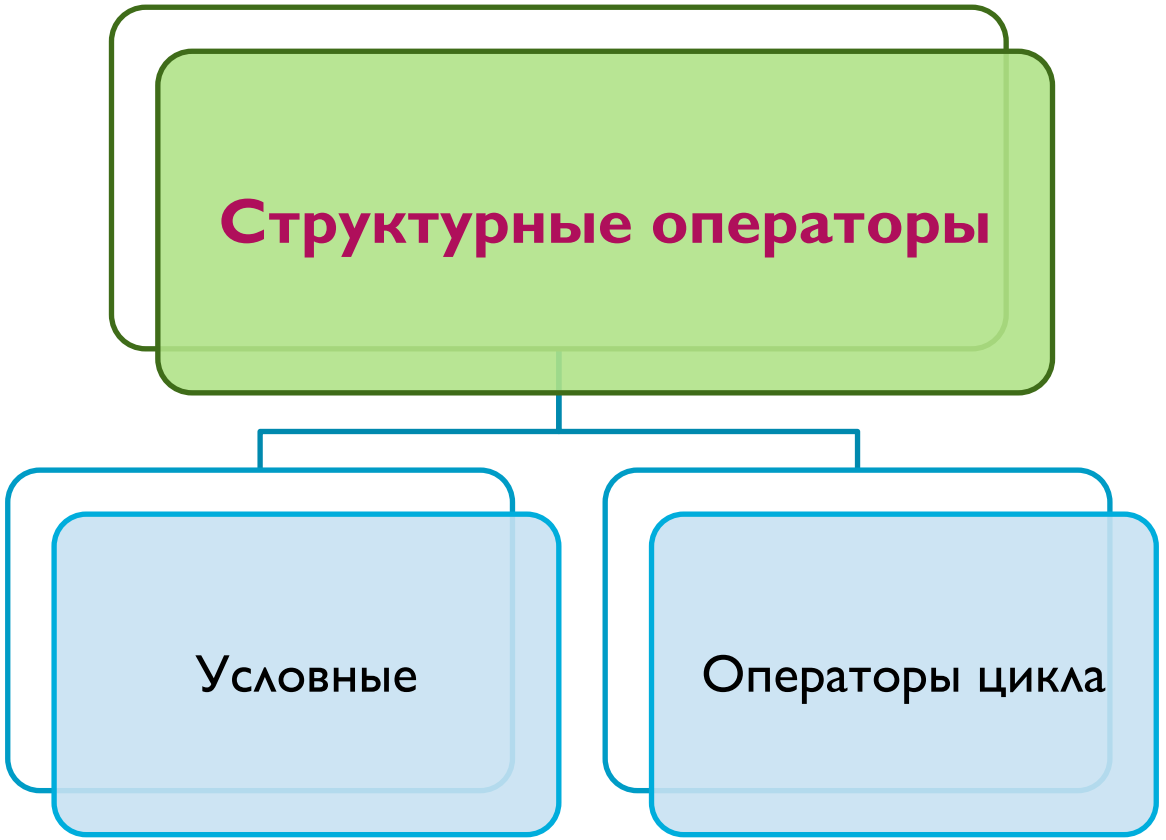
Синтаксис:

With <выражение> do <оператор;>

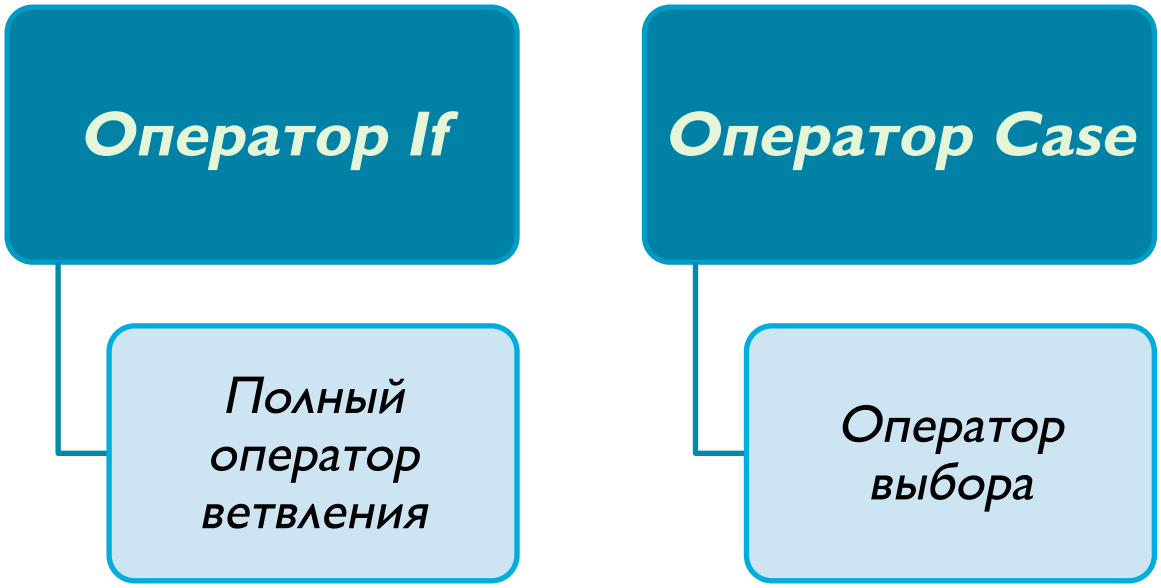
Пример:

```
Type TPoint = record
          x, y : integer;
          color : integer;
         End;
Var Point: TPoint;
BEGIN
    With Point do begin
       x := 10; y := 20;
       color := Red;
    end;
…
END.
```

Структурные операторы

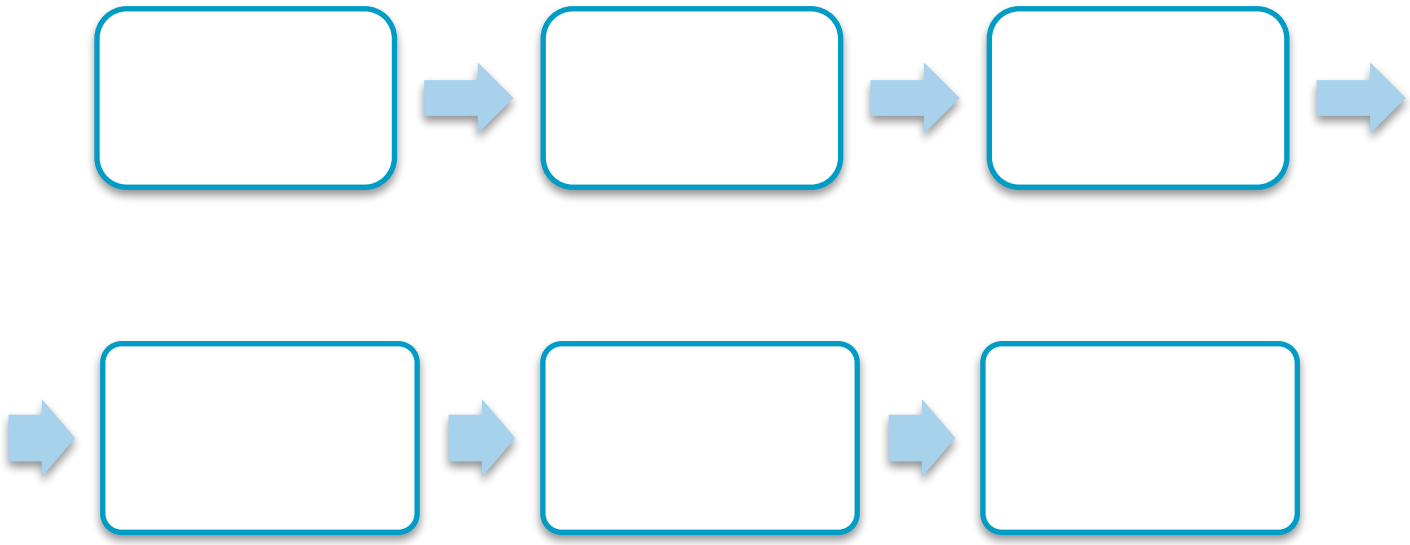


Условные операторы



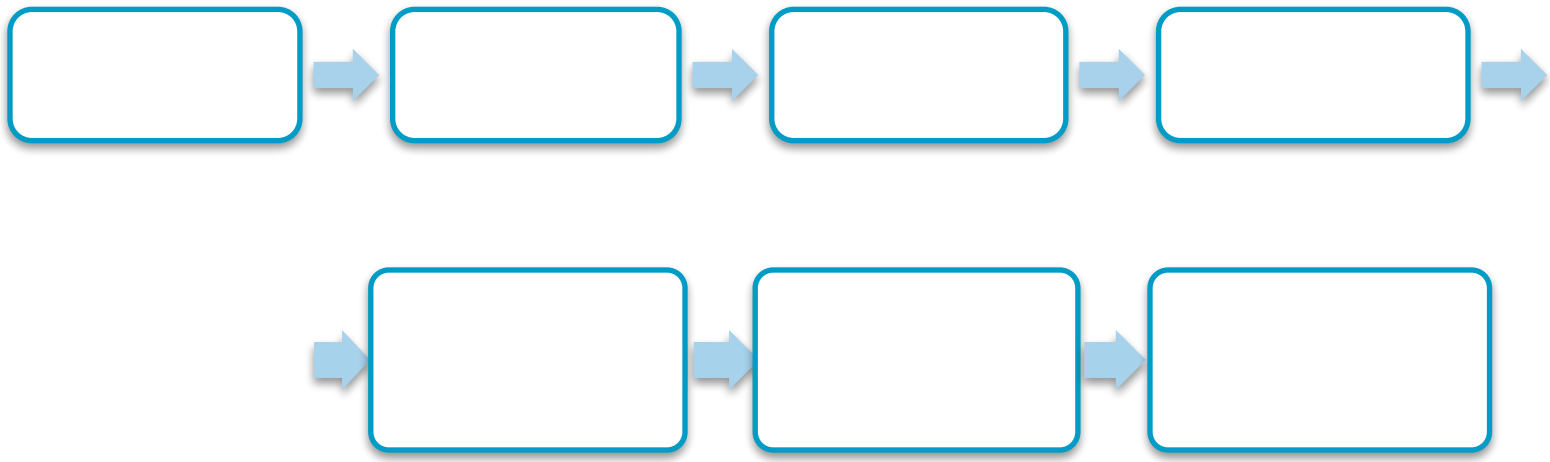
Оператор if

Синтаксис оператора *if* можно представить следующим образом:



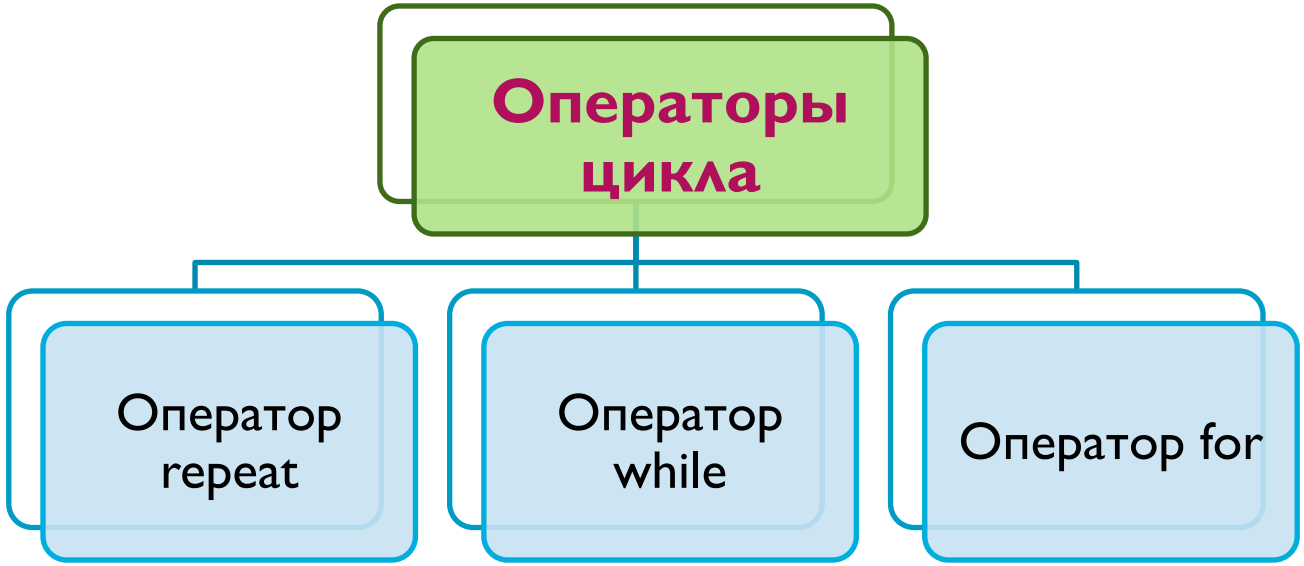
Оператор Case

Синтаксис оператора *case* можно представить следующим образом:



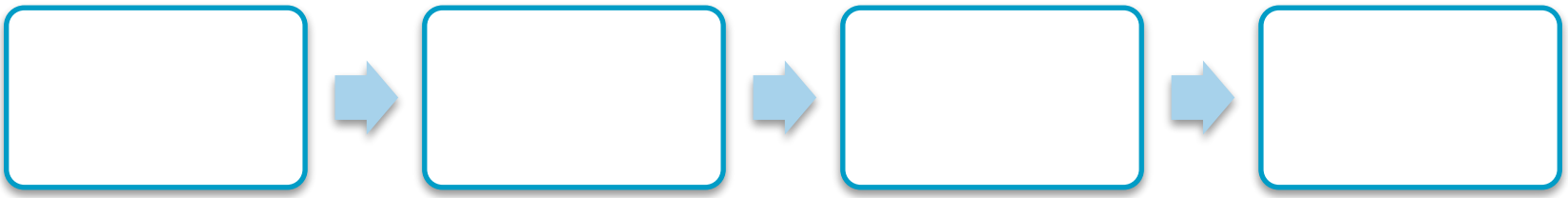
Оператор цикла

Операторы цикла позволяет многократно выполнить некоторое множество действий.



Оператор Repeat

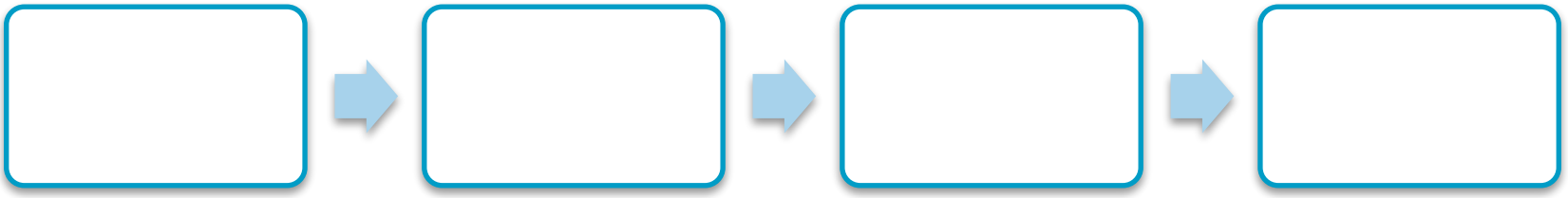
Синтаксис оператора *repeat* можно представить следующим образом:



Выполняется до тех пор, пока условие, стоящее после слова until, ложно

Оператор While

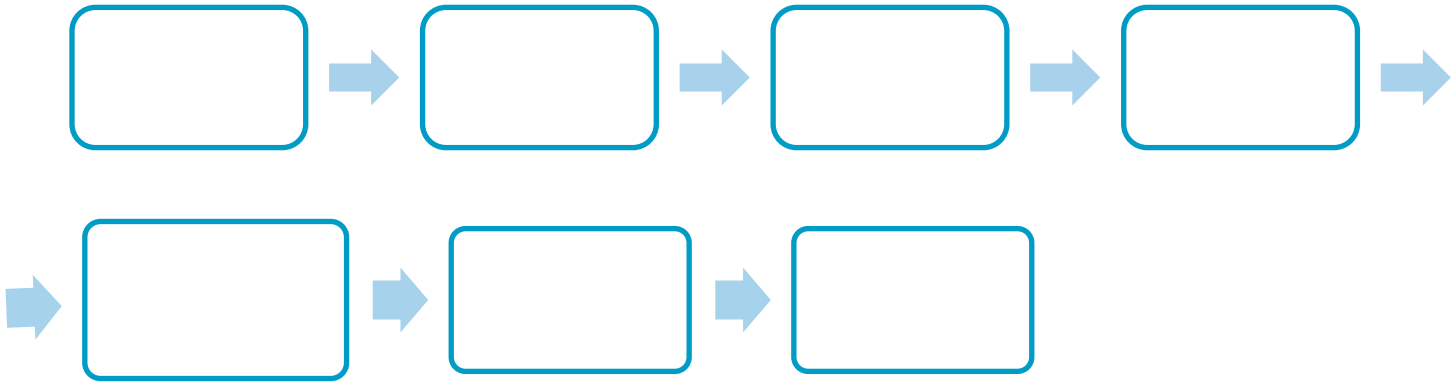
Синтаксис оператора *while* можно представить следующим образом:



Выполняется до тех пор, пока условие, записанное после слова while, истинно.

Оператор For

Синтаксис оператора *for* можно представить следующим образом:



Количество повторений цикла определяется начальным и конечным значениями переменной-счетчика

Контрольные вопросы

1) Что называется выражением?
2) Из чего состоят выражения?
3) Для чего предназначены операции?
4) Какие виды операций вы знаете?
5) Что называется приоритетом операции?
6) Перечислите типы приоритетов.
7) Дайте определение операнду.
8) Чем могут быть представлены операнды?
9) Что такое константы и переменные?
10) Что называется оператором?
11) На какие группы подразделяются операторы?
12) Расскажите про простые операторы.
13) Расскажите про составные операторы.

Список литературы

- Немнюгин, С.А. Turbo Pascal: программирование на языке высокого уровня / С.А. Немнюгин. – 2-е изд. – СПб.: Питер, 2006. – 544с.
- Культин Н. Б. Turbo pascal в задачах и примерах. – СПб.: БХВ – Петербург 2006. – 256 с.
- Симонович С.В. Информатика: Базовый курс/С.В. Симонович – СПб.: «Питер», 2001.
- Интернет-ресурс Wikipedia.org.