# Операторы языка

- Арифметические операторы
- Операторы сравнения
- Оператор конкатенации
- Логические операторы
- Оператор присваивания
- Условный оператор
- Оператор выбора
- Циклические операторы

# Арифметические операторы:

- Возведение в степень(^)
- Унарный минус(-)
- Умножение(*)
- Деление(/)
- Целочисленное деление(\)
- Деление по модулю(Mod)
- Сложение(+)
- Вычитание(-)

# Операторы сравнения:

| Оператор | Синтаксис | Описание |
|---|---|---|
| = | A = B | Равенство: Если A равно B, то - True. Иначе - False |
| < | A < B | Меньше: Если A меньше B, то - True. Иначе - False |
| <= | A <= B | Меньше или равно: Если A меньше или равно B, то - True. Иначе - False |
| > | A > B | Больше: Если A больше B, то - True. Иначе - False |
| >= | A >= B | Больше или равно: Если A больше или равно B, то - True. Иначе - False |
| <> | A <> B | Не равно: Если A не равно B, то - True. Иначе - False |
| Like | A Like B | Сравнение по образцу: Если A равно B, то – True. Иначе – False<br>? – любой одиночный символ<br>* - любое количество символов или их отсутствие<br># - любая цифра |
| Is | A Is B | |

# Оператор конкатенации:

Конкатенация строк – присоединение одной строки к другой.

Для конкатенации можно использовать 2 оператора:

- Оператор конкатенации (&)
- Оператор сложения (+)

# Логические операторы:

## Оператор AND:

Синтаксис: *A* AND *B*

Если A и B имеют значение True, то True. Иначе – False.

## Оператор OR:

Синтаксис: *A* OR *B*

Если любой из операндов имеет значение True, то – True. Иначе – False.

## Оператор NOT:

Синтаксис: NOT *A*

Если A имеет значение False, то - True. Иначе – False.

## Оператор XOR:

Синтаксис: *A* XOR *B*

Если A имеет значение True или B имеет значение True, то - True. Иначе - False

## Оператор EQV :

Синтаксис: *A* EQV *B*

Если A имеет такое же значение что и B, то - True. Иначе – False.

## Оператор IMP:

Синтаксис: *A* IMP *B*

Если A имеет значение True и B имеет значение False, то - False. Иначе – True.

# Оператор присваивания:

Оператор присваивания используется для присваивания результата выражения переменной.

Синтаксис:

Переменная = выражение

*Пример: x=x*10 +15*

# Условный оператор:

## Однострочная форма:

If *условие* Then [*оператор*][Else *оператор*]

## Сложная форма:

If *условие* Then
    [*операторы*]
[ElseIf *условие1* Then
    [*операторы1*]…
[Else
    *Операторы n*]
End If

# Оператор выбора (Select Case):

Синтаксис:

Select Case *выражение*

[Case *список выражений-n*

[*операторы-n*]…]

[Case Else

[*операторы_иначе*]]

End Select

# Операторы цикла с предусловием (While и Do While):

Синтаксис While:

While *условие*
  [*операторы*]
Wend

Синтаксис Do While:

Do [While *условие*]
  [*операторы*]
[Exit Do]
  [*операторы*]
Loop

# Оператор цикла с предусловием (Do Until)

Синтаксис:

Do [Until *условие*]

  [*операторы*]

[Exit Do]

  [*операторы*]

Loop

# Операторы цикла с постусловием (Do… Loop While, Do… Loop Until )

Синтаксис Do … Loop While:

```
Do
   [операторы]
[Exit Do]
   [операторы]
Loop [While условие]
```

Синтаксис Do … Loop Until:

```
Do
   [операторы]
[Exit Do]
   [операторы]
Loop [Until условие]
```

# Оператор цикла с предусловием (For … Next)

For *счетчик* = *начало* To *конец* [Step]

[*операторы*]

[Exit For]

[*операторы*]

Next [*счетчик*]

# Оператор цикла с предусловием (For Each … In …)

For Each *элемент* In *группа*

[*операторы*]

[Exit For]

[*операторы*]

Next [*элемент*]

# *Спасибо за внимание*