

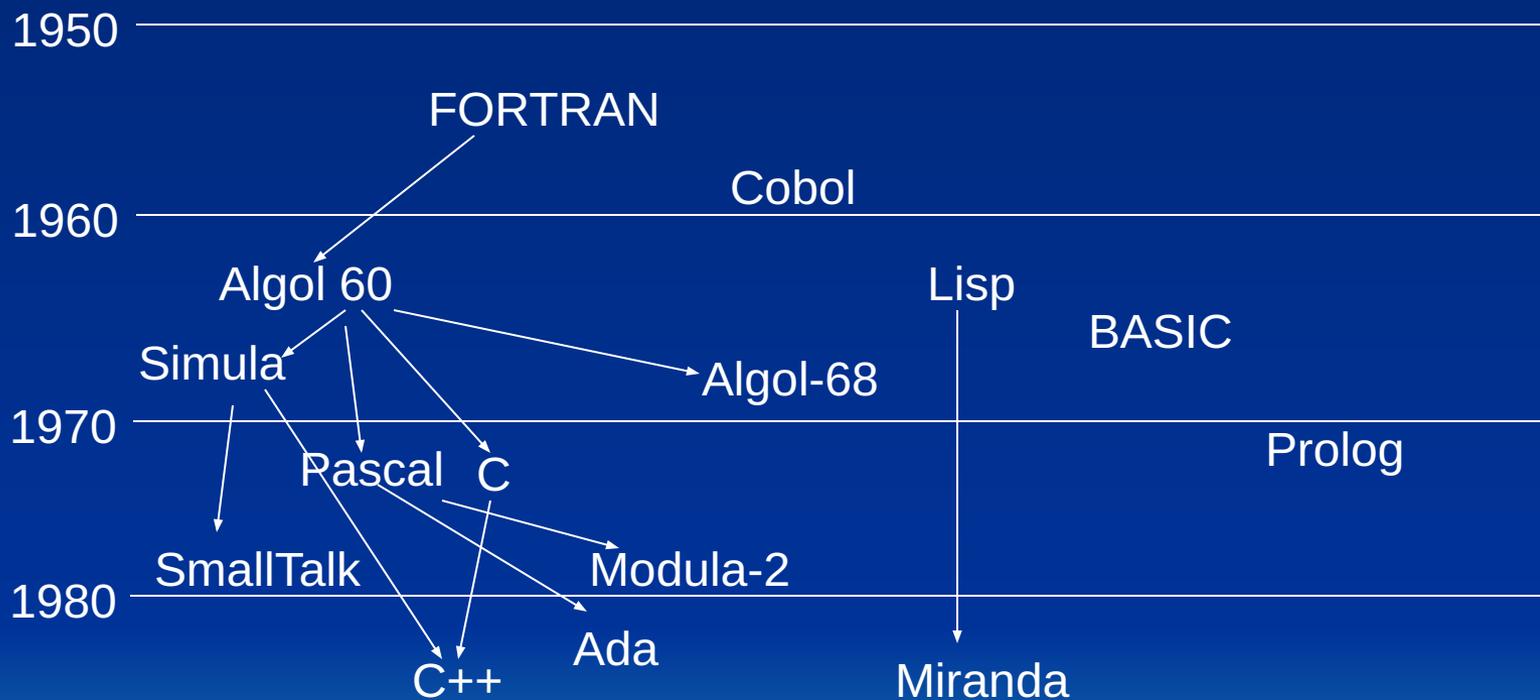
# Основы программирования на Visual Basic

Кафедра экономической информатики МГУ

Андрей Олегович Медников



# История создания ЯП



# Требования к алгоритмам

- Наличие ввода исходных данных
- Наличие вывода результата выполнения
- Однозначность
- Общность
- Корректность
- Конечность
- Эффективность.



# Создание компьютерной игры

- Создание трехмерного графического движка.
- Написание модуля искусственного интеллекта созданий, населяющих уровень.
- Реализация управления с различных игровых манипуляторов.
- Создание звукового оформления.
- ...и так далее.



# Сложение 2 чисел

- Считать число  $A$
- Считать число  $B$
- Выполнить суммирование  $A + B$
- Вывести результат сложения
- Закончить работу



# Решение уравнения $ax=b$

- Считать число  $A$
- Считать число  $B$
- Если число  $A = 0$  и число  $B$  равно  $0$ , то вывести на экран ответ ( $x$  – любое) и закончить программу.
- Если число  $A = 0$  и число  $B$  не равно  $0$ , то вывести на экран ответ (решений нет) и закончить программу.
- Вывести ответ  $B/A$ .
- Закончить программу



# Переменные в ЯП

- Переменная – это одна или несколько ячеек оперативной памяти компьютера, которым присвоено определенное имя, то есть просто некое место в памяти компьютера, причем чаще всего, определенного размера, в котором хранятся какие-то данные.



# Замечание об именах переменных

Сравните:

- `x1`, `y1` и `MaxXResolution`, `MaxYResolution`



# Обязательное объявление переменных до использования

- В VB для того, чтобы включить обязательное объявление переменных, необходимо в начале модуля поставить строку `Option Explicit`



# Объявление переменных

- Dim <имя переменной> As <имя типа>
- Private <имя переменной> As <имя типа>
- Public <имя переменной> As <имя типа>
- Static <имя переменной> As <имя типа>

# Если слово *As* опущено

```
Dim MaxXRes, MaxYRes as Integer
```

Переменной *MaxXRes* будет присвоен тип не *Integer*, как мы бы могли ожидать, а *Variant*



# Простые типы в VB

Boolean	True, False	2 байта
Byte	0...255	1 байт
Integer	-32768..32767	2 байта
Long	~-2000000000...2000000000	4 байта
Decimal	Примерно 30 десятичных цифр	12 б.
Single	-3.4E38...-1.4E-45, 1.4E-45...3.4E38	4 байта
Double	-1.7E308...-4.9E-324, 4.9E-324...1.7E308	8 б.
String	До $2^{16}$ символов	10 + 1
Date	от 1 января 100 г. до 31 декабря 9999 г.	8 б.
Object	Ссылка на объект (указатель)	4 байта
Variant		> 16 б.

# Примеры объявлений перемен.

```
Dim MaxXRes As Integer, MaxYRes as Integer
```

```
Dim Count As Long
```

```
Dim CellsEmpty As Boolean
```

```
Dim QuestionText As String
```



# Оператор присваивания

$x=x+1$

Этот оператор решает 3 важные задачи:

- Вычисление значения выражения в правой части оператора.
- Вычисление выражения в левой части оператора присваивания, выражение должно определять адрес ячейки памяти.
- Копирование значения, вычисленного на шаге 1, в ячейки памяти, начиная с адреса, полученного на шаге 2.



# Оператор присваивания

[Let] <имя переменной> = <выражение>.

MaxXRes = 1024

MaxYRes = 768

Count = 1000000

CellsEmpty = True

QuestionText = "Who am I?"

Count = Count + 1

MaxNumberOfPoints = MaxXRes\* MaxYRes

NumberOfBits = NumberOfBytes\*8



# Модуль в VB

```
Sub Name1()
```

```
...
```

```
End Sub
```



# Ввод информации в VB

```
stringvar = InputBox (Prompt [, Title])
```

```
MaxXRes = InputBox (“Введите макс. разр. по оси X”)
```



# Вывод информации в VB

```
MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```



# Программа сложения 2 чисел

```
'Option Explicit
```

```
Sub SumExamp()
```

```
Dim A as Integer, B As Integer, Sum As Integer
```

```
A = InputBox("Введите первое слагаемое")
```

```
B = InputBox("Введите второе слагаемое")
```

```
Sum = A + B
```

```
MsgBox ("Сумма " & A & " и " & B & " = " & Sum)
```

```
End Sub
```



# Обмен значений

```
Dim A As Integer, B As Integer, Temp As Integer
```

```
A = 5
```

```
B = 10
```

```
Temp = A
```

```
A = B
```

```
B = Temp
```



# Обмен значений в случае целых

Dim A As Integer, B As Integer

A = 5

B = 10

A = A + B

B = A - B

A = A - B



# Константы

`Const ИмяКонст [As ТипКонст] = ЗначКонст`

`Const Pi As Double = 3.141592`

`Const Tax As Single = 0.13`

`Const Greeting = "Привет"`



# Процедурная область действия

```
Sub Sum()
```

```
Dim A As Integer, B As Integer, C As Integer
```

```
A = 5
```

```
B = 7
```

```
C = A + B
```

```
MsgBox "Сумма A и B = " & C
```

```
End Sub
```

```
Sub Difference()
```

```
Dim A As Integer, B As Integer, C As Integer
```

```
A = 5
```

```
B = 7
```

```
C = A - B
```

```
MsgBox "Разность A и B = " & C
```

```
End Sub
```



# Модульная область действия

```
Option Explicit
```

```
Const Pi = 3.14159219841984
```

```
Sub CircleLength()
```

```
Dim Radius As Double
```

```
Dim Length As Double
```

```
Radius = InputBox("Введите радиус...")
```

```
Length = Radius * 2 * Pi
```

```
MsgBox "Длина окружности радиуса " & Radius & " равна " &  
Length
```

```
End Sub
```



# Переменные с один. именами в разных областях действия

```
Option Explicit
```

```
Const Test As Integer = 100
```

```
Sub Test1()
```

```
    Dim Test As Double
```

```
    Test = 3.1415
```

```
    MsgBox Test
```

```
End Sub
```

```
Sub Test2()
```

```
    Dim Test As String
```

```
    Test = "aaaaaaaaaa"
```

```
    MsgBox Test
```

```
End Sub
```

```
Sub Test3()
```

```
    MsgBox Test
```

```
End Sub
```



# Время жизни переменных

Время жизни переменных определяет, как долго переменная сохраняет свое значение.

Переменная, объявленная внутри процедуры при помощи Dim, теряет свое значение при выходе из нее.

Для объявления переменной на процедурном уровне, но со временем жизни модуля – Static.

Static NumberOfCalls as Integer



# Математические операции

- ^ Возведение в степень
- \* / Умножение и деление
- \ Целочисленное деление
- Mod Нахождение остатка
- + - Сложение и вычитание



# Операции сравнения

$>$  Больше, чем

$<$  Меньше, чем

$\geq$  Больше или равно, чем

$\leq$  Меньше или равно, чем

$=$  Равно

$\neq$  Не равно



# Сравнение строк

"фио"="фио"

"фио"<>" фио"

"фио" < "фио "

"фиот" > "фио"



# Сравнение строки с шаблоном

StrExpr Like Templ

Символы в шаблоне:

#

\*

?

[list]

[!list]

В квадратных скобках также можно указать и диапазон ([a-z])



# Сравнение строки с шаблоном - пример

```
Sub TemplateExamp()
```

```
Dim MyCheck
```

```
MyCheck = "aBBBa" Like "a*a" ' Возвращает True.
```

```
MyCheck = "F" Like "[A-Z]" ' Возвращает True.
```

```
MyCheck = "F" Like "[!A-Z]" ' Возвращает False.
```

```
MyCheck = "a2a" Like "a#a" ' Возвращает True.
```

```
MyCheck = "aM5b" Like "a[L-P]#[!c-e]" ' Возвращает True.
```

```
MyCheck = "BAT123khg" Like "B?T*" ' Возвращает True.
```

```
MyCheck = "CAT123khg" Like "B?T*" ' Возвращает False
```

```
End Sub
```



# Логические операции

Not      Логическое Не

And      Логическое И

Or      Логическое Или



# Таблица истинности

X	TRUE	TRUE	FALSE	FALSE
Y	TRUE	FALSE	TRUE	FALSE
NOT X	FALSE	FALSE	TRUE	TRUE
X AND Y	TRUE	FALSE	FALSE	FALSE
X OR Y	TRUE	TRUE	TRUE	FALSE



# Порядок вычисления

1. Вначале всегда части выражения, заключенные в круглые скобки, если выражение в круглых скобках само сложное, то применяем эти же правила.
2. Затем выполняем в соответствии с приоритетом (смотри таблицу ниже)
3. Если равный уровень приоритета, то вычисляем слева направо.



# Приоритет операций

Dim A As Integer, B As Integer

$$A = 6 + 5 * 4$$

$$B = (6 + 5) * 4$$

‘ Внимание, A не равно B!!!



# Приоритет операций

- Знаки арифметических операций
- Знаки конкатенации строк
- Операторы сравнения
- Логические операторы



# Приоритет операций

$\wedge$  - возведение в степень

- - унарный минус

$*/$  - умножение и деление

$\backslash$  - целочисленное деление

Mod - остаток

+, -

&

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

Not

And

Or

Xor

Eqv



# Найти значение Result

Dim Result As Boolean, X As Boolean

Dim Y As Boolean, Z As Boolean

Dim A As Integer, B As Integer

A = 5: B = 10

X = FALSE: Y = TRUE: Z = FALSE

**Result** = X AND (Y OR (Z AND Y) AND \_  
(NOT (X OR Z))) OR (A > B)



# Оформление программы

- Символ разделения 2 операторов в одной строке - двоеточие:

A = 5: B = 7

- Если оператор не помещается в одной строке, пробел и подчеркивание:

```
MyAddr = "Город:" & City & ", улица:" & Street _  
& "дом:" & Number
```

- Комментарий – все, что после апострофа

```
Dim Count As Integer 'Count – счетчик числа элем.
```



# Оператор простого выбора

If Условие Then Операторы [Else Операторы]

Пример 1:

```
If Balance - Check < 0 Then MsgBox "На вашем счету перерасход"
```

Пример 2:

```
Sub TestIf()
```

```
Dim Number as Integer
```

```
Number = InputBox("Введите число")
```

```
If Number > 0 Then MsgBox ("Положительн.") Else MsgBox  
("Отрицат.")
```

```
End Sub
```



# Многострочный оператор простого выбора

```
If Условие Then  
  Оператор1  
  ...  
  ОператорN  
End If
```

Или же, при необходимости второй ветви:

```
If Условие Then  
  Оператор1  
  ...  
  ОператорN  
Else  
  Оператор1  
  ...  
  ОператорN  
End If
```

# Многострочный оператор простого выбора

```
If Balance - Check < 0 Then
```

```
    MsgBox "Ваши средства израсходованы!"
```

```
    MsgBox "Банк уведомлен!"
```

```
Else
```

```
    Balance = Balance - Check
```

```
End If
```



# Вложенные операторы If... Then

```
Option Explicit
Sub DiscountCalculation()
Dim InitialPrice As Double, PriceToPay As Double
InitialPrice = InputBox("Введите цену покупки")

If InitialPrice > 10000 Then
    PriceToPay = InitialPrice * 0.9
Else
    If InitialPrice > 5000 Then
        PriceToPay = InitialPrice * 0.95
    Else
        If InitialPrice > 1000 Then
            PriceToPay = InitialPrice * 0.9
        Else
            PriceToPay = InitialPrice
        End If
    End If
End If
End If
MsgBox "С вас - " & PriceToPay & " руб."
End Sub
```

# Решение уравнения $ax=b$

```
Option Explicit
Sub LinearEq()
Dim A as Double, B as Double, X as Double
A = InputBox("Введите коэф. A")
B = InputBox("Введите коэф. B")
If (A = 0) And (B=0) Then
    MsgBox "X – любое число"
Else
    If A=0 Then
        MsgBox "Решений нет"
    Else
        X = B / A
        MsgBox "X =" & X
    End If
End If
End Sub
```



# Решение квадратного уравнения (не полн.)

```
Option Explicit
Sub SqrEq()
Dim A as Double, B as Double, C as Double, D As Double
Dim X1 as Double, X2 as Double
A = InputBox("Введите коэф. A")
B = InputBox("Введите коэф. B")
C = InputBox("Введите коэф. C")
D = B*B -4*A*C

If D < 0 Then
    MsgBox "Решений нет"
Else
    If D = 0 Then
        X = - B / (2*A)
        MsgBox "Решение – x = " & X
    Else
        X1 = (-B - SQR(D))/(2*A)
        X2 = (-B + SQR(D))/(2*A)
        MsgBox "Решения: x1 = " & X1 & "x2 = " & X2
    End If
End If
End Sub
```



# Оператор множественного выбора

```
Select Case <Проверяемое выражение>  
Case <Список выражений1>  
    <Операторы1>  
Case <Список выражений2>  
    <Операторы2>  
...  
[Case Else  
    <ОператорыElse> ]  
End Select
```

Пример:

```
Select Case Argument  
Case Is < 0  
...  
Case 0 To 10  
...  
Case 100, 1000, 10000  
...  
Case Else  
...  
End Select
```



# Встроенные функции преобразования данных

`Asc(S)` – код первого символа строки `S`

`Chr(X)` – строка из 1 символа, соотв. коду `X`

`Str(X)` – строка, представляющая число `X`

`Val(S)` – число, соотв. числу, представленному строкой `S`, если строка содержит не только цифры и 1 точку, но возвращает 0.

