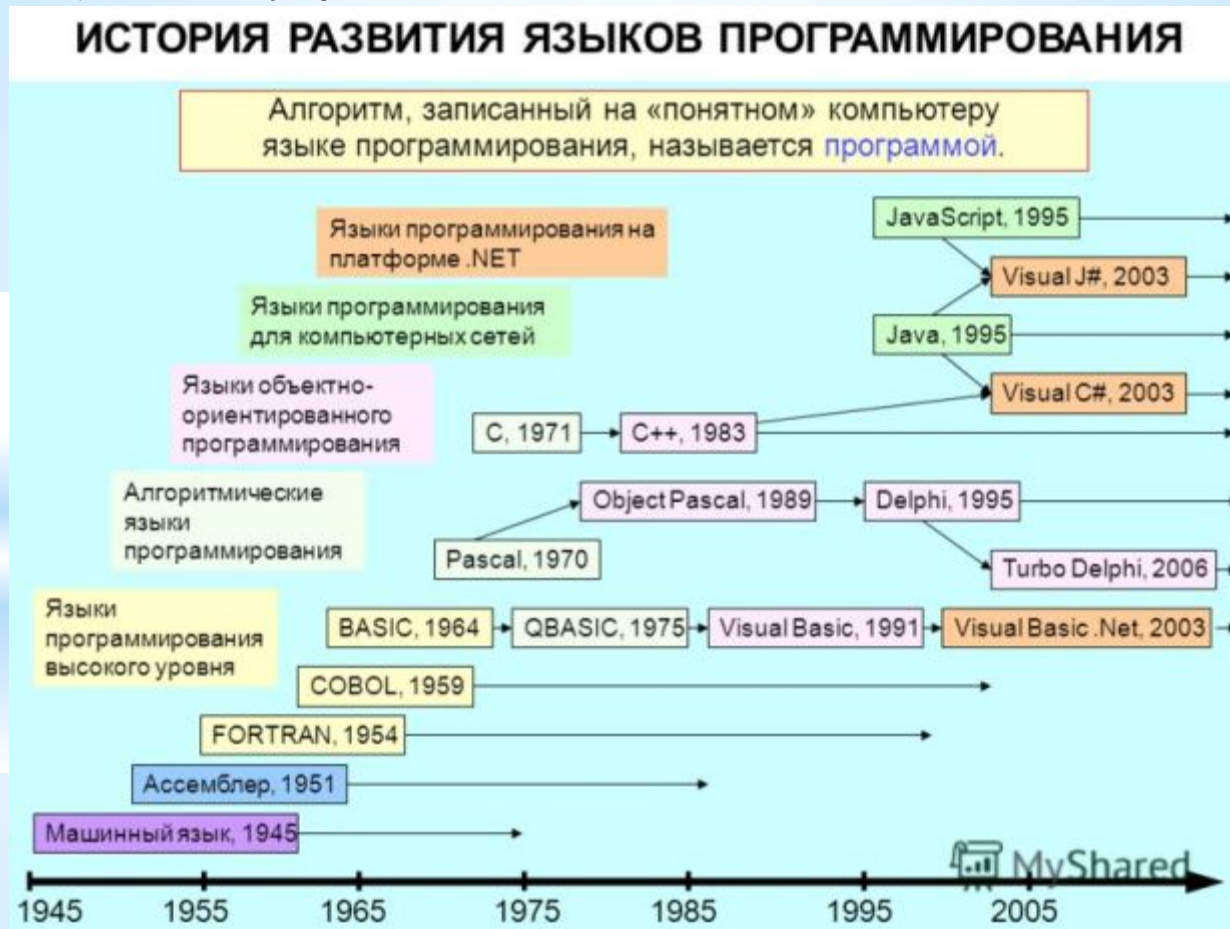


Введение

Языки программирования

Язык программирования – формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно – ЭВМ) под её управлением.



Краткая история языков программирования

Первые языки программирования были очень примитивными и мало чем отличались от формализованных упорядоченных последовательностей единиц и нулей, понятных компьютеру.

Использование таких языков было крайне неудобно с точки зрения программиста, так как он должен был знать числовые коды всех машинных команд, должен был сам распределять память под команды программы и данные.

Пример: машинный код

```
00000100: 2E 74 65 78-74 00 00 00-CA 65 00 00-00 10 00 00 .text  ve
00000110: 00 66 00 00-00 06 00 00-00 00 00 00-00 00 00 00 f
00000120: 00 00 00 00 20 00 00 60 2E 64 61 74 61 00 00 00 .data
00000130: 44 19 00 00-00 80 00 00-00 06 00 00-00 6C 00 00 D1  M
00000140: 00 00 00 00-00 00 00 00-00 00 00 00-40 00 00 C0 @
00000150: 2E 72 73 72-63 00 00 00-00 60 00 00-00 A0 00 00 .rsrc
00000160: 00 54 00 00-00 72 00 00-00 00 00 00-00 00 00 00 l r
00000170: 00 00 00 00-40 00 00 40-4D 22 D1 38-48 00 00 00 @ @M"811
00000180: 46 22 D1 38 55 00 00 00 27 C2 F2 37 61 00 00 00 Г"8U '7a
00000190: 34 D0 44 38 6C 00 00 00 34 D0 44 38 79 00 00 00 4D81 4D8y
000001A0: 84 D3 2B 38-86 00 00 00-34 D0 44 38-90 00 00 00 д7+8Ж 4D8v
000001B0: 46 22 D1 38-9B 00 00 00-00 00 00 00-00 00 00 00 F"8H
000001C0: 63 6F 6D 64-6C 67 33 32-2E 64 6C 6C-00 53 48 45 comdlg32.dll SHE
000001D0: 4C 4C 33 32-2E 64 6C 6C-00 4D 53 56-43 52 54 2E LL32.dll MSVCRT.
000001E0: 64 6C 6C 00 41 44 56 41 50 49 33 32 2E 64 6C 6C dll ADVAPI32.dll
000001F0: 00 4B 45 52 4E 45 4C 33 32 2E 64 6C 6C 00 47 44 KERNEL32.dll CD
00000200: 49 33 32 2F-64 6C 6C 00-55 53 45 52-33 32 2F 64 T32.dll USER32.d
00000210: 6C 6C 00 57-49 4F 53 50-4F 4F 4C 2F-44 52 56 00 11 WTSPOOL.DRV
00000220: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00000230: EB 02 EB 05-C8 F9 FF FF-FF 58 83 C0-1B 8D A0 01
00000240: FC FF FF 83-C4 FC 8B EC-33 C9 66 B9-8F 01 80 30
00000250: 00 40 E2 GA E8 60 00 00 00 47 65 74 50 72 6F 63 @
00000260: 41 64 64 72 65 73 73 00 4C 6F 61 64 4C 69 62 72 Address LoadLibr
00000270: 61 72 79 41-00 43 72 65-61 74 65 50-72 6F 63 65 aryA CreateProce
00000280: 73 73 41 00-45 78 69 74-50 72 6F 63-65 73 73 00 ssA ExitProcess
00000290: 41 00 62 69-6E 64 00 6C-69 73 74 65-6E 00 61 63 ws2 32 WSASocket
000002A0: 63 65 70 74 00 63 6D 64 00 5A 52 BB 00 00 F0 77 A bind listen ac
000002B0: 81 3B 4D 5A 90 00 74 03 4B EB F5 8B 73 3C 03 F3 ;MZ tKvns<v
000002C0: 8B 76 78 03-F3 8B 7F 20-03 FB 8B 4F-14 56 33 C0 0xv~ vnnv3v
000002D0: 57 51 8B 3F-03 FB 8B F2-33 C9 B1 0F-F3 A6 59 5F NOv?v v3v:8jY_
000002E0: 74 08 83 C7-04 40 E2 E8-FF E1 5E 8B-56 24 03 D3 (v@v v v v v
```


Ассемблер

Для того, чтобы облегчить общение человека с ЭВМ были созданы языки программирования типа Ассемблер. Переменные величины стали изображаться символическими именами. Числовые коды операций заменились на мнемонические обозначения, которые легче запомнить. Язык программирования приблизился к человеческому языку, и отдалился от языка машинных команд.

```
class2_ctor    proc near                ; CODE XREF: ...
arg_0_p_compare_func= dword ptr  4

    push     esi
    push     450h                       ; dwBytes
    call    new
    mov     esi, eax
    pop     ecx
    test    esi, esi
    jz     short loc_100125B3
    lea    eax, [esi+class_2.csec]
    push   eax                          ; lpCriticalSection
    call   ds:InitializeCriticalSection
    mov    eax, [esp+4+arg_0_p_compare_func]
    mov    [esi+class_2.setup_class13], offset class2_setup_class13
    mov    [esi+class_2.append], offset append_to_existing
    mov    [esi+class_2.remove], offset class2_remove ; (this, key)
    mov    [esi+class_2.clear], offset class2_clear
    mov    [esi+class_2.exists], offset class2_exists
    mov    [esi+class_2.count], offset class2_count
    mov    [esi+class_2.get_next_value], offset class2_get_next_value
    mov    [esi+class_2.get_prev_value], offset class2_get_prev_value
    mov    [esi+class_2.get_values_as_array], offset class2_get_values_in_array
    mov    [esi+class_2.dtor], offset class2_dtor
    mov    [esi+class_2.p_compare_func], eax
    call   class2_allocate_block_pair ; 1 = success
                                           ; 0 = fail
    test   eax, eax
    jnz   short loc_100125B7
    push   esi                          ; lpMem
    call   class2_dtor
    pop    ecx

loc_100125B3:                                ; CODE XREF: ...
    xor    eax, eax
    pop    esi
    retn

; -----
loc_100125B7:                                ; CODE XREF: ...
    mov    eax, esi
    pop    esi
    retn
class2_ctor endp
```

Фортран

Фортран (*Fortran*) — первый язык программирования высокого уровня, имеющий транслятор. Создан в период с 1954 по 1957 год группой программистов под руководством Джона Бэкуса в корпорации IBM. (Язык Планкалкюль, претендующий на пальму первенства, был изобретён ещё в 1945 году, но не был реализован вплоть до 2000 года.) Название Fortran является сокращением от **FOR**mula **TRAN**slator (переводчик формул). Фортран широко используется в первую очередь для научных и инженерных вычислений. Одно из преимуществ современного Фортрана — большое количество написанных на нём программ и библиотек подпрограмм.

Первоначально язык создавался специально для вычислительной системы IBM 407. Но популярность FORTRAN'a привела к тому, что даже производители других архитектур стали выпускать собственные трансляторы. Вследствие этого, в 1966 году был выпущен общий стандарт FORTRAN 66.

Пример программы на языке Фортран

```
*L
C      CALCULATE STATISTICS ON DATA FROM LOW SPEED READER
      SUM=0
      SUMSQ=0
      TYPE 100
100    FORMAT("ENTER THE NUMBER OF VALUES TO CALCULATE STATISTICS ON",/)
      ACCEPT 10,N
10     FORMAT(I)
      DO 200 I=1,N
      READ 1,110,V
110    FORMAT(E)
      SUM=SUM + V
      SUMSQ=SUMSQ + V*V
      TYPE 120,I,V
120    FORMAT("VALUE",I,"IS",E,/)
200    CONTINUE
      SAMP=N
      AVRG=SUM/SAMP
      STD=SQRT(SUMSQ/SAMP - AVRG**2)
      TYPE 300,N,AVRG,STD
300    FORMAT("NUMBER OF VALUES",I,"MEAN",E,"STANDARD DEVIATION",E,/)
      END
```

```
*
.R FORT
?
```


АЛГОЛ

Уже в конце 50-х гг. появилась хорошая альтернатива FORTRAN'у. Питер Наур разработал алгоритмический язык, независимый от архитектуры. Его назвали ALGOL. В данном случае, разработчики также постарались расширить возможности языка, максимально приблизив систему обозначений к математической.

```
FLOATING POINT ALGOL TEST'  
BEGIN REAL A,B,C,D'  
  
READ D'  
  
FOR A:= 0.0 STEP D UNTIL 6.3 DO  
BEGIN  
  PRINT PUNCH(3),££L??'  
  B := SIN(A)'  
  C := COS(A)'  
  PRINT PUNCH(3),SAMELINE,ALIGNED(1,6),A,B,C'  
END'  
END'
```

Бейсик

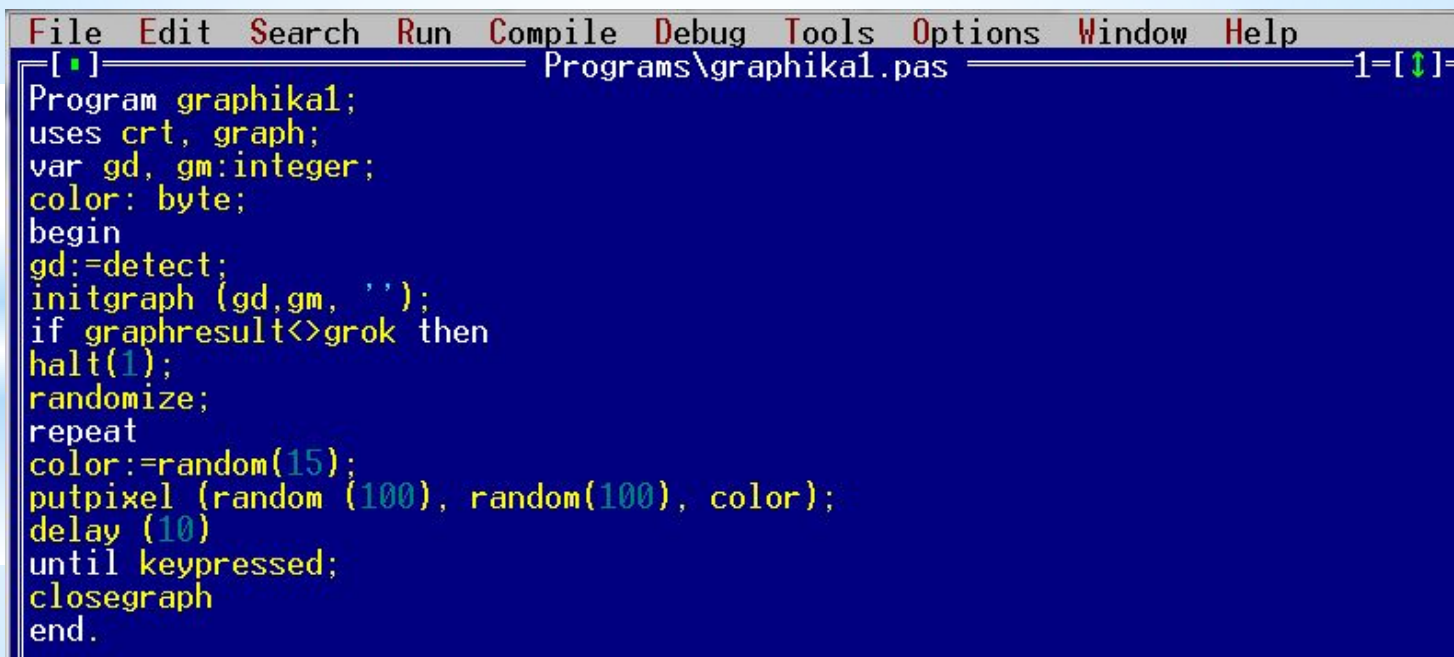
Широкое распространение в школах в качестве обучающего языка получил язык Бейсик, позволяющий взаимодействовать с ЭВМ в режиме непосредственного диалога. Спустя много лет после изобретения Бейсика, он и сегодня самый простой для освоения из десятков языков общецелевого программирования.

```
10 dim A(5)
20 for i=1 to 5
30 input A(i)
40 next i
50 if i=5 then goto 140
60 if A(i)<=A(i+1) then goto 90
70 i=i+1
80 goto 130
90 z=A(i)
100 A(i)=A(i+1)
110 A(i+1)=z
120 i=1
130 goto 50
140 for i=1 to 5
150 print A(i)
160 next i
```


Pascal

70-е гг. произвели на свет Pascal. Он быстро завоевал массовую популярность.

Никлаус Вирт создал язык Паскаль, достаточно простой, удобный, с наличием мощных средств структурирования данных. Хотя Паскаль был разработан как язык для обучения программированию, он впоследствии получил широкое развитие и в настоящее время считается одним из самых используемых языков.

A screenshot of a Pascal IDE window. The title bar reads 'File Edit Search Run Compile Debug Tools Options Window Help' and 'Programs\graphikal.pas'. The code is as follows:

```
Program graphikal;
uses crt, graph;
var gd, gm:integer;
color: byte;
begin
gd:=detect;
initgraph (gd,gm, '');
if graphresult<>grok then
halt(1);
randomize;
repeat
color:=random(15);
putpixel (random (100), random(100), color);
delay (10)
until keypressed;
closegraph
end.
```

Язык СИ

Необходимость разработки больших программ, управляющих работой ЭВМ, потребовала создания специального языка программирования СИ в начале 70-х г. Он является одним из универсальных языков программирования. В отличие от Паскаля, в нем заложены возможности непосредственного обращения к некоторым машинным командам и к определенным участкам памяти компьютера. Си широко используется как инструментальный язык для разработки операционных систем, трансляторов, баз данных и других системных и прикладных программ. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Во многих случаях программы, написанные на Си, сравнимы по скорости с программами, написанными на языке Ассемблера. При этом они имеют лучшую наглядность и их более просто сопровождать. Си сочетает эффективность и мощность в относительно малом по размеру языке. На основе этого языка были разработаны современные языки Java и C++.

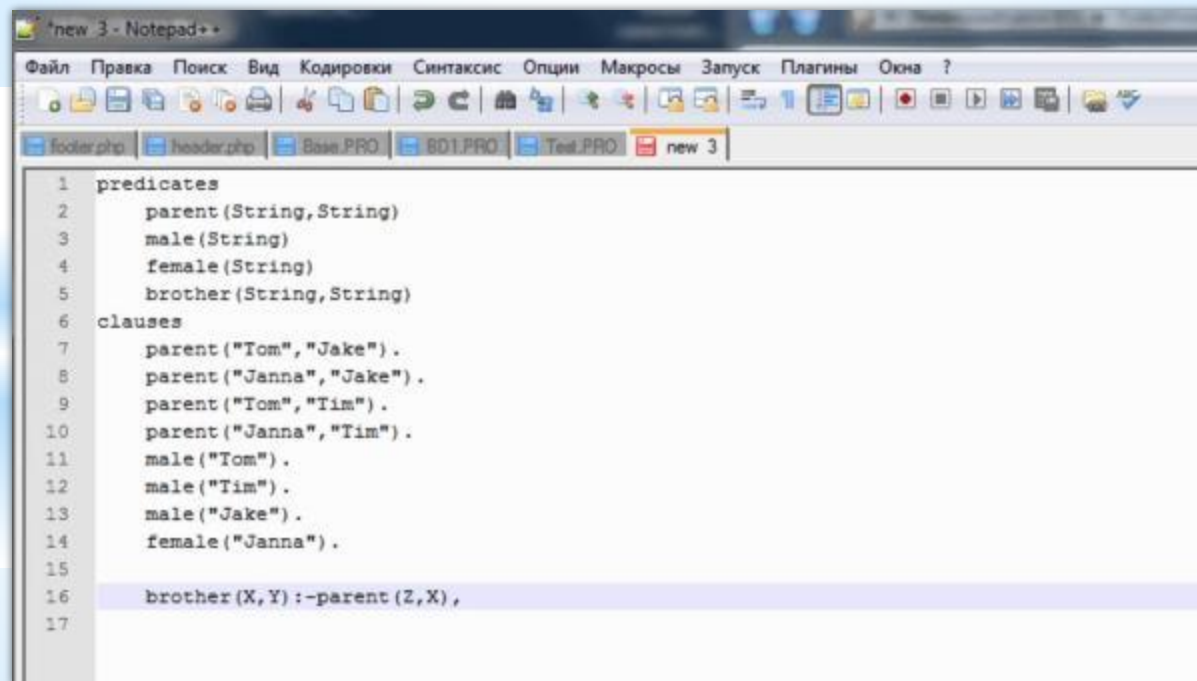
* Пример программы на языке СИ

```
#include <iostream.h>
#include <math.h>
void main()
{float a,b,c,p,s;
cout<<"\na="; cin>>a;
cout<<"\nb="; cin>>b;
cout<<"\nc="; cin>>c;
p=(a+b+c)/2;
s=sqrt(p*(p-a)*(p-b)*(p-c));
cout<<"\nПлощадь треугольника="<<s;
}
```


Другие языки

программирования

Помимо языков, работающих с алгоритмами, свое развитие получили и языки другой направленности. Для обработки деловой информации был разработан COBOL, языками искусственного интеллекта считаются Prolog и LISP. Кроме того, последний широко применяется и в теории игр. С переходом на персональные компьютеры, языки программирования превратились в отдельную часть сред разработки. Сейчас существуют даже языки, применяемые в офисных программах - например, VBA.



```
new 3 - Notepad++
Файл  Правка  Поиск  Вид  Кодировки  Синтаксис  Опции  Макросы  Запуск  Плагины  Окна  ?
[Icons]
fodier.php  header.php  Base.PRO  BD1.PRO  Test.PRO  new 3
1  predicates
2      parent (String,String)
3      male (String)
4      female (String)
5      brother (String,String)
6  clauses
7      parent ("Tom","Jake").
8      parent ("Janna","Jake").
9      parent ("Tom","Tim").
10     parent ("Janna","Tim").
11     male ("Tom").
12     male ("Tim").
13     male ("Jake").
14     female ("Janna").
15
16     brother (X,Y):-parent (Z,X),
17
```