

Лекция:

«Этапы развития языков программирования»



Язык программирования -
совокупность основных
символов (алфавит) и правил
составления из них смысловых
конструкций, предназначенная
для создания программных
средств.

Языки низкого уровня:

- машинный код
- ассемблеры

Машинный код

СИСТЕМА КОМАНД ЭВМ-220

Операции над числами (Р, Q, T — порядки чисел X, Y, Z по A', A'', A''')					Логические операции (F, G, S — значения кодов по A', A'', A''')					Операции сдвига					Команды передачи управления с изменением [РА]						
Операция	КОП	Результат	ω=1	t, мксек	Операция	КОП	Результат	ω=1	t, мксек	Операция	КОП	Величина сдвига	ω=1	t, мксек	КОП в РА	Передача управл. в RA	Передача управл. в KPA+1	t, мксек			
Сложение	01 21 41 61				Сравнение	15	$C_k = F_k - T_k $			Сдвиг кода по A'	54	$C_{\text{сдв}} = P_k \cdot S \cdot 2^{P-100}$ но 027 при C=0	C=0	24/65	11	$PA < A' \text{ и } \omega=1$	$PA > A' \text{ или } \omega=0$				
Вычитание	02 22 42 62		Z=0	285	Сравнение с ост.	35	аналогично 15, но 027 при C=0	C=0	24	Сдвиг кода по P	74	$C_{\text{сдв}} = P_k \cdot S \cdot 2^{P-100}$	C=0	24/65	31	$PA > A' \text{ и } \omega=1$	$PA < A' \text{ или } \omega=0$				
Вычитание модулей	03 23 43 63				Логическое умножение	55	$C_k = F_k \cdot T_k$			Сдвиг мантиссы по A'	14	$C_{\text{сдв}} = P_k \cdot S \cdot A' \cdot 2^{P-100}$	C=0	24/65	51	$PA < A' \text{ и } \omega=0$	$PA > A' \text{ или } \omega=1$				
Деление	04 24 - -			107	Логическое сложение	75	$C_k = F_k \vee T_k$			Сдвиг мантиссы по P	34	$C_{\text{сдв}} = P_k \cdot S \cdot 2^{P-100}$	C=0	24/65	71	$PA > A' \text{ и } \omega=0$	$PA < A' \text{ или } \omega=1$				
Умножение	05 25 45 65		Y=100	51	Специальные операции над кодами					Операции засылки					12	$PA < A'$	$PA > A'$	24			
Извлечение кв. корня	44 64 - -			272	Цикл сложения	07	$C' = (F' + T') \cdot 2^{P-100}$ $C' = (F' + T') \cdot 2^{P-100}$			Операция	КОП	Результат	ω=1	t, мксек	32	$PA > A'$	$PA < A'$				
Выбор младших разрядов	47			24	Цикл вычитания	27	$C' = (F' - T') \cdot 2^{P-100}$ $C' = (F' - T') \cdot 2^{P-100}$			Выборка из динамики	17	Засылка кода [Z] → Y у-ячейка по A'			40	$PA > A' \text{ зона } A_2[A']$	$PA < A' \text{ зона } A_2[A']$				
Операции изменения порядка					Сложение мантисс	13	$C' = F' + T'$			Засылка в динамику	37	Засылка кода [Y] → Z у-ячейка по A'	C=0	90	60	$PA > A' \text{ зона } A_2[A']$	$PA < A' \text{ зона } A_2[A']$				
Операция	КОП	Результат	ω=1	t, мксек	Вычитание мантисс	33	$C' = F' - T'$			Засылка кода	00	A' безразличен			16	Команды передачи управления с занесением в A'					
Сложение порядка с A'	06	$G = q + (A' - 100)$			Сложение КОПов	53	$C' = (F' + T') \cdot 2^{P-100}$			Засылка кода с КЗУ	20	[KЗУ] → A' Эта разряда A' и КЗУ			56	КОП	Код, заносимый в A'	Передача управления в A'	t, мксек		
Сложение порядка с P	26	$G = q + (P - 100)$	G=100	24	Вычитание КОПов	73	$C' = (F' - T') \cdot 2^{P-100}$			Анализ окончания раб. АМ	20	A' = 0000 — переход к прир. 0 Трасс. АМ			36	0160000 A' 0000	Всегда	—			
Вычитание порядка из A'	46	$G = q - (A' - 100)$			Цикл сдвига	67	$C' = F' \cdot 2^{P-100}$			Ма (КОП 50)					76	[A']	ω=1	ω=0	24		
Вычитание порядка из P	66	$G = q - (P - 100)$			Операции в М2, выполняемые по сигналам из М1					М5 (КОП 70)											
Команда останова					Операции изменения [РА]					Режим											
КОП	Совместные операции				КОП	Код в РА	Код в A'		t, мксек	УЧ	12 11 10 9 8 7 6 5 4 3 2 1	A'	A'	t, мксек	A'	A'	A'				
77	Вызов [A'] и [A''] на P и P ₂ гашение [A'] при продолжении				52	A'	0520000 A' 0000		24	Обмен с МБ	0010	x	0 0 0 0 1 1/2								
Операции изменения [РА]					72	Зона A' [A']	0520000 A' 0000			Обмен с МЛ	0020	0	0 0 1 0 0 0 x x x								
Команды ввода					Операции в М2, выполняемые по сигналам из М1					Печать „8“											
Операция	КОП	A'	A''	A'''	Обмен командами в М2 по адресу из М1	AP2	40÷38р 34р 24÷13р			Печать „10“	0100	x	0 0 1 0 0 0 x x x								
Ввод с останова при несоблюдении КС	10	A _н МЗУ	КРА	АКС	Обмен кодами М1=М2	AP1	40÷38р 36÷25р 24÷13р			Печать A _н Ц	0140	x	0 0 1 0 0 0 x x x								
Ввод без ост. при несоблюд. КС	30	A _н МЗУ	КРА	АКС	Операции, выполняемые по сигналам от линии связи с аналоговой машины					Перфорация	0200	x	x 1 0 0 0 0 x x x								
Операции изменения регистров приращения					Операция					Накопление	0300	x	x 1 1 1 0 0 x x x								
КОП	A'	A''	код заносимый в A'	t, мксек	Автоматизм от ЛС	APAC	Выборка команды из яч. ки 0022 „0“ МЗУ без занесения 0022 на КРА			Разметка МЛ	0040	0	x 0 0 1 0 0 x								
21р-РПМЛ					Автоматизм от АМ	APAM	Выборка команды из яч. ки 0023 „0“ МЗУ без занесения 0023 на КРА			Подвод МЛ	0060	x	x 0 0 1 1 0 x								
54р-РПМЕ	321р-РПА3				Операции изменения [РА]					Засылка состояния	AP3	0030	РПА2 РПА3 x 1 1 0 0 0								
7р-Пр АЗ	634р-РПА2				КОП					Возврат состояния	AP3	0034	РПА2 0 0 0 x 1 1 1 0 0								
8р-Пр А2	987р-РПА1				Автоматизм от ЛС					Выборка кода в М2 адреса из М1	AP2	0033	РПА2 x x x 1 1 1 x 1								
9р-Пр А1	2144р-РПАМ				Автоматизм от АМ					Адреса М1=М2	AP1	0031	РПА2 1 1 1 1/2 0 1								
10р-Пр КРА					Автоматизм от ЛС					Работа с АМ	AM	0032	x x x x 1 1 1 1/2 1 0								
11р-Пр МЛ					Автоматизм от ЛС					Заполняются только по команде Ма											
12р-Пр М5					Автоматизм от ЛС					Заполняются только по команде Ма											

1. Емкость промежуточного накопителя выбрала пив 1024 слова.
2. После вывода содержимое ПНВ не стирается.
3. При обмене с МБ возможен переход с данного барабана на следующий (4096—4097).
4. Команда останова имеет код только 77.
5. Команда 20 при A' = 0000 используется как команда перехода по признаку работы с аналоговой машины.
6. В МЛ M3 = 0 используется только для записи дефектных зон МЛ.

- ***
1. При чтении адреса с ПК 1-й РБФ-блокировка АВ ост. при несоблюдении КС.
 2. При A' = 0 блокируется запись в МЗУ последующего массива.
 3. При A' = 0 блокируется запись в МЗУ последующего массива.

- ***
1. При чтении адреса с ПК 1-й РБФ-блокировка АВ ост. при несоблюдении КС.
 2. При A' = 0 блокируется запись в МЗУ последующего массива.
 3. При A' = 0 блокируется запись в МЗУ последующего массива.

- ***
1. При чтении адреса с ПК 1-й РБФ-блокировка АВ ост. при несоблюдении КС.
 2. При A' = 0 блокируется запись в МЗУ последующего массива.
 3. При A' = 0 блокируется запись в МЗУ последующего массива.

Примечание: Знак „х“ означает, что состояние разряда безразлично. Отсутствие знака — означает, что в разряде может быть „0“ или „1“; A' — адрес исполнительный; [A'] — содержимое ячейки с адресом A.

Ассемблеры

Команда MIP	Интерпретация ко- манды MIP в мнемо- нических машинных кодах	Содержательная (интуитивная) операционная семантика мнемо- нических машинных кодов
Z(n)	MOV AL,00 MOV [адрес=R _n],AL	Пересылка в регистр AL нуля. Пересылка в ячейку памяти содер- жимого регистра AL.
S(n)	MOV AL,[адрес=R _n] INC AL MOV [адрес=R _n],AL	Пересылка в регистр AL содержи- мого ячейки памяти. Увеличение регистра AL на 1. Пересылка в ячейку памяти содер- жимого регистра AL.
T(m,n)	MOV AL,[адрес=R _m] MOV [адрес=R _n],AL	Пересылка в регистр AL содержи- мого ячейки памяти. Пересылка в ячейку памяти содер- жимого регистра AL.

```

.MODEL TINY
CODE SEGMENT
ASSUME CS:CODE, DS:CODE
ORG 100h
START:

    mov ah,9
    mov dx,OFFSET Msg
    int 21h
    int 20h
    Msg DB 'Hello World',13,10,'$'

CODE ENDS
END START

```

ЯЗЫКИ ВЫСОКОГО УРОВНЯ

- проблемно-ориентированные (Фортран, Алгол, Кобол, Ада, 1С и др.);
- универсальные (Паскаль, Basic, C++, C# и др.)
- языки разметки (HTML, XML и др.);
- языки описания сценариев (JavaScript и др.);
- языки моделирования систем (GPSS и др.)

Fortran (Фортран)

1954-1957 гг., IBM (Джон Бэкус)



```

C IF (N1) 1,2,3 between:
C spring near label 1 else N = 0.
C spring near label 2 else N = 5
C spring near label 3 else N = 9

      IF (100)10,10,20
20 IF (IX-100)12,21,12
21 IF (IY-100)12,22,12
22 IF (IS-100)12,23,12
23 CALL PLALC(AM,MMX,IPC)
      GO TO 73
10 IF (IX2)13,50,13
13 IF (IX2-MT)19,19,50
19 IF (IY2)11,50,11
11 IF (IY2-MT)12,12,50
12 IF (TIND(IX2,IY2,1)) GO TO 73
      IF (CV-AM(IX2,IY2))206,206,5
206 IF (IDX**2+IDY**2-1)213,6,213
213 DCP=(AM(IX,IY)+AM(IX2,IY)+
      + AM(IX,IY2)+AM(IX2,IY2))/4.0
      IF (DCP-CV)5,217,217
217 IF (INX(IS-1))214,215,214
214 IX=IX+IDX
C .....
```

Algol (Алгол)

1958-1960 гг., *IFIP*

(Международная федерация по
обработке информации)

```
BEGIN
FILE F (KIND=REMOTE);
EBCDIC ARRAY E [0:11];
REPLACE E BY "HELLO, WORLD!";
WHILE TRUE DO
  BEGIN
    WRITE (F, *, E);
  END;
END.
```

```
procedure Absmax(a) Size:(n, m) Result:(y) Subscripts:(i, k);
  value n, m; array a; integer n, m, i, k; real y;
comment Максимальный элемент матрицы a, размера n на m
  передаётся в виде результата в y, а его индексы — в параметры i и k;
begin integer p, q;
  y := 0; i := k := 1;
  for p:=1 step 1 until n do
    for q:=1 step 1 until m do
      if abs(a[p, q]) > y then
        begin y := abs(a[p, q]);
          i := p; k := q
        end
      end
    end Absmax
```


Cobol (Кобол)

1959 г., Грейс Хоппер



```
IDENTIFICATION DIVISION.  
  PROGRAM-ID. HELLO-WORLD.  
*  
  ENVIRONMENT DIVISION.  
*  
  DATA DIVISION.  
*  
  PROCEDURE DIVISION.  
    PARA-1.  
      DISPLAY "Hello, world."  
*  
      EXIT PROGRAM.  
  END PROGRAM HELLO-WORLD.
```

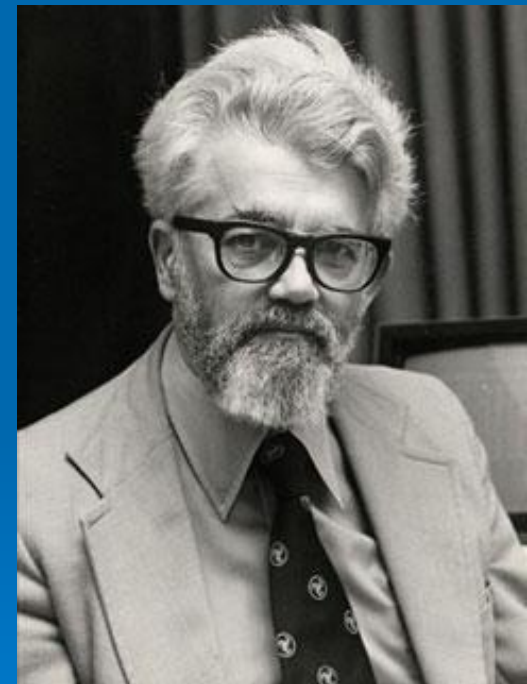
PL/1 (ПЛ/1)

1964 г., IBM

```
Test: proc options (main);  
      put list ('Hello, world!');  
end;
```

LISP (Лисп)

1961-1965 гг., Джон Маккарти



Prolog (Пролог)

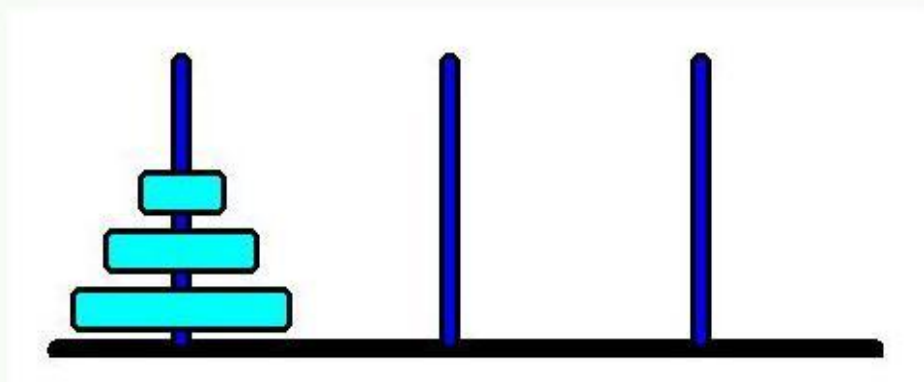
1971-1972 гг., Alain Colmerauer



Задача «Ханойская башня»

"Где-то в непроходимых джунглях, недалеко от города Ханоя, есть монастырь бога Браммы. В начале времен, когда Брами создавал Мир, он воздвиг в этом монастыре три высоких алмазных стержня и на один из них возложил 64 диска, сделанных из чистого золота. Он приказал монахам перенести эту башню на другой стержень (в соответствии с правилами, разумеется). С этого времени монахи работают день и ночь. Когда они закончат свой труд, наступит конец света."

Правила перемещения дисков таковы: разрешается снимать со стержня только верхний диск, запрещается класть больший диск на меньший, при каждом ходе передвигается только один диск.



Решение задачи «Ханойская башня» на Прологе (для трех дисков)

```
% move(число_дисков, откуда, куда, через)
move(1,X,Y,_) :- write('Move top disk from '),
                 write(X), write(' to '),
                 write(Y), nl.

move(N,X,Y,Z) :- N>1, M is N-1,
                 move(M,X,Z,Y),
                 move(1,X,Y,_),
                 move(M,Z,Y,X).
```

Файл Сеансы Настройки Помощь

?- [hanoi].

% hanoi compiled 0.00 sec, 1,268 bytes

Yes

?- move(3,left,right,center).

Move top disk from left to right

Move top disk from left to center

Move top disk from right to center

Move top disk from left to right

Move top disk from center to left

Move top disk from center to right

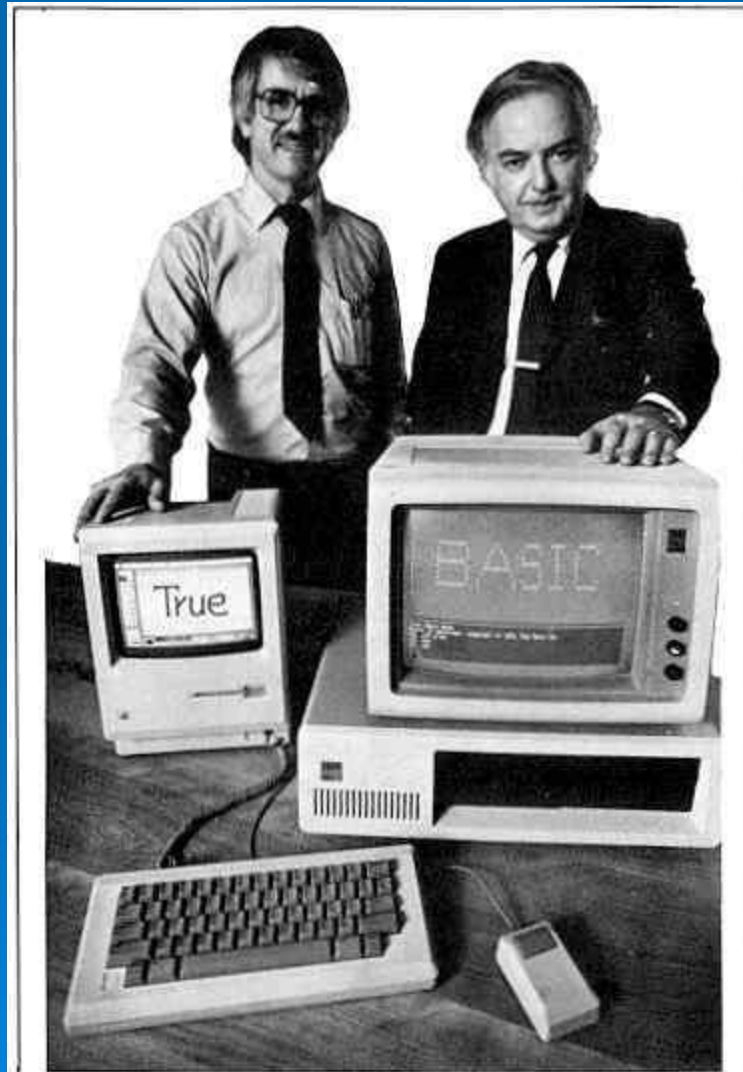
Move top disk from left to right

Yes

?- █

BASIC (Бейсик)

1965 г., Т. Куртс и Дж. Кемени



Thomas Kurtz (left) and John Kemeny with the latest implementation of their BASIC programming language.

Симула-67

конец 1960-х гг.,
Кристен Ньюгор и Уле-Юхан Даль



Pascal (Паскаль)

1970 г, Никлаус Вирт



Ada (Ада)

1975-1980 гг., Жан Ишбиа

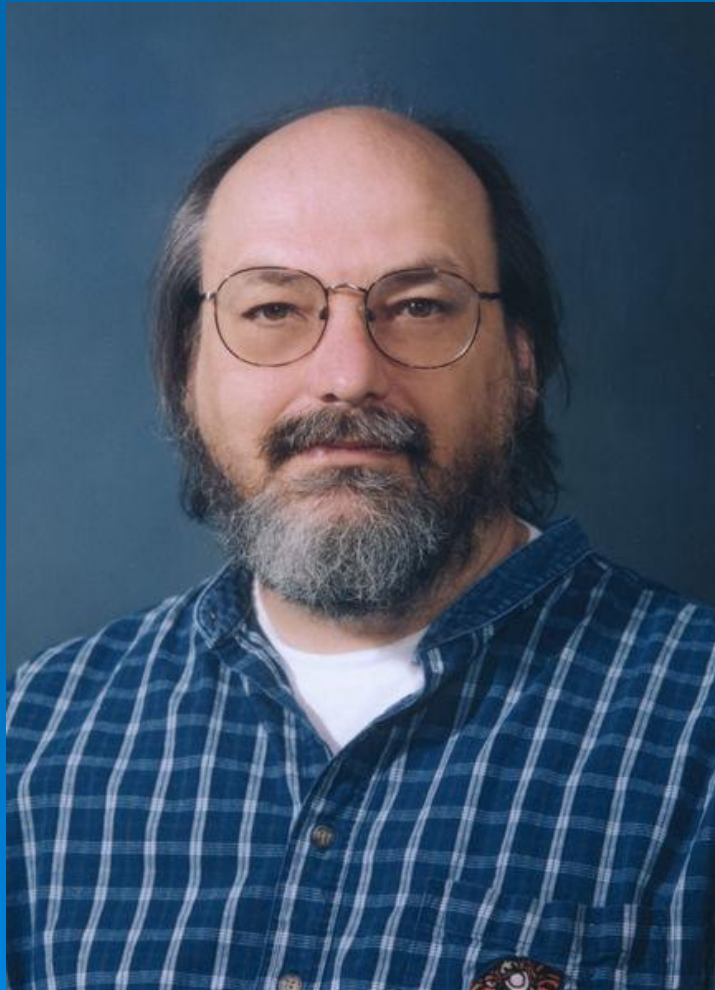
```
with Ada.Text_IO;  
  
procedure Hello is  
  use Ada.Text_IO;  
begin  
  Put_Line("Hello, world!");  
end Hello;
```


Modula-2 (Модула-2)
1979 г., Никлаус Вирт

Oberon (Оберон)
1987 г., Никлаус Вирт

A decorative graphic consisting of several sets of concentric circles, resembling ripples in water, located in the bottom right corner of the slide. The circles are light blue and vary in size and opacity.

С (Си)
1972 г.,
Кен Томпсон, Денис Ритчи



C++ (*Cu++*)

1983 г., Бьерн Страуструп



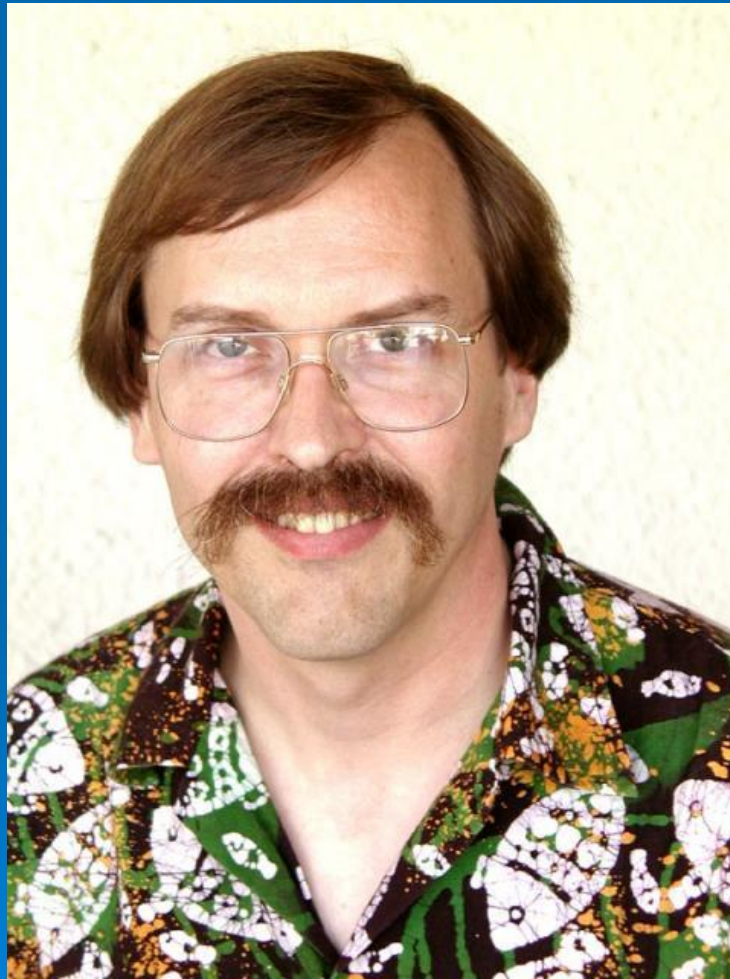
Системы программирования (наиболее распространенные):

- ▣ ***Visual Basic*** (1991-1993 гг., Microsoft)
- ▣ ***Delphi*** (1995 г., Borland)
- ▣ ***Visual C++*** (1995 г., Microsoft)
- ▣ ***Visual Studio*** (1997 г., Microsoft)
- ▣ ***Visual Studio .NET*** (2002 г., Microsoft)
- ▣ ***Eclipse*** (2004 г., Eclipse Foundation)

и др.

Perl (Перл)

1987 г., Ларри Уолл



```
perl -wle '(1 x $_) !~ /^(11+)\1+$/ && print while ++ $_'
```


Java (Джава)

1991-1995 г., Sun Microsystems

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

LiveScript → *JavaScript*

1995 г., Netscape (Брендон Айх)



```
<script type="text/javascript">  
alert('Hello, World!');  
</script>
```

HTML – HyperText Markup Language 1991-1992 гг., Тим Бернерс-Ли



```
<b>  
    Этот текст будет жирным,  
    <i>а этот - ещё и курсивным</i>  
</b>
```

SQL – Structured Query Language

1986 г.,

IBM (Donald D. Chamberlin, Ray Boyce, Pat Selinger, Raymond Lorie)

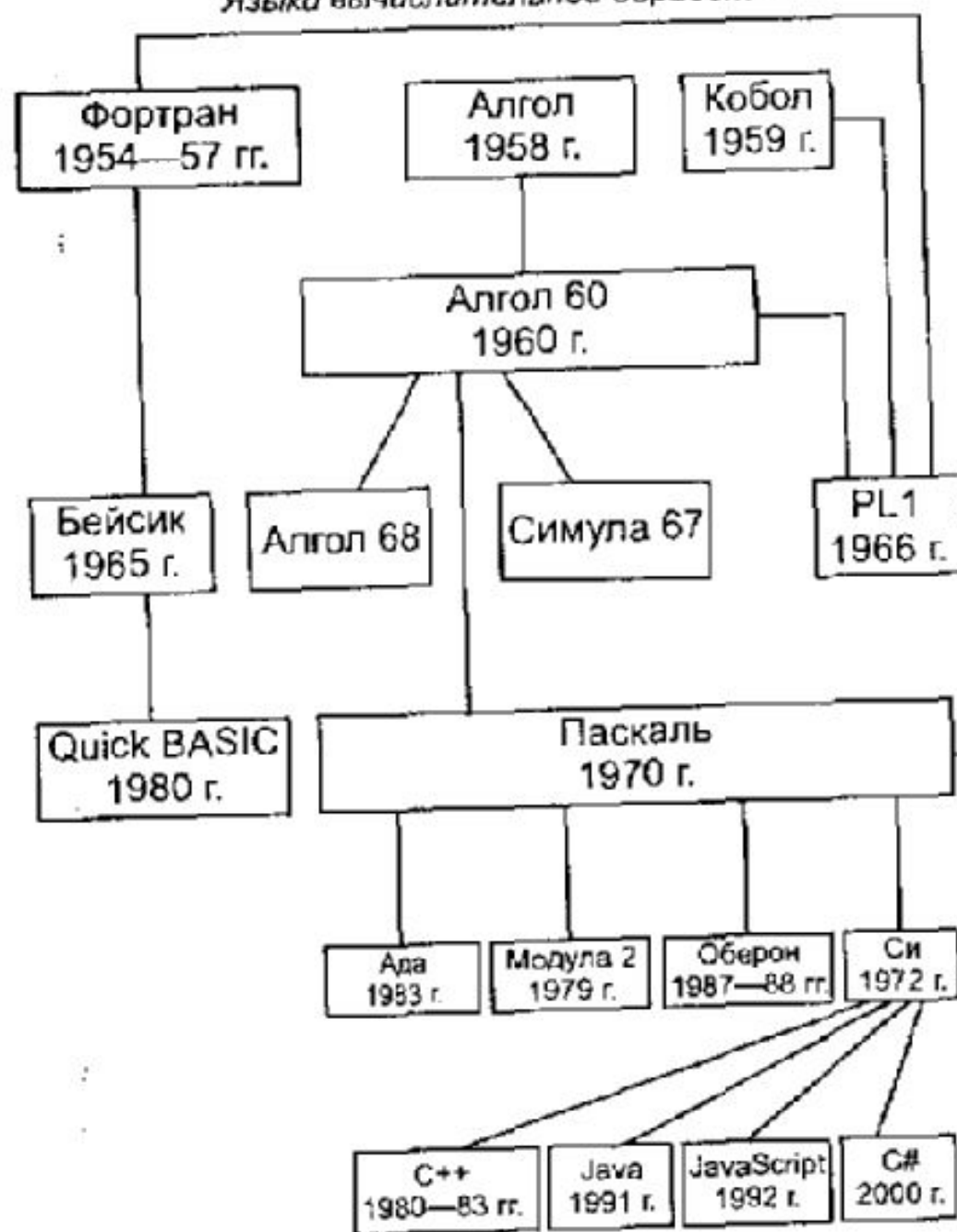
```
SELECT
    ...
FROM
    table_1,
    table_2,
    table_3
WHERE
    ...
GROUP BY
    ...
HAVING
    ...
ORDER BY
    ...
```

С# (Си-шарп)

1998-2001 гг.,
Microsoft (Андерс Хейлсберг)



Языки вычислительной обработки



Языки символьной обработки

