


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 (П.9.1-порядки чисел X,Y,Z по A', A'', A''')					Логические операции (Ф.9.5-значения кодов по A', A'', A''')					Операции сдвига					Команды передачи управления с изменением [РА]					
Операция	КОП	Результат	$\omega-1$	$t$ миксек	Операция	КОП	Результат	$\omega-1$	$t$ миксек	Операция	КОП	Величина сдвига	$\omega-1$	$t$ миксек	КОП	Код в РА	Передача управл. в A'	Передача управл. в КРА+1	$t$ миксек	
Сложение	01 21 41 61				Сравнение	15	$C_k =  F_k - P_k $			Сдвиг кода по A'	54	$C_{\text{сдв}} = P_k \cdot S - P_{k-1} \cdot 100$	C=0	24/155	11		РА < A', $\omega=1$	РА > A', или $\omega=0$		
Вычитание	02 22 42 62		Z<0	285	Сравнение с ост.	35	аналогично 15, но ОЕТ при C=0		24	Сдвиг кода по P	74	$C_{\text{сдв}} = P_k \cdot S - P_{k-1} \cdot 100$	C=0	24/155	31		РА > A', $\omega=1$	РА < A', или $\omega=0$		
Вычитание модулей	03 23 43 63				Логическое умножение	55	$C_k = F_k \cdot P_k$			Сдвиг мантииссы	14	$C_{\text{сдв}} = P_k \cdot S - A_{k-1} \cdot 100$	C=0	24/153	51		РА < A', $\omega=0$	РА > A', или $\omega=1$		
Деление	04 24 - -			107	Логическое сложение	75	$C_k = F_k \vee P_k$			Сдвиг мантииссы по P	34	$C_{\text{сдв}} = P_k \cdot S - P_{k-1} \cdot 100$	C=0	24/155	71		РА > A', $\omega=0$	РА < A', или $\omega=1$		
Умножение	05 25 45 65		Y=100	51	Специальные операции над кодами					Операции засылки					Команды передачи управления с занесением в A'					
Извлечение кб. кода	44 64 - -			272	Цикл сложения	07	$C_k = (F_k + P_k) \bmod 2^8$			Операция	КОП	Результат	$\omega-1$	$t$ миксек	12		РА < A',	РА > A',		
Выбор младших разрядов	47			24	Цикл вычитания	27	$C_k = (F_k - P_k) \bmod 2^8$			Выборка из динамики	17	Засылка кода [Z] → Y у-ячейка по A'			32		РА > A',	РА < A',		
Операции изменения порядка					Сложение мантиис					Засылка в динамику					Команды передачи управления с занесением в A'					
Операция	КОП	Результат	$\omega-1$	$t$ миксек	Вычитание мантиис	33	$C_k = (F_k - P_k) \bmod 2^{26}$		24	Засылка кода [Y] → Z у-ячейка по A'	37		C=0	90	40		РА-зона A <sub>2</sub> [A <sub>1</sub> ]	РА > зона A <sub>2</sub> [A <sub>1</sub> ]		
Сложение порядка с A'	06	$\Gamma = q + (A_0^{(q)} - 100)$			Сложение КОПов	53	$C_k = (F_k + P_k) \bmod 2^9$			Засылка кода с КЗУ	20	[A <sub>1</sub> ] → A <sub>2</sub> A <sub>2</sub> безразличен			60		РА > зона A <sub>2</sub> [A <sub>1</sub> ]	РА < зона A <sub>2</sub> [A <sub>1</sub> ]		
Сложение порядка с P	26	$\Gamma = q + (P - 100)$		24	Вычитание КОПов	73	$C_k = (F_k - P_k) \bmod 2^9$			Анализ окончания раб АМ	20	[KЗУ] → A <sub>2</sub> Эле. разряда A <sub>2</sub> KЗУ		24	16	016000 A <sub>1</sub> 0000	всегда	-		
Вычитание порядка из A'	46	$\Gamma = q - (A_0^{(q)} - 100)$			Цикл сдвига	67	$C_k = (F_k \bmod 2^8) \bmod 2^8$			Ма (КОП 50)					М5 (КОП 70)					
Вычитание порядка из P	66	$\Gamma = q - (P - 100)$			Команда останова					Операции в M2, выполняемые по сигналам из M1					Операции, выполняемые по командам Ма и М5					
КОП					КОП					КОП					КОП					
77					77					77					77					
Вызов [A'] и [A''] на P <sub>1</sub> и P <sub>2</sub> гашение [A'] при продолжении					Информация переданная на РБФ М2 из M1					Режим					A <sub>2</sub>					
52					52					52					52					
72					72					72					72					
КОМАНДЫ ВВОДА					Обмен кодами M1 ↔ M2					Обмен с МБ					Обмен с МЛ					
10					10					10					10					
30					30					30					30					
КОМАНДЫ РЕГИСТРОВ ПРИРАЩЕНИЯ					Автораэрыб от ЛС					Автораэрыб от АМ					Засылка состояния					
57					57					57					57					
21р-РПМЛ					21р-РПМЛ					21р-РПМЛ					21р-РПМЛ					
54р-РПМЕ					54р-РПМЕ					54р-РПМЕ					54р-РПМЕ					
7р-Пр А3					7р-Пр А3					7р-Пр А3					7р-Пр А3					
8р-Пр А2					8р-Пр А2					8р-Пр А2					8р-Пр А2					
9р-Пр А1					9р-Пр А1					9р-Пр А1					9р-Пр А1					
10р-Пр КРА					10р-Пр КРА					10р-Пр КРА					10р-Пр КРА					
11р-Пр МЛ					11р-Пр МЛ					11р-Пр МЛ					11р-Пр МЛ					
12р-Пр М5					12р-Пр М5					12р-Пр М5					12р-Пр М5					

1. Емкость промежуточного накопителя вывода пнв 1024 слова
2. После вывода содержимое пнв не стирается
3. При адмене с М5 возможен переход с данного барабана на следующий (4096-4097)
4. Команда останова имеет код только 77
5. Команда 20 при A<sub>1</sub>=0000 используется как команда перехода по признаку работы с аналоговой машиной
6. В ИМЛ М3=0 используется только в режиме "Разметка" для записи дефектных зон МЛ.

- \*\*\*
1. При чтении адреса с ПК 1<sup>й</sup> тр РБФ-блокировка АВ ост при несоблюдении КΣ
  - 1<sup>й</sup> тр РБФ-блокировка записи в МЗУ последующего массива 37-38р РБФ(РПАА) → -13-14р СМ А
  2. При A<sub>1</sub>=0 блокируется запись в МЗУ последующего массива
- \*\*\*
- Обращение к МЗУ по команде ИРП производится по старым значениям ИРП МЗУ. ИРП изменяется только по командам ИРП, АР3В
- \* Операции АР1м2, АР2м2, АР3С, АР4М не изменяют состояния машины.

Примечание: Знак, X означает, что состояние разряда безразлично  
Отсутствие знак. 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 мемо- рических машинных кодах	Содержательная (интуитивная) операционная семантика мемо- рических машинных кодов
Z(n)	MOV AL,00 MOV [адрес=R <sub>n</sub> ],AL	Пересылка в регистр AL нуля. Пересылка в ячейку памяти содер- жимого регистра AL.
S(n)	MOV AL,[адрес=R <sub>n</sub> ]  INC AL MOV [адрес=R <sub>n</sub> ],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 <sub>m</sub> ]  MOV [адрес=R <sub>n</sub> ],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)



```
* IF (N1) 1,2,3 between
* spring near label 1 via N = 0.
* spring near label 2 via N = 0
* spring near label 3 via N = 0

      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(205,2000,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 г., 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
```



# *Собоl (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)

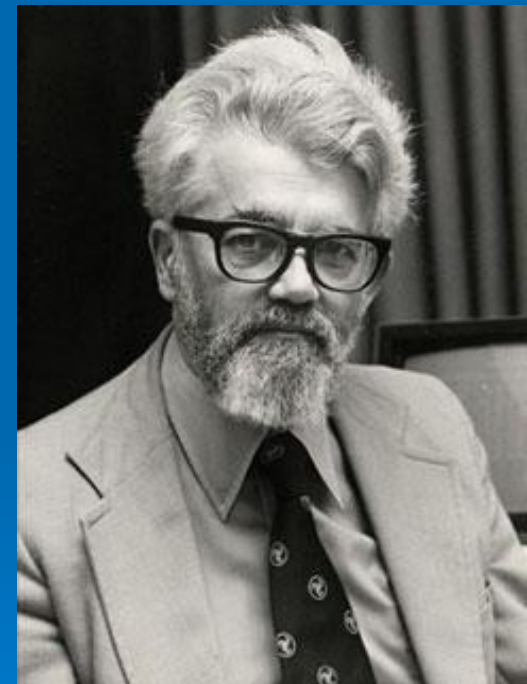
1961-1965 гг., Джон Маккарти



```
LispMe Symbols Scratch
(+ (* (+ x z) (* 3 (expt x 2)))) (expt
x 3))
3099.416-558.552i

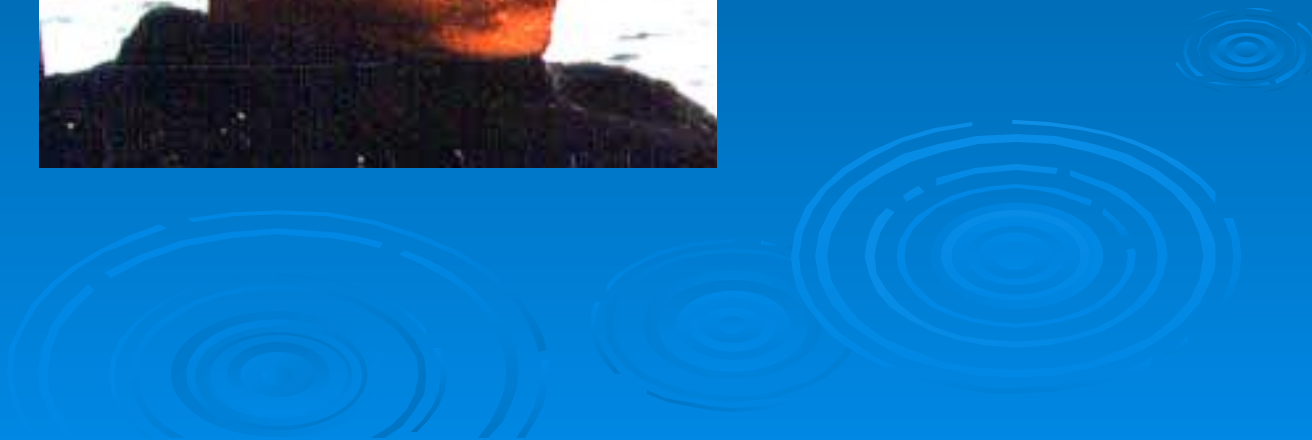
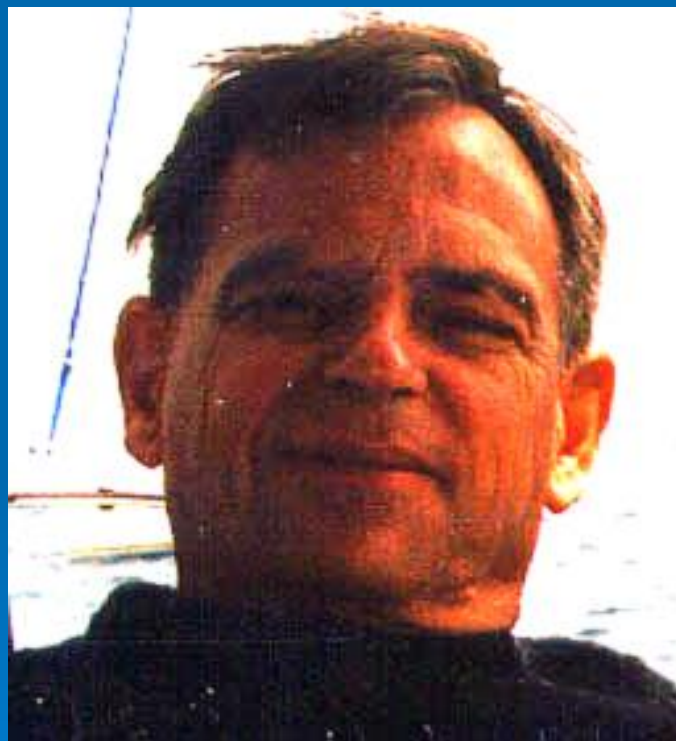
(let ((x 7.4) (z 9-3.4i))
  (eval (write (diff
    (* (+ x z) (expt x 3))))))
  )
```

Load Reload Pop Names Eval •



# *Prolog (Пролог)*

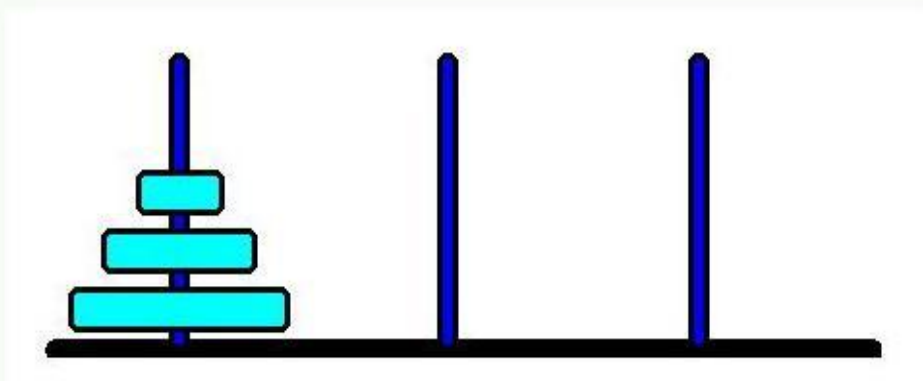
1971-1972 г., Alain Colmerauer



# 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

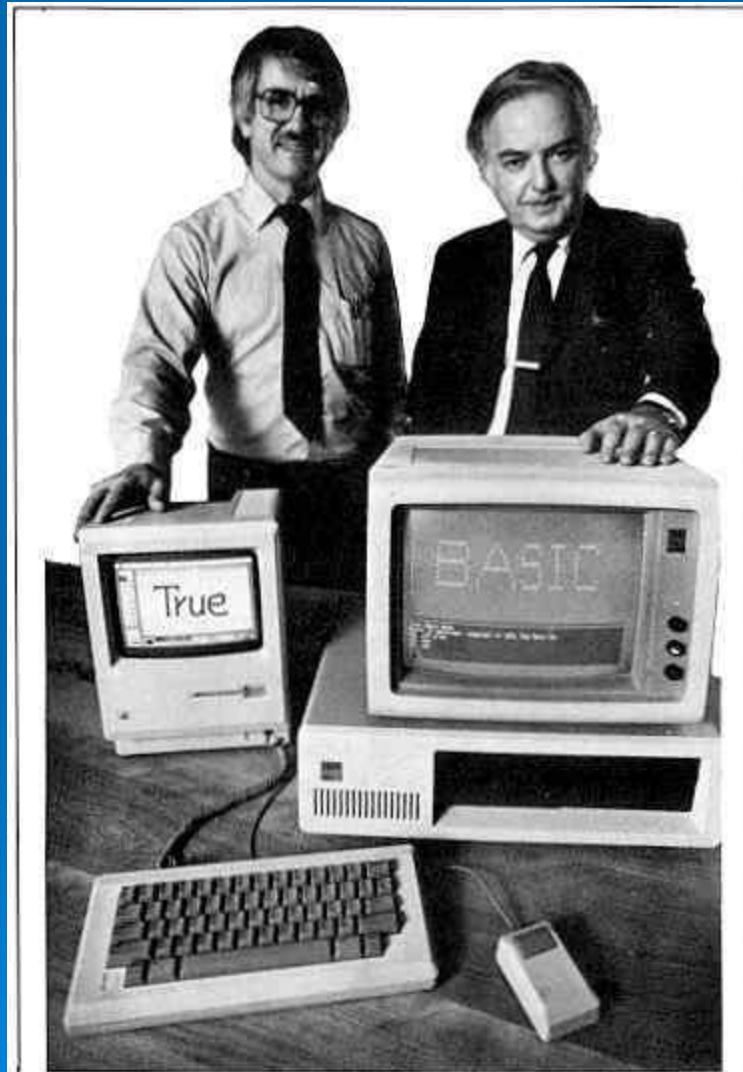
```
?- [hanoi].
% hanoi compiled 0.00 sec, 1,268 bytes

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

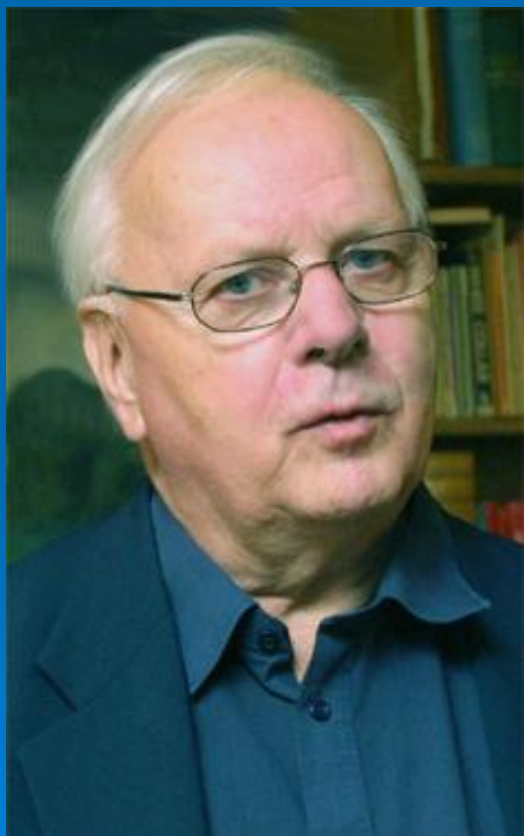


*Thomas Kurtz (left) and John Kemeny with the latest implementation of their BASIC programming language.*

# *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)***

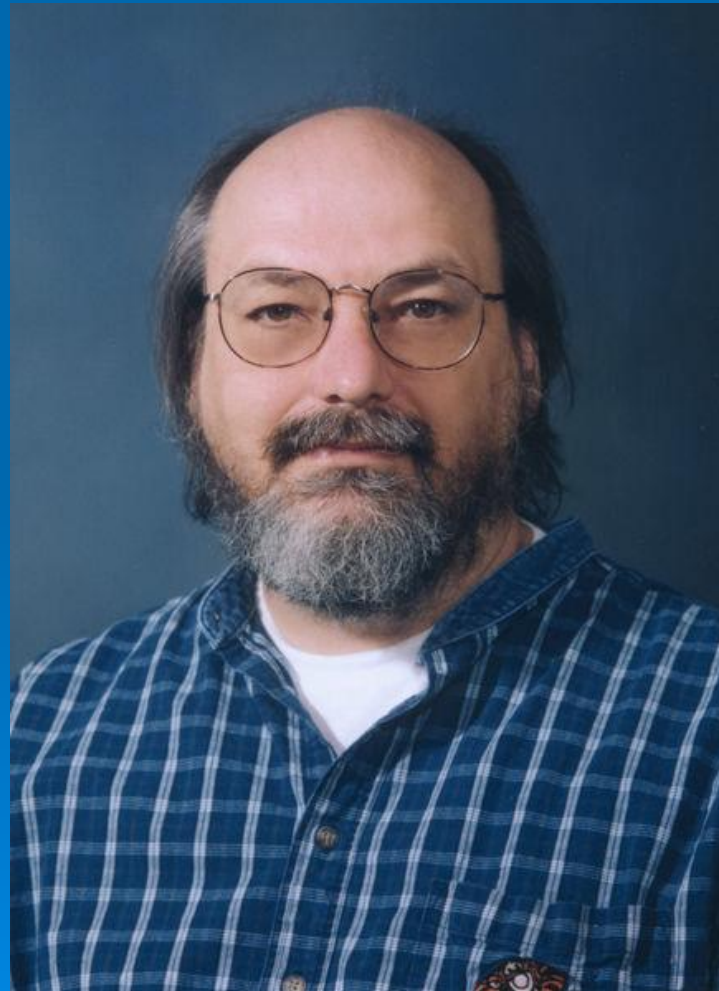
1987 г., Никлаус Вирт



C (C)u

1972 г.,

Кен Томпсон, Денис Ритчи



# C++ (C*u*++)

1983 г., Бьерн Страуструп



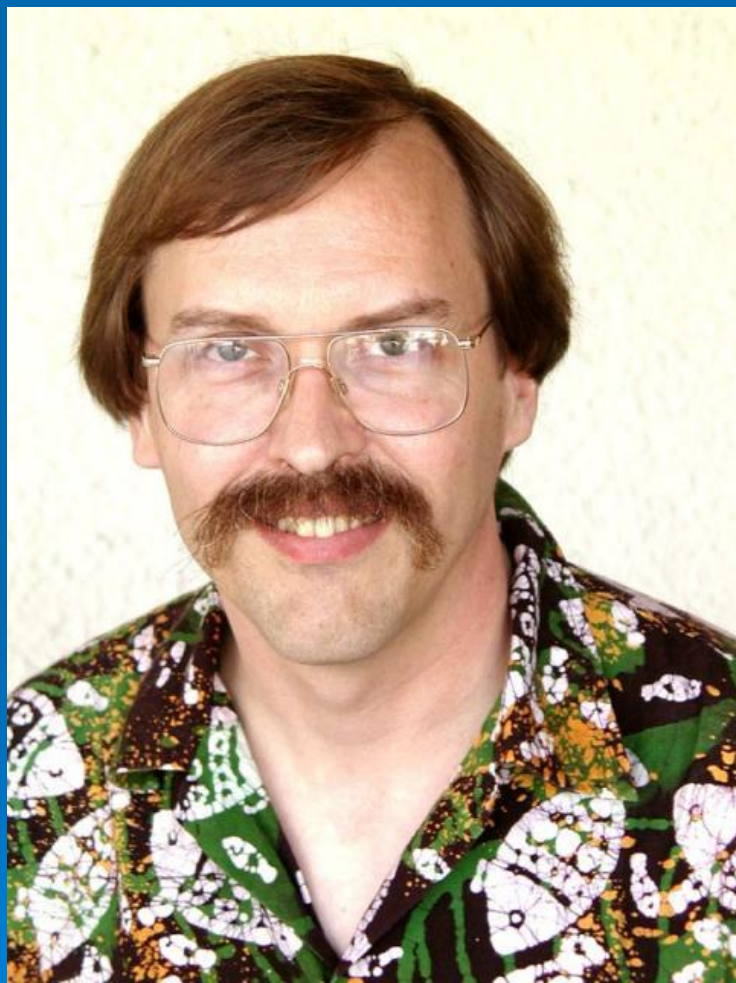
# Системы программирования (наиболее распространенные):

- ▣ ***Visual Basic*** (1991-1993 г., 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
    ...
```

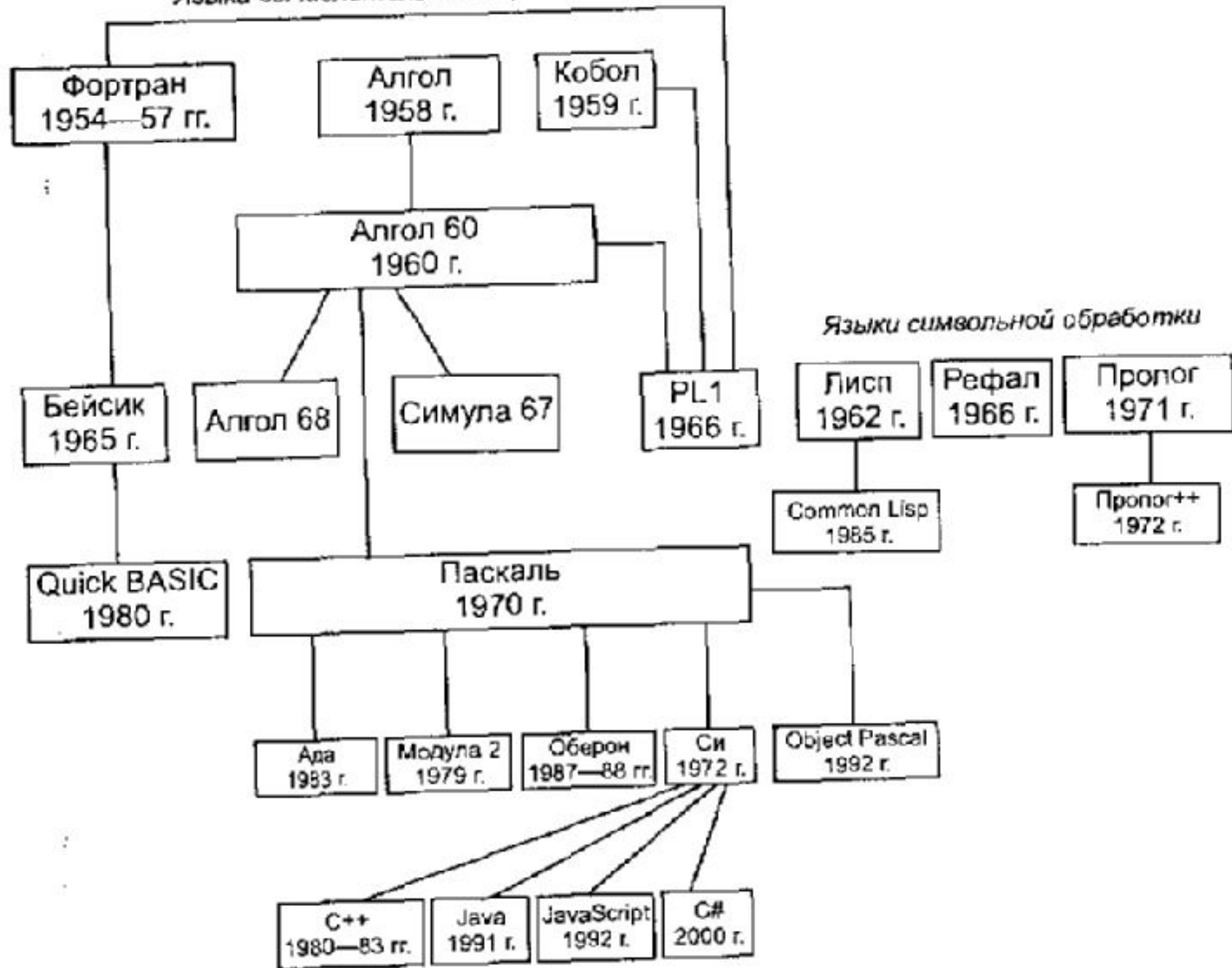
# С# (Си-шарп)

1998-2001 гг.,  
Microsoft (Андерс Хейлсберг)





*Языки вычислительной обработки*



*Языки символьной обработки*