

# ИСТОРИЯ ПРИКЛАДНОЙ МАТЕМАТИКИ ЭВМ

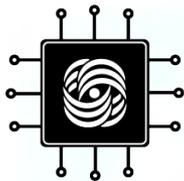
## Лекция 13:

## *Биографии основных теоретиков Computer science*

ВМиК МГУ им. М.В. Ломоносова, Кафедра АСВК

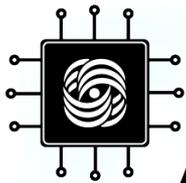
Доцент Власов В.К.,

Ассистент Волканов Д.Ю.



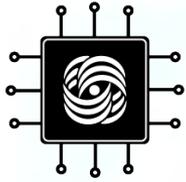
# План лекции

- Алан Тьюринг
- Джон Фон Нейман
- Эдсгер Вайб Дейкстра
- Дональд Эрвин Кнут
- ЯП и их создатели



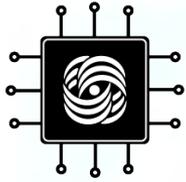
# Алан Тьюринг (1912 – 1954)





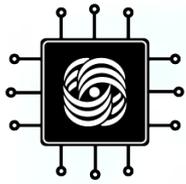
# Основные результаты

- Тезис Чёрча-Тьюринга
- Машина Тьюринга
- Криптография
- Тест Тьюринга
- Колоссус

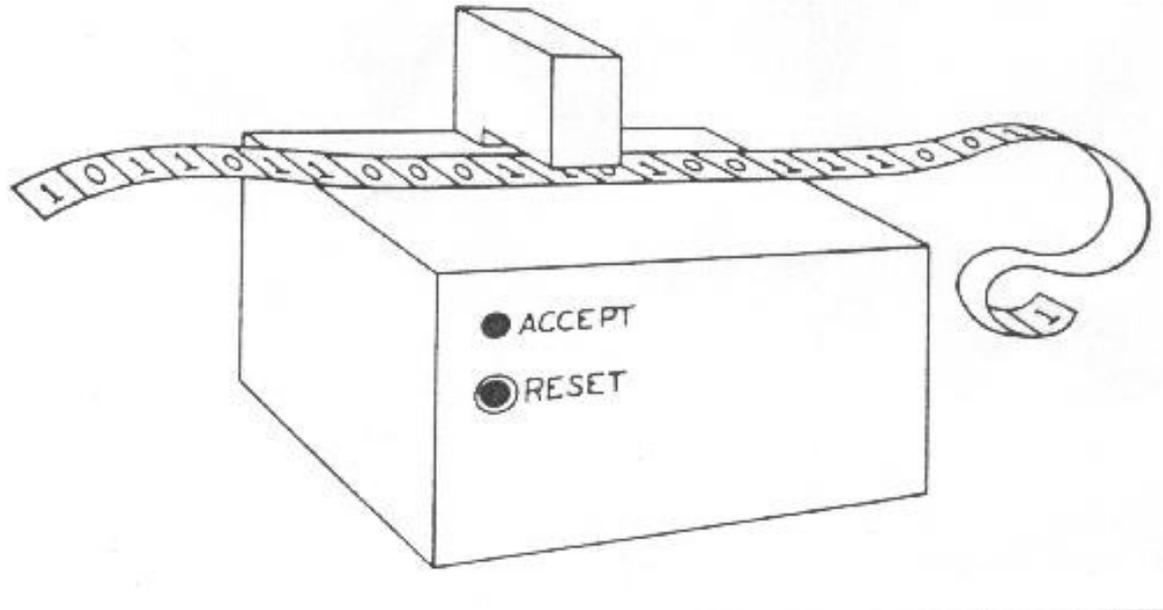


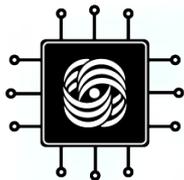
# Тезис Чёрча-Тьюринга

- Любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга



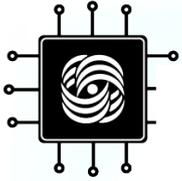
# Машина Тьюринга (1)





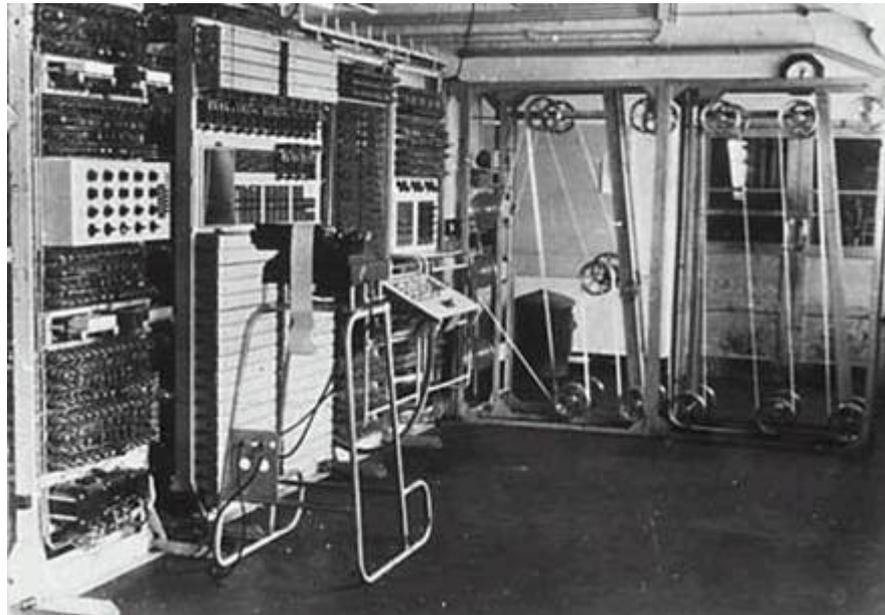
# Криптография

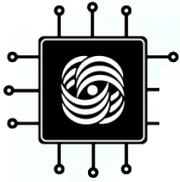
- Тьюринг помогал взломать код Энигмы
- Построен первый программируемый компьютер Колоссус
- Базировался:
  - на его концепции универсальной машины 1936
  - потенциальной скорости и надёжности электронных технологий
  - неэффективность разностных машин для различных логических процессов
- Шифр-код был расшифрован в 1943
- Все компьютеры были разрушены по приказу Черчилля



# Colossus

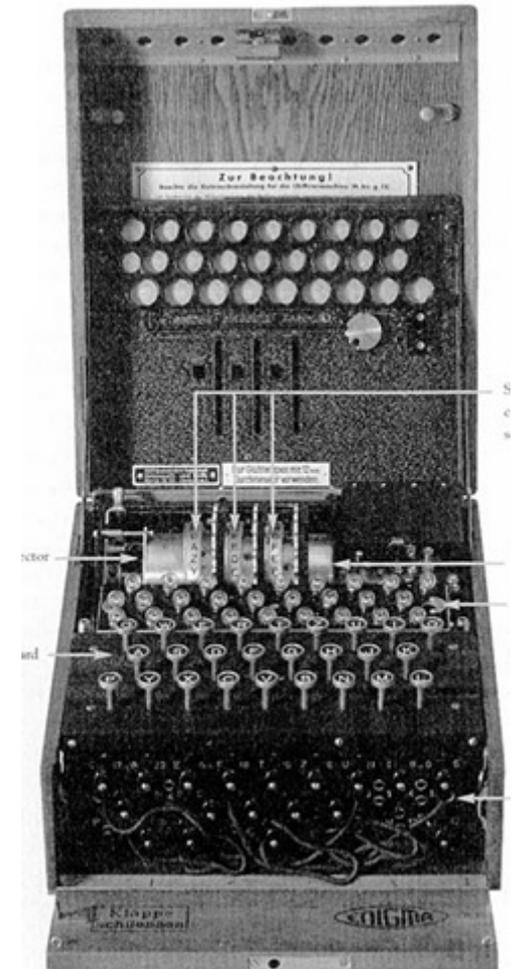
Под руководством выдающегося математика Алана Тьюринга была построена специализированная электронная вычислительная машина **Colossus**. Она насчитывала 2000 радиоламп и обрабатывала 25000 СИМВ./с

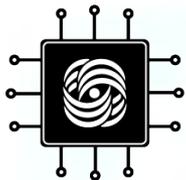




# Enigma

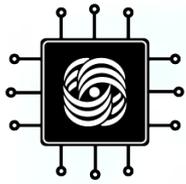
В местечке Блечли-Парк (Bletchley Park) под Лондоном была организована сверхсекретная криптоаналитическая лаборатория для расшифровки немецких военных шифров, используемых в шифровальной машине **Enigma**.





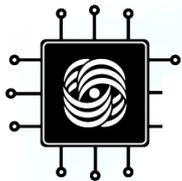
# Тест Тьюринга

- Опубликован в 1950 году
- Человек обменивается сообщениями на естественном языке с двумя собеседниками (человек и компьютер)
- Если человек не может определить кто есть кто, то считается что компьютер прошёл тест
- Переписка должна производиться через контролируемые промежутки времени
- Тьюринг оценил что программы в 2000 году пройдут тест
- Пока не подошли даже близко

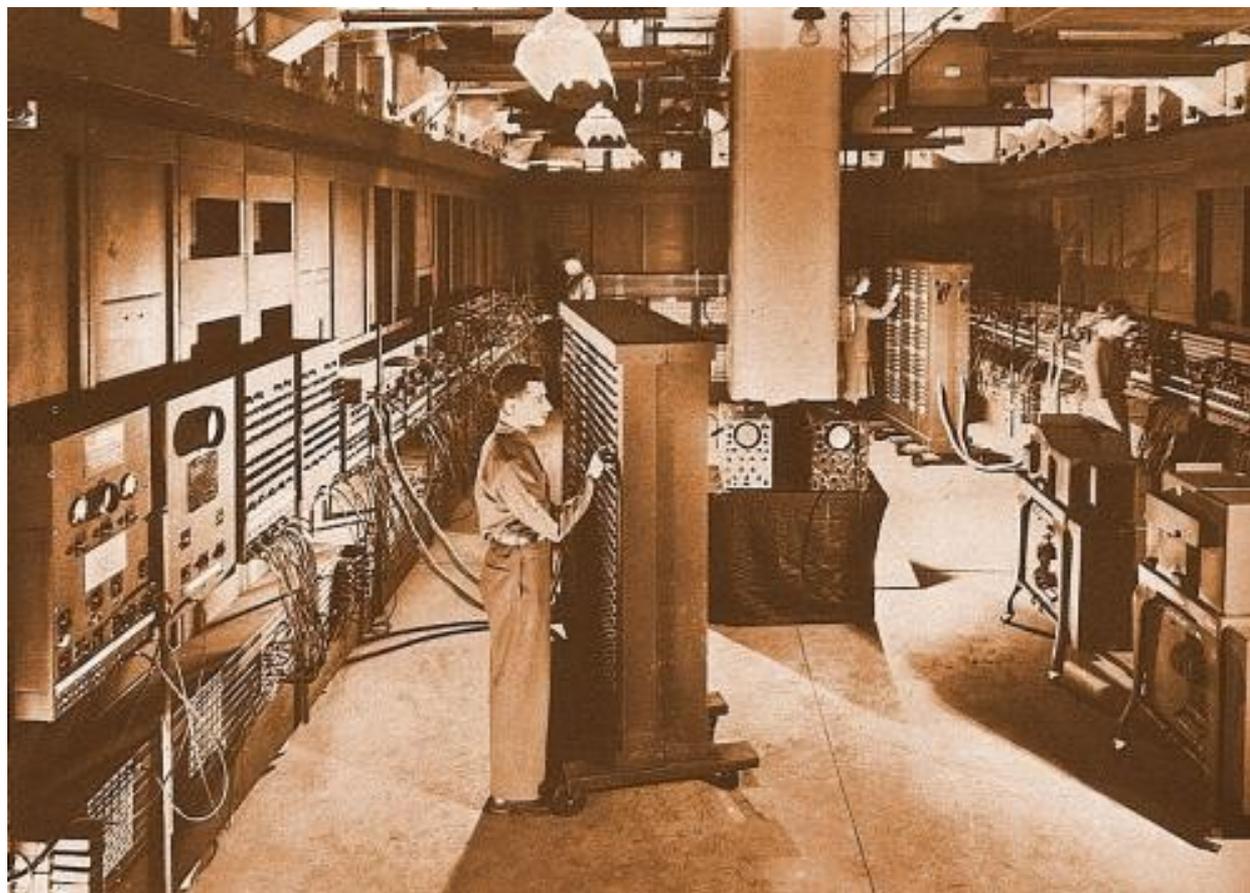


# Другие важные результаты

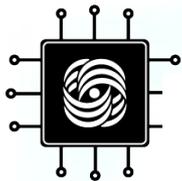
- Проблема самоприменимости МТ
- Универсальная МТ
- Основатель направления ИИ



## Первая ЭВМ ENIAC



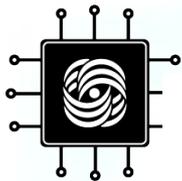
Первая работающая ЭВМ **ENIAC (Electronic Numerical Integrator And Calculator)** была создана в 1945 г. в Пенсильванском университете. Длина 26 м, высота 6 м, масса 30 т. 18 000 ламп, 1500 реле, потребляемая мощность 150 квт.



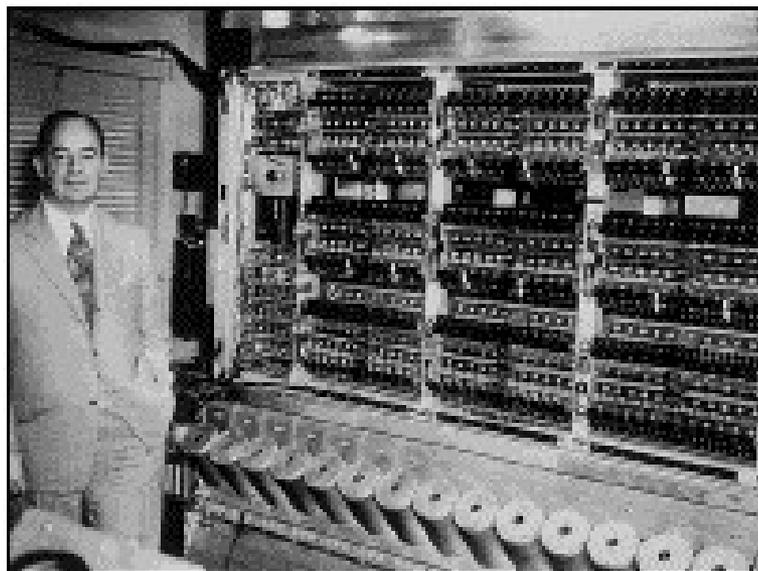
# Первая ЭВМ ENIAC



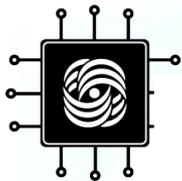
ЭВМ ENIAC. Вид сзади



## Проект фон Неймана и его вклад в архитектуру ЭВМ



Понятие «архитектура ЭВМ» связано с именем выдающегося математика XX столетия Джона фон Неймана (Neumann, John von; 1903-1957)



# Проект фон Неймана и его вклад в архитектуру ЭВМ

## ПРЕДВАРИТЕЛЬНОЕ РАССМОТРЕНИЕ ЛОГИЧЕСКОЙ КОНСТРУКЦИИ ЭЛЕКТРОННОГО ВЫЧИСЛИТЕЛЬНОГО УСТРОЙСТВА<sup>1)</sup>

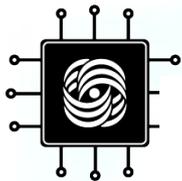
*А. Беркс, Г. Голдстейн, Дж. Нейман*

### 1. Основные компоненты машины

1.1. Так как законченное устройство будет универсальной вычислительной машиной, оно должно содержать несколько основных органов, таких, как органы арифметики, памяти, управления и связи с оператором. Мы хотим, чтобы машина была полностью автоматической, т. е. чтобы после начала вычислений работа машины не зависела от оператора. Более полно это за-

1.3. Выше мы в принципе указали на два различных вида памяти — память чисел и память приказов. Если, однако, приказы машине свести к числовому коду и если машина сможет некоторым образом отличать число от приказа, то орган памяти можно использовать для хранения как чисел, так и приказов. Кодирование приказов в числовой форме рассматривается в п. 6. 3.

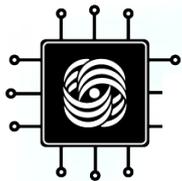
Фрагменты статьи фон Неймана с соавторами (русский перевод)



# Проект фон Неймана и его вклад в архитектуру ЭВМ

## Основные черты классической фон-неймановской архитектуры ЭВМ

2. Машина должна состоять из следующих основных блоков: арифметического устройства, оперативной памяти, устройства управления, устройства ввода, устройства вывода, устройства внешней памяти;
3. Команды программы должны храниться в оперативной памяти, откуда они последовательно выбираются и исполняются арифметическим устройством, система команд должна иметь операции условной и безусловной передачи управления. Команды должны рассматриваться как обычные данные, т.е. программа должна иметь возможность модифицировать себя в процессе вычислений;
4. Команды и данные должны храниться и обрабатываться в двоичной системе счисления.



## Проект фон Неймана и его вклад в архитектуру ЭВМ

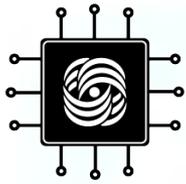


Морис Уилкс у машины  
EDSAC. 3000 ламп, ОЗУ  
512 слов

Из-за разногласий в команде разработчиков реализация проекта фон Неймана в США затянулась.

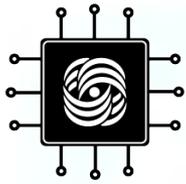
Первая ЭВМ с хранимой программой **EDSAC (Electronic Delay Storage Automatic Calculator)** была построена в Англии в 1949 г. под руководством Мориса Уилкса (Wilkes, Maurice; р. 1913).

Английские ученые опирались на собственный опыт разработки электронных вычислительных устройств во время Второй мировой войны



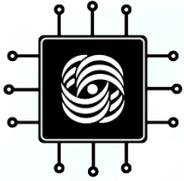
# Другие значимые достижения

- Квантовая физика
- Функциональный анализ
- Теория множеств
- Создатель теории игр
- Создатель теории клеточных автоматов



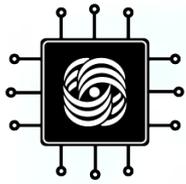
# Эдсгер Вайб Дейкстра (1930 – 2002)



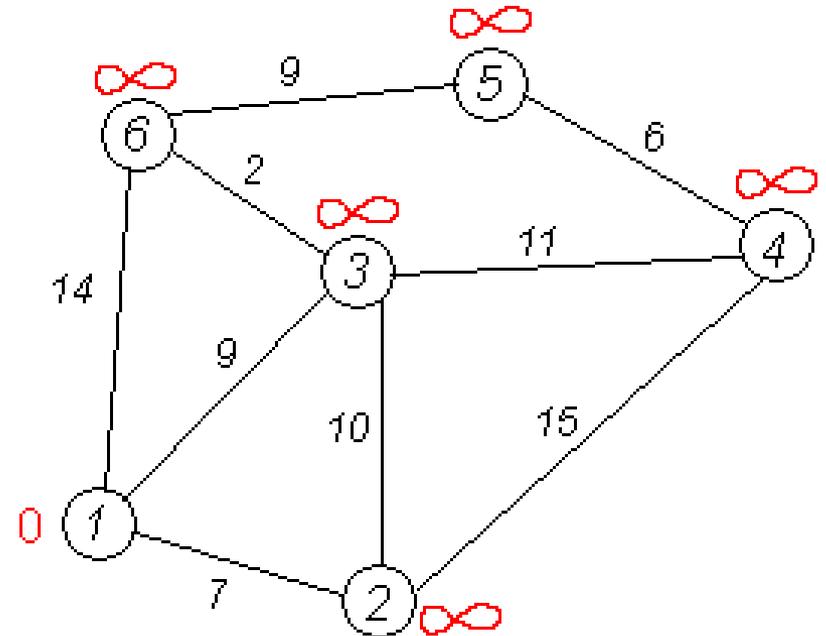
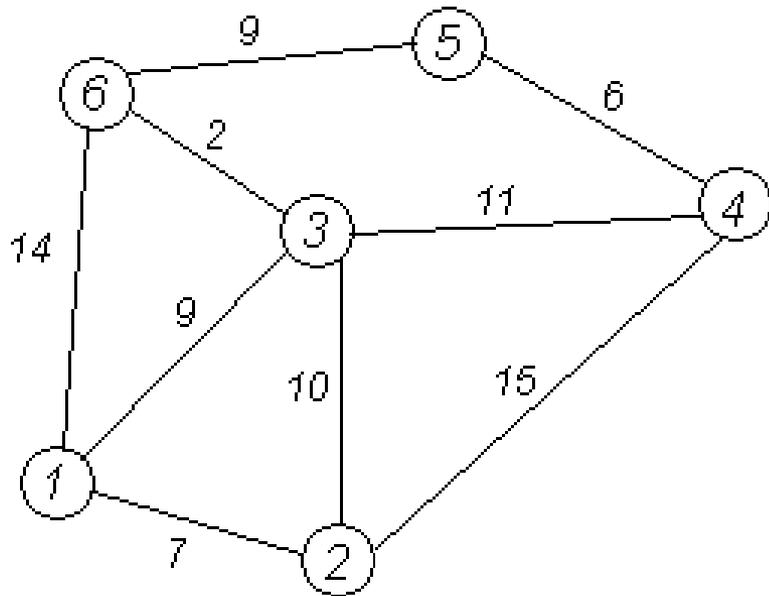


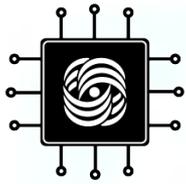
# Основные результаты

- Математическая логика
- Algol-60
- Концепция семафоров
- Алгоритм Дейкстра
- Борьба с оператором GOTO

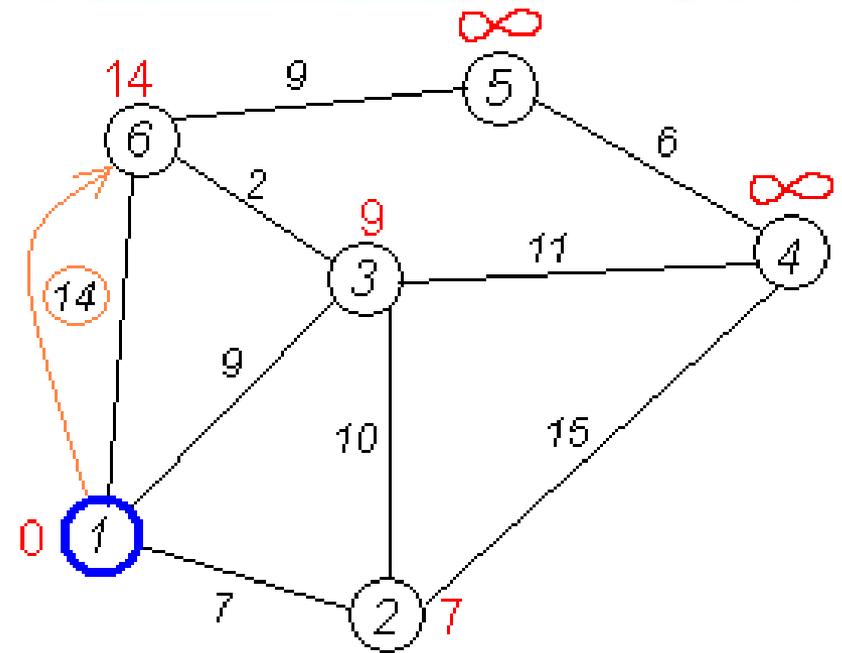
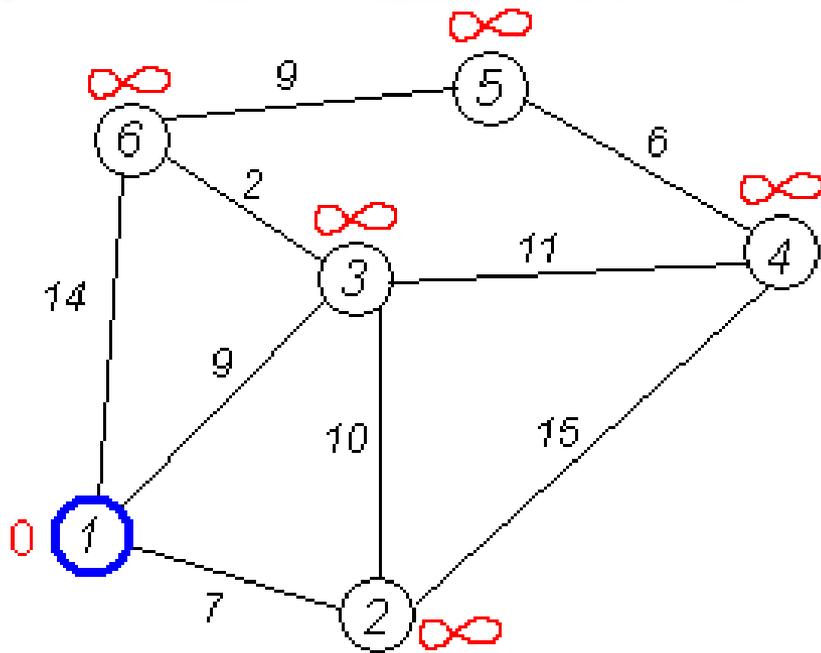


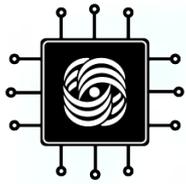
# Алгоритм Дейкстры (1)



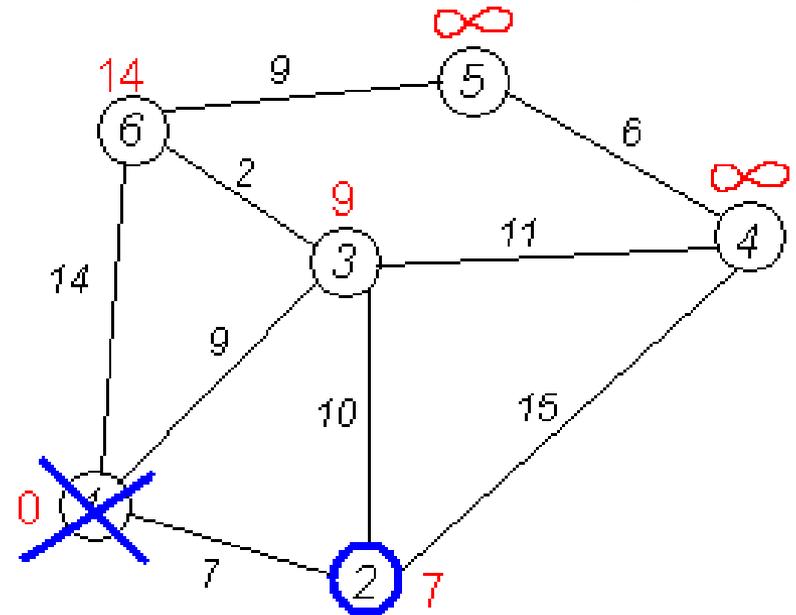
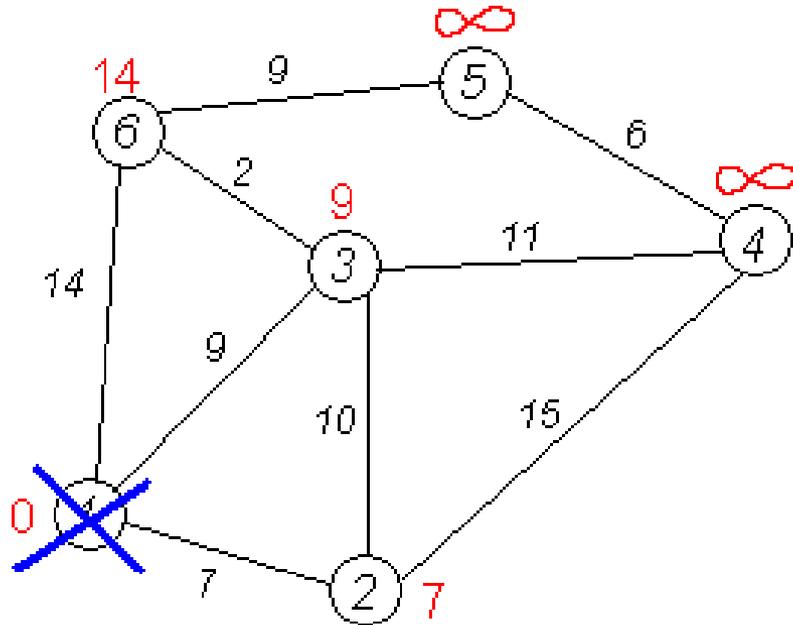


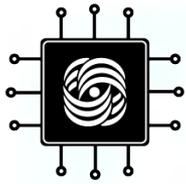
# Алгоритм Дейкстры (2)



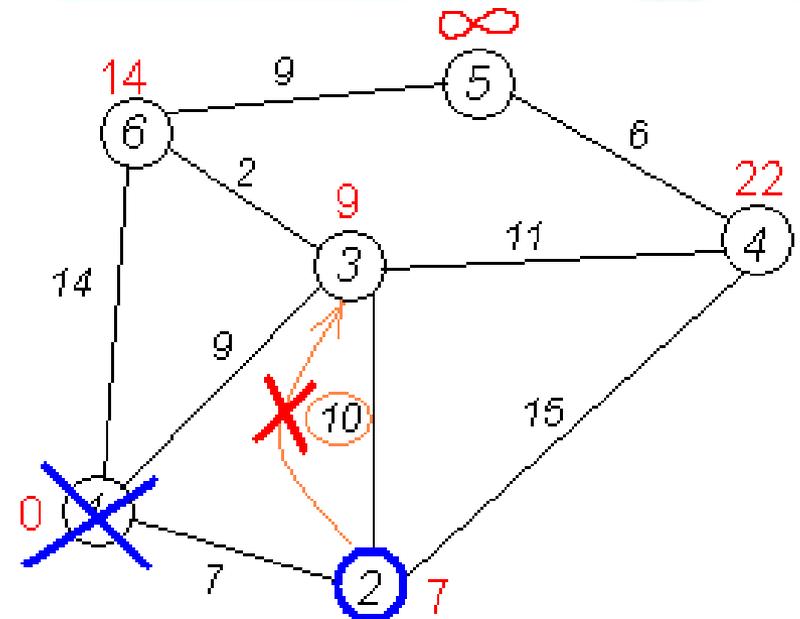
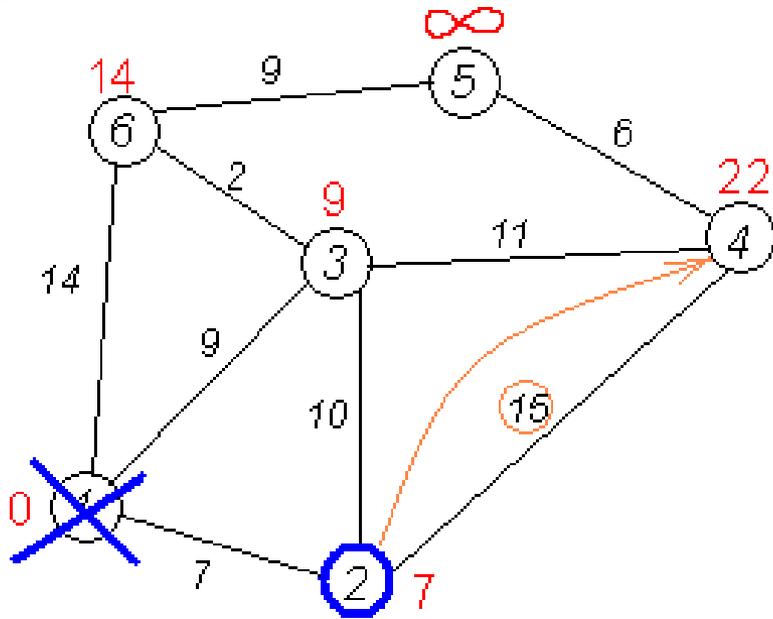


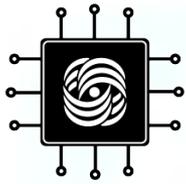
# Алгоритм Дейкстры (3)



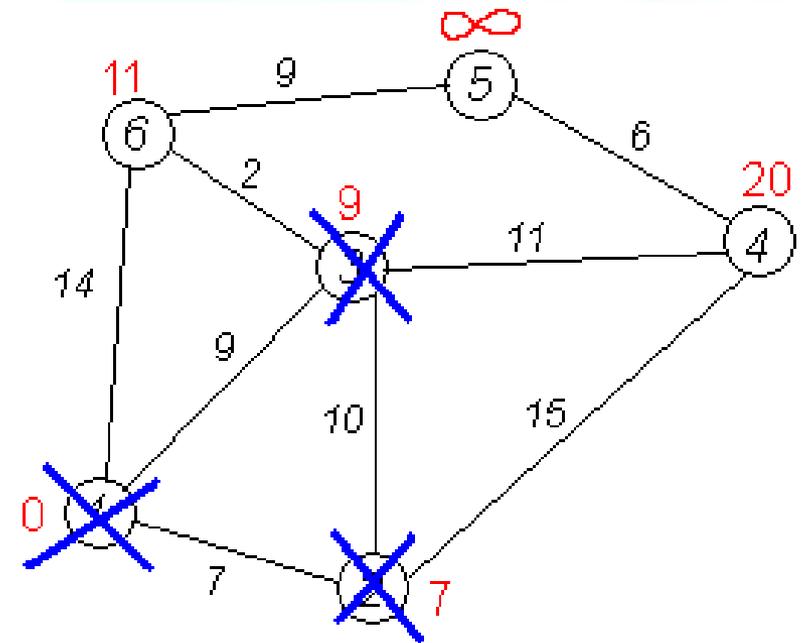
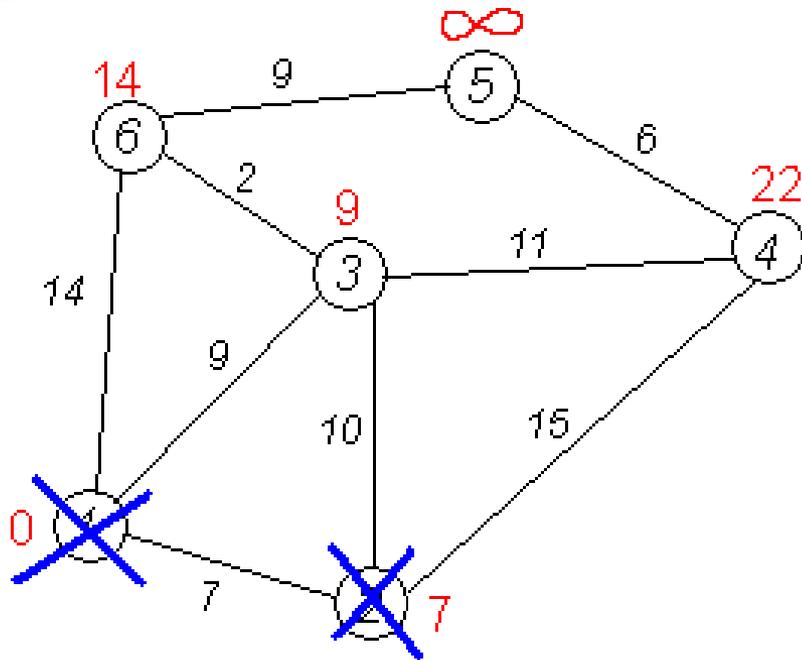


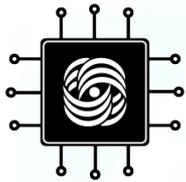
# Алгоритм Дейкстры (4)





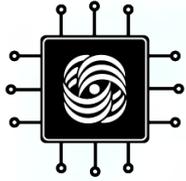
# Алгоритм Дейкстры (5)





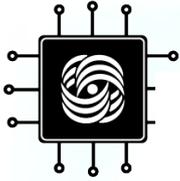
# Афоризмы (1)

- Студентов, ранее изучавших Бейсик, практически невозможно обучить хорошему программированию. Как потенциальные программисты они подверглись необратимой умственной деградации
- Вопрос «умеет ли компьютер думать» имеет не больше смысла, чем вопрос «умеет ли подводная лодка плавать»
- Проекты, предлагающие программирование на естественном языке, губельны по своей сути

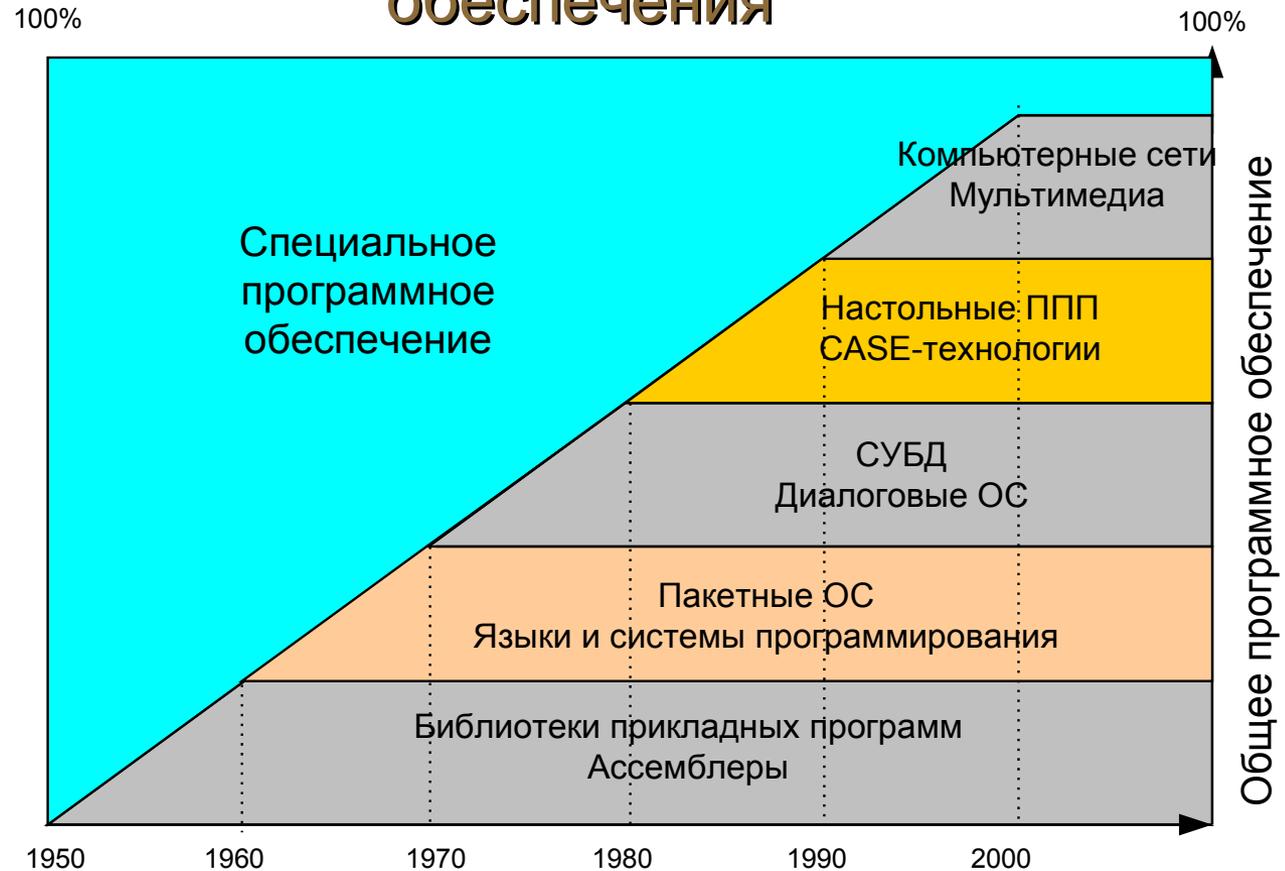


# Афоризмы (2)

- Дейкстра назвал модель IBM/360 (прообраз советской ЕС ЭВМ) — величайшей диверсией Запада против СССР
- На пустом диске можно искать вечно
- Если отладка — процесс удаления ошибок, то программирование должно быть процессом их внесения

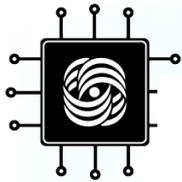


# Классификация и эволюция программного обеспечения



Эволюция программного обеспечения. Подобно тому, как в океане из плавающей мути откладываются геологические пласты, из специального программного обеспечения с течением времени образуются слои общего ПО



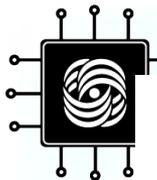


# Языки и системы программирования

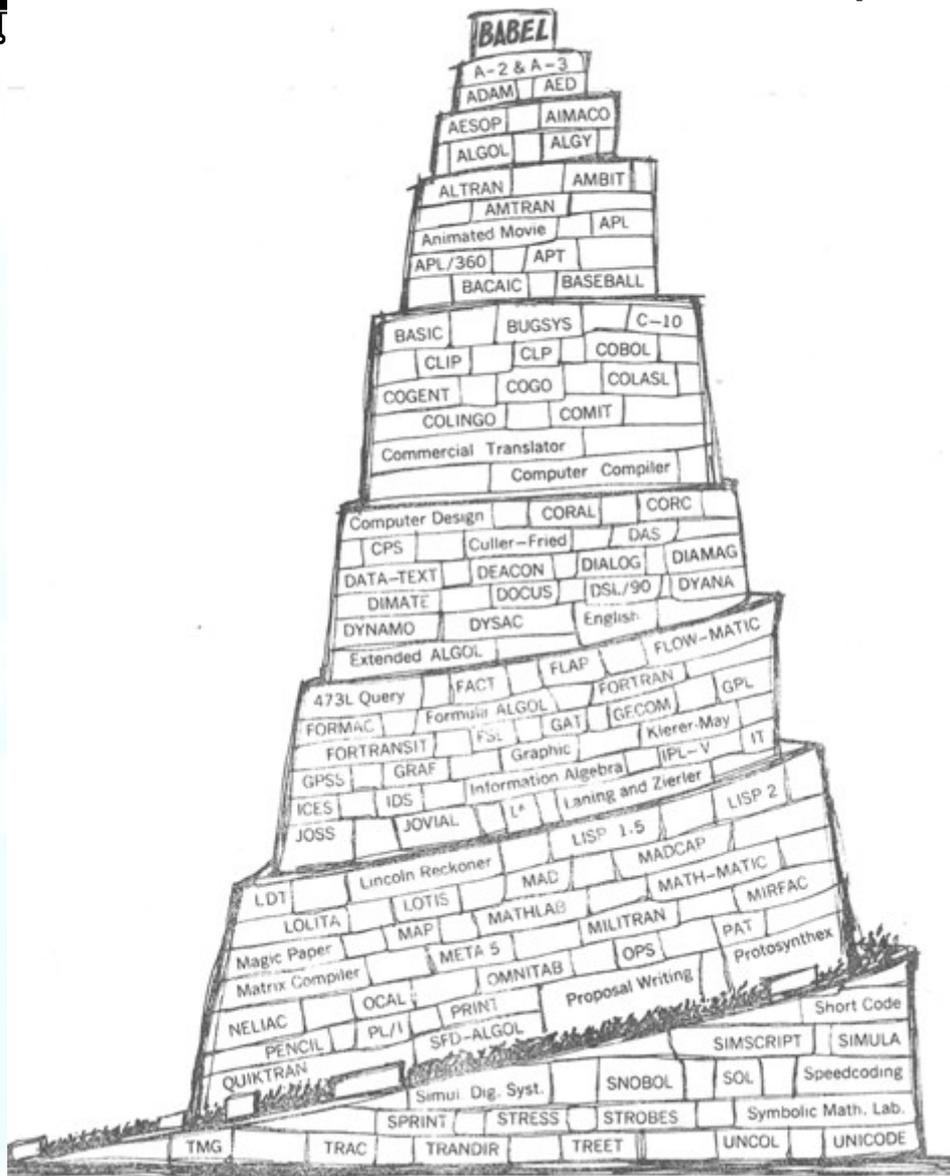
## Предыстория



Михаил Романович Шура-Бура и А.П. Ершов – создатели первых отечественных систем автоматизации программирования для ЭВМ «БЭСМ» и «Стрела» (1954-1956 годы)



# Языки и системы программирования

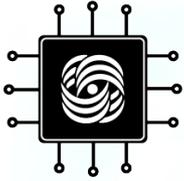


Наиболее активный период разработки языков и систем программирования приходится на 1960-е годы.

За это десятилетие в мире родилось более тысячи разнообразных языков, как универсальных, так и специализированных, но выжили и доросли до XXI века дожили немногие, в том числе бессмертные **Fotran**, **Basic**, **Algol**, **Cobol**, **Simula**, **Lisp** и их потомки.

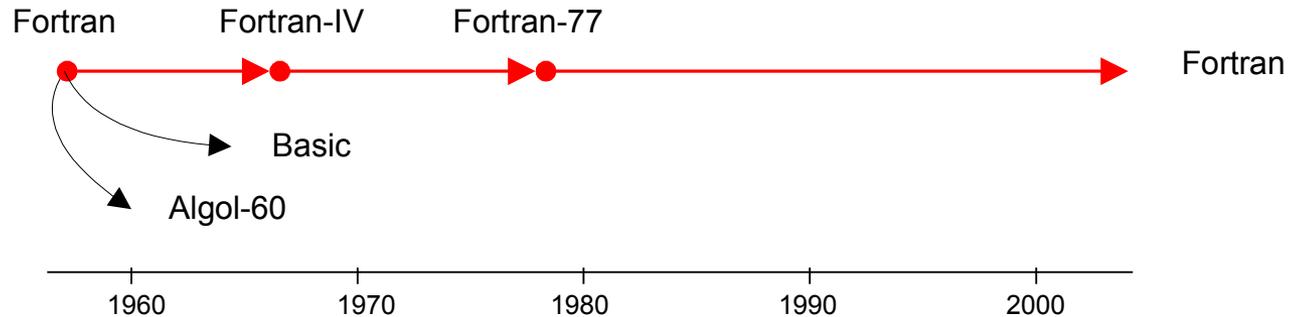
На рисунке: «вавилонская башня» языков программирования, созданных в 1960-е годы





# Языки и системы программирования

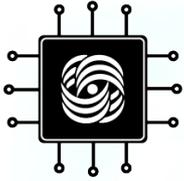
## Бессмертный Fortran



**Fortran = FORmula TRANslator**

Первый высокоуровневый язык программирования Fortran был разработан в фирме IBM под руководством Джона Бэкуса (Backus, John; р. 1924).

Работа над языком началась в 1954 г., первая реализация для IBM 704 в выполнена в 1957 г.



# Языки и системы программирования

## Бессмертный Fortran

```
C          MAIN PROGRAM

101      FORMAT(208)

102      FORMAT(//'N=',15, 5X, 'R=', 15
              1//6X, 'M', 5X, PROB)

103      FORMAT(18, F14.10)

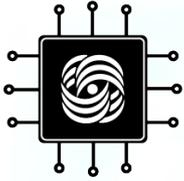
201      READ(1,101) N, IR
          WRITE(3,102) N, IR
          IF(N) 202, 202, 203

202      STOP

203      IF(IR) 202, 202, 204

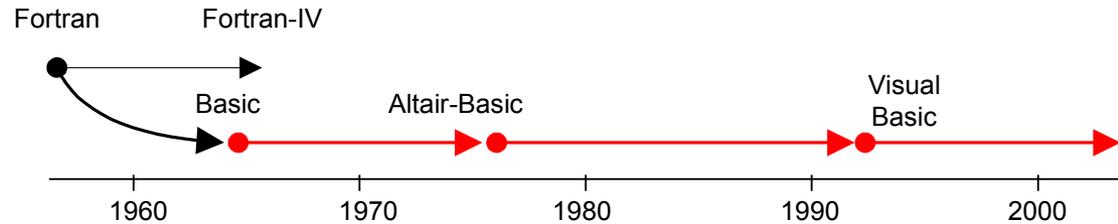
204      M=0
          P=COMBF(N,M)*COMBF(IR-1,N-M-1)
          1/COMBF(N+IR-1,IR)      ...
```

Фрагмент программы на языке Fortran



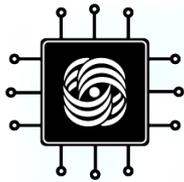
# Языки и системы программирования

## Basic – язык для начинающих



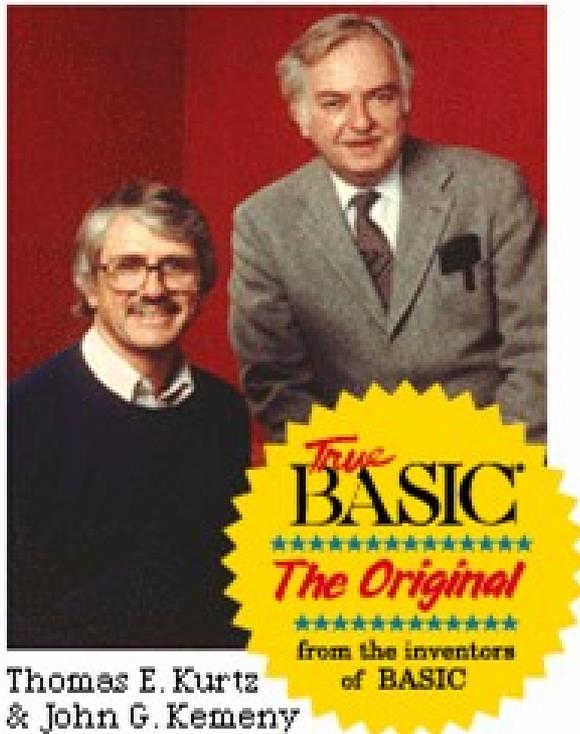
**BASIC = Beginners All-purpose Symbolic Instruction Code**

Язык Basic был разработан в 1964 г. в Дармутском колледже в г. Хановере (Dartmouth College, Hanover), штат Нью-Хемпшир



# Языки и системы программирования

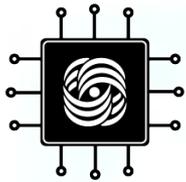
## Basic – язык для начинающих



Авторы языка Basic.  
Стоит Джон Кемени  
(Kemeny, John G.; 1926-1993),  
сидит Томас Курц  
(Kurtz, Thomas E.; р. 1928)

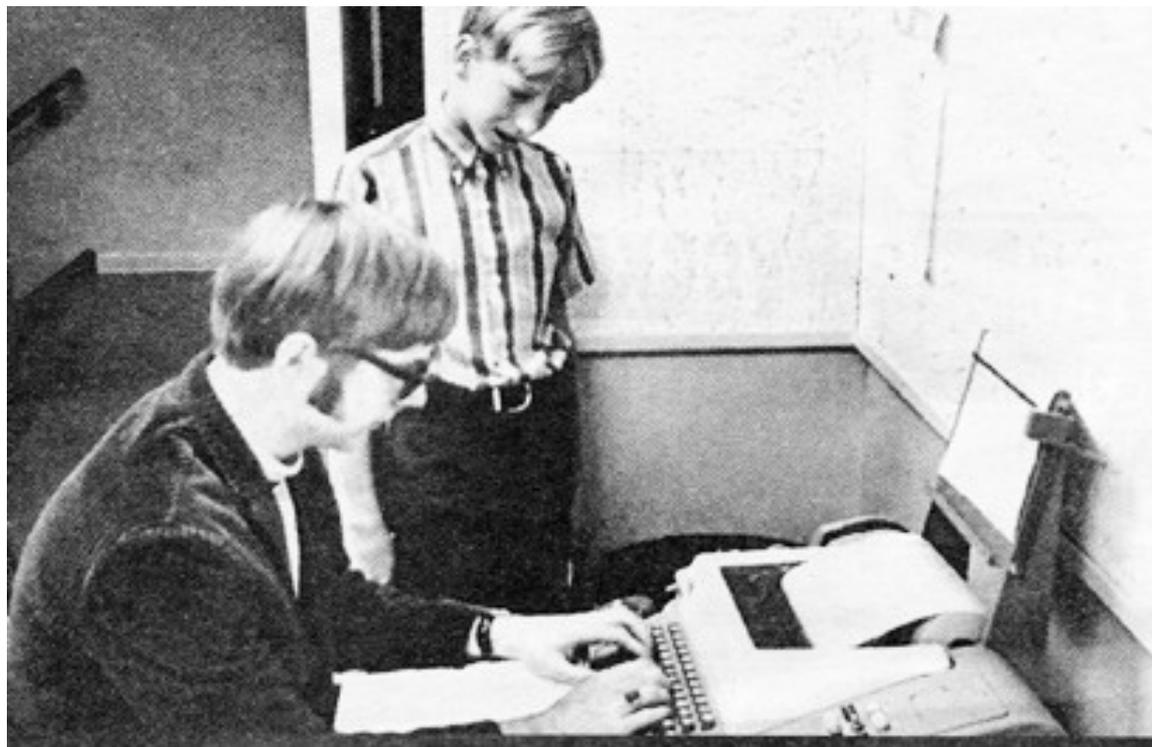
```
10 dim A(5)
20 for i=1 to 5
30 input A(i)
40 next i
50 if i=5 then goto 140
60 if A(i)<=A(i+1) then goto 90
70 i=i+1
80 goto 130
90 z=A(i)
100 A(i)=A(i+1)
110 A(i+1)=z
120 i=1
130 goto 50
140 for i=1 to 5
150 print A(i)
160 next i
```

Простейшая  
программа на  
языке Basic

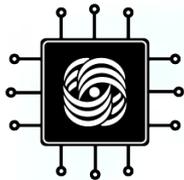


# Языки и системы программирования

## Basic – язык для начинающих



Будущие создатели Microsoft Пол Аллен (Allen, Paul; р. 1954) и Билл Гейтс (Gates, William; р. 1955) познакомились с Бэйсиком, работая в компьютерном классе школы в Сиэтле (снимок 1968 г.)



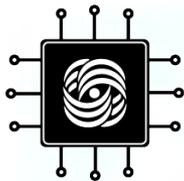
# Языки и системы программирования

## Basic – язык для начинающих

Начав с Бэйсика, компания Microsoft превратилась в крупнейшую софтверную империю, а Билл Гейтс – стал самым богатым человеком на планете

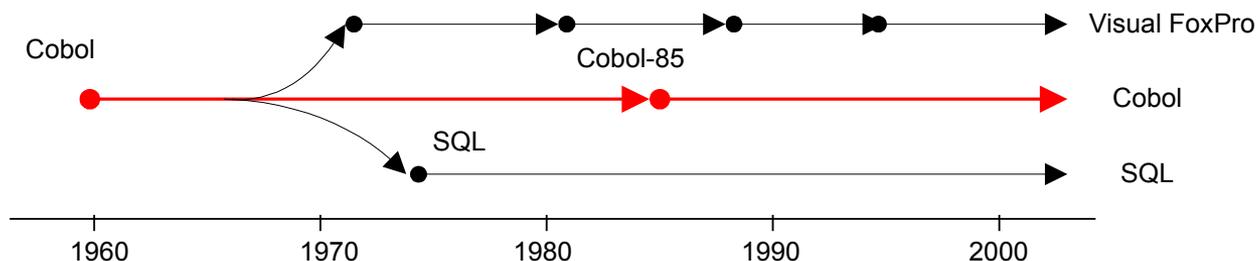


Штаб - квартира  
корпорации Microsoft в  
Редмонде (пригород  
Сиэтла)



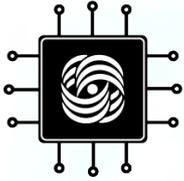
# Языки и системы программирования

## Cobol – язык для бухгалтеров



**COBOL = COmmon  
Business-Oriented  
Language**

На фото: разработчики языка Cobol у шуточного обелиска, присланного в их адрес в качестве намека на безнадежно медленную работу, способную похоронить саму идею. Справа внизу – Грейс Хоппер

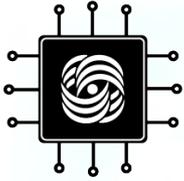


# Языки и системы программирования

## Cobol – язык для бухгалтеров

Основные свойства языка Cobol:

- независимость программ от оборудования;
- независимость программ от данных;
- сложные структуры данных;
- синтаксис, приближенный к естественному английскому языку.



## 3.2. Языки и системы программирования Cobol – язык для бухгалтеров

### 1010 IDENTIFICATION DIVISION.

1020 PROGRAM-ID “EXAMPLE”.

### 1030 ENVIROMENT DIVISION.

1040 INPUT-OUTPUT SECTION.

1050 FILE-CONTROL.

1060 SELECT CD ASSIGN TO “SYS010” UNIT-RECORD 2540R.

1070 SELECT TT ASSIGN TO “SYS009” UTILITY 2400.

### 1080 DATA DIVISION.

1090 FILE SECTION.

1100 FD CD DATA RECORD IS C

1110 LABEL RECORDS ARE OMITTED.

1120 01 C.

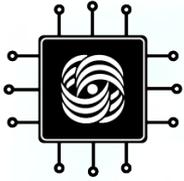
1130 02 C1 PICTURE 9(4).

1140 02 C2 PICTURE 9.

1150 02 C3 PICTURE X(70).

...

Программа на Коболе  
(начало)



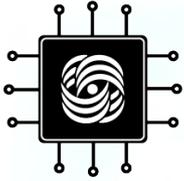
## 3.2. Языки и системы программирования

### Cobol – язык для бухгалтеров

#### 1290 PROCEDURE DIVISION.

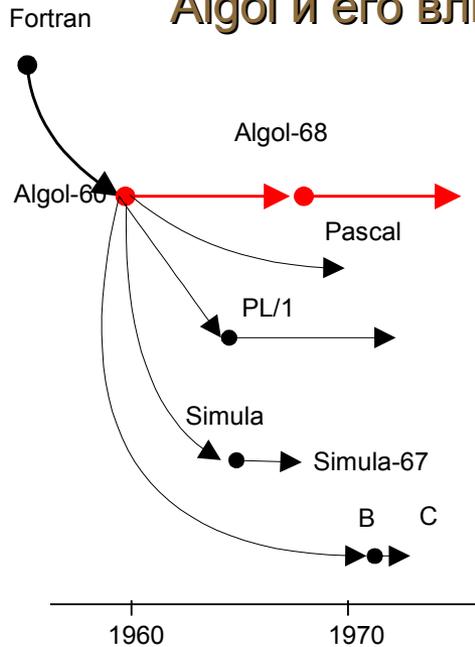
```
1300 P1.  OPEN INPUT CD, OUTPUT TT.  
1310 P2.  READ CD, AT END GO TO P3.  
1320          MOVE C1 TO D1.  
1330          MONE C2 TO D2.  
1340          MOVE C3 TO D3.  
1350          ADD C1, C2, GIVING D4.  
1360          WRITE T FROM D.  
1370          GO TO P2.  
1380 P3.  CLOSE SD, TT.  
1390          STOP RUN.
```

Программа на Коболе (окончание)



# Языки и системы программирования

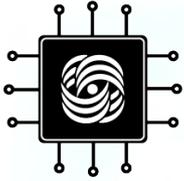
## Algol и его влияние на языки программирования



### ALGOL = ALGORitmic Language

В 1958 году в Цюрихе (Швейцария) состоялась международная конференция, предложившая проект нового универсального международного языка программирования Algol-58. В 1960 году на парижской конференции была принята окончательная версия под названием Algol-60.

На снимке: участники парижской конференции голосуют за Алгол-60.



# Языки и системы программирования

## Algol и его влияние на языки программирования

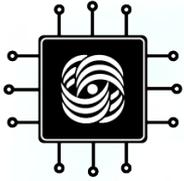
### Основные свойства языка Algol-60:

- машинная независимость;
- формальный синтаксис;
- описание переменных и блочная структура;
- рекурсия

### Нормальная форма Бэкуса-Наура (БНФ)

$\langle \text{цифра} \rangle ::= 1|2|3|4|5|6|7|8|9|0$

$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \langle \text{целое без знака} \rangle$



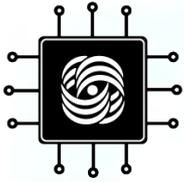
# Языки и системы программирования

## Algol и его влияние на языки программирования

```
begin  
  
    integer i, n;  
  
    real s;  
  
    real array x[1:n];  
    s:=0;  
  
    for i:=1 step 1 to n do  
        s:=s+x[i];  
    s:=s/n  
  
end
```

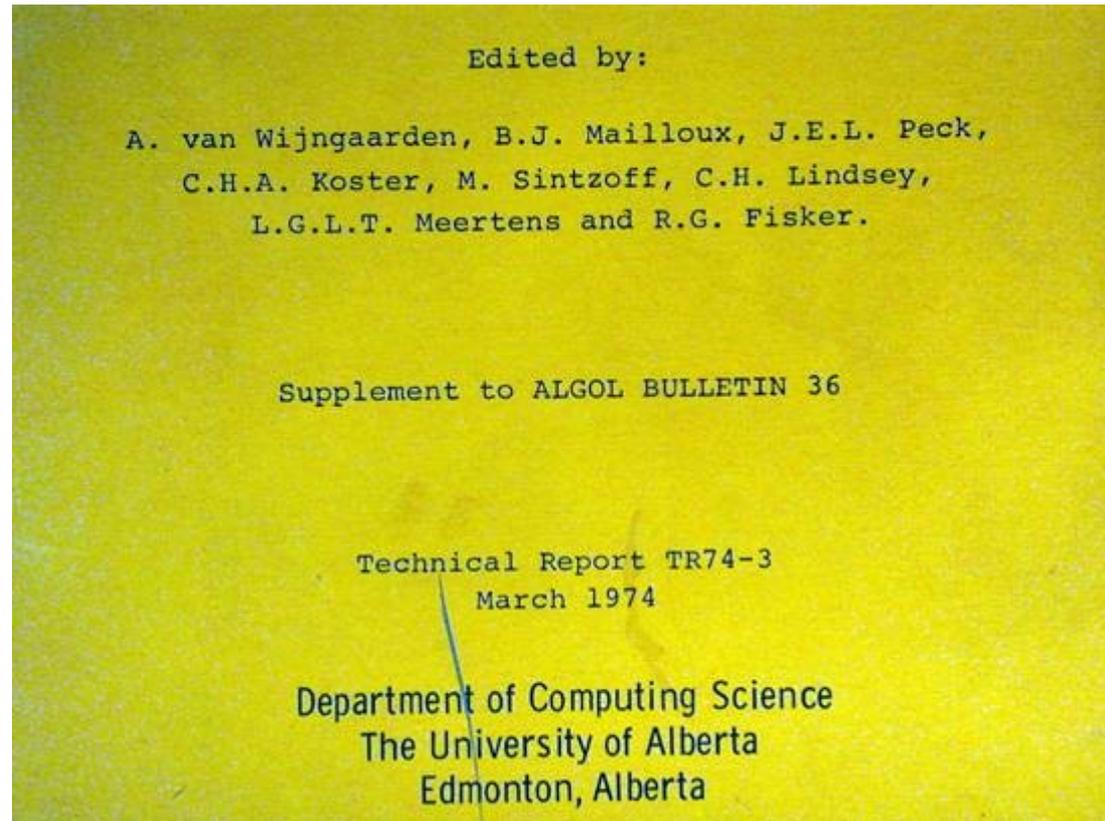
Простейшая программа на Алголе-60, вычисляющая среднее арифметическое n чисел.

Синтаксис Алгола-60 сформировал стандарт для всех последующих языков программирования

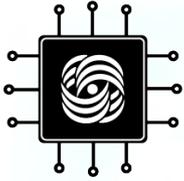


# Языки и системы программирования

## Algol и его влияние на языки программирования

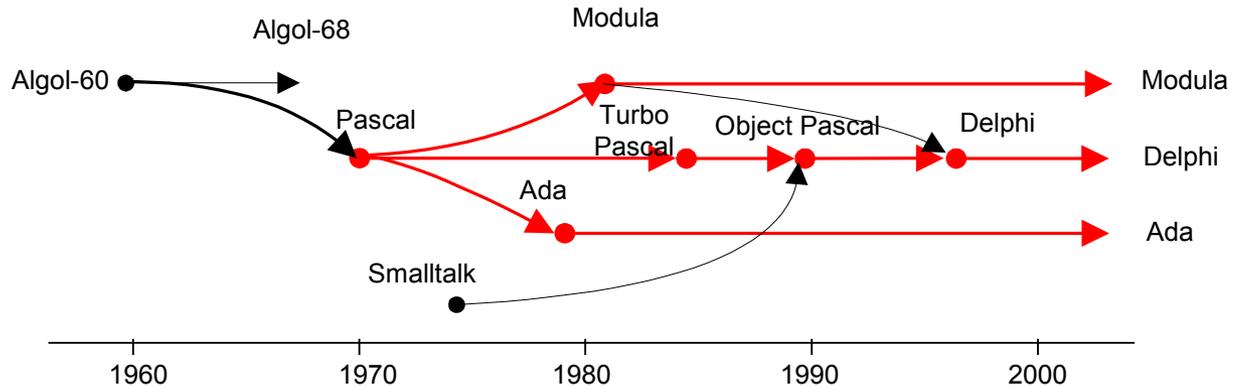


В результате многолетней переработки Алгола-60 комитетом **IFIP** появился язык **Алгол-68** (пересмотренное сообщение под ред. А. ван Вейнгаардена (A. van Wijngaarden) и др. опубликовано в 1975 г.)



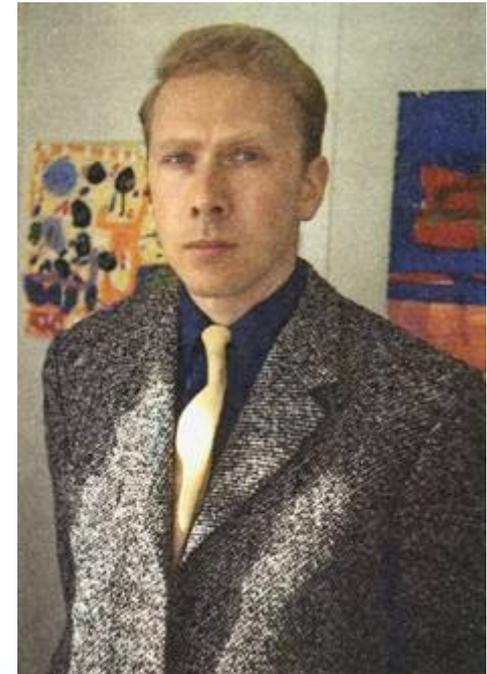
# Языки и системы программирования

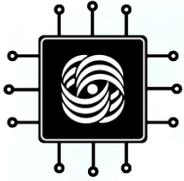
## Pascal и его потомки



Член комитета по Алголу-68 Никлаус Вирт (Wirth, Niklaus; р. 1934) был против принятия переусложненного стандарта.

В знак доказательства своей правоты он разработал в 1971 г. простой и ясный алголоподобный язык, предназначенный прежде всего для обучения студентов в Федеральном техническом университете в Швейцарии. В честь изобретателя первой вычислительной машины Вирт назвал язык **Паскалем**.



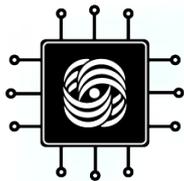


# Языки и системы программирования

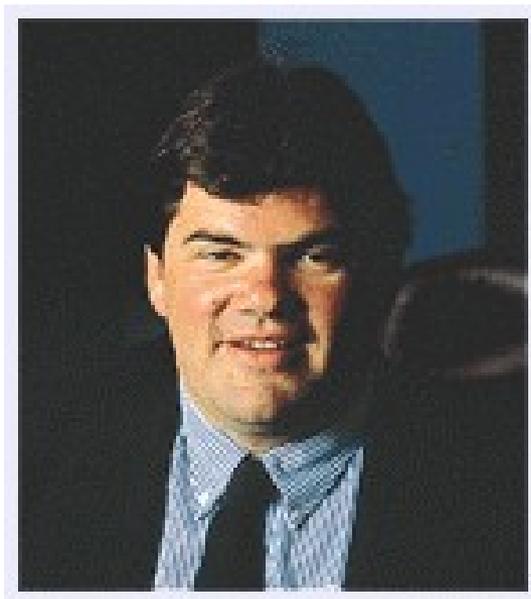
## Pascal и его потомки

```
var
    i, n: integer;
    s: float;
    x: array[1..n] of real;
begin
    s:=0;
    for i:=1 to n do
        s:=s+x[i];
    s:=s/n
end.
```

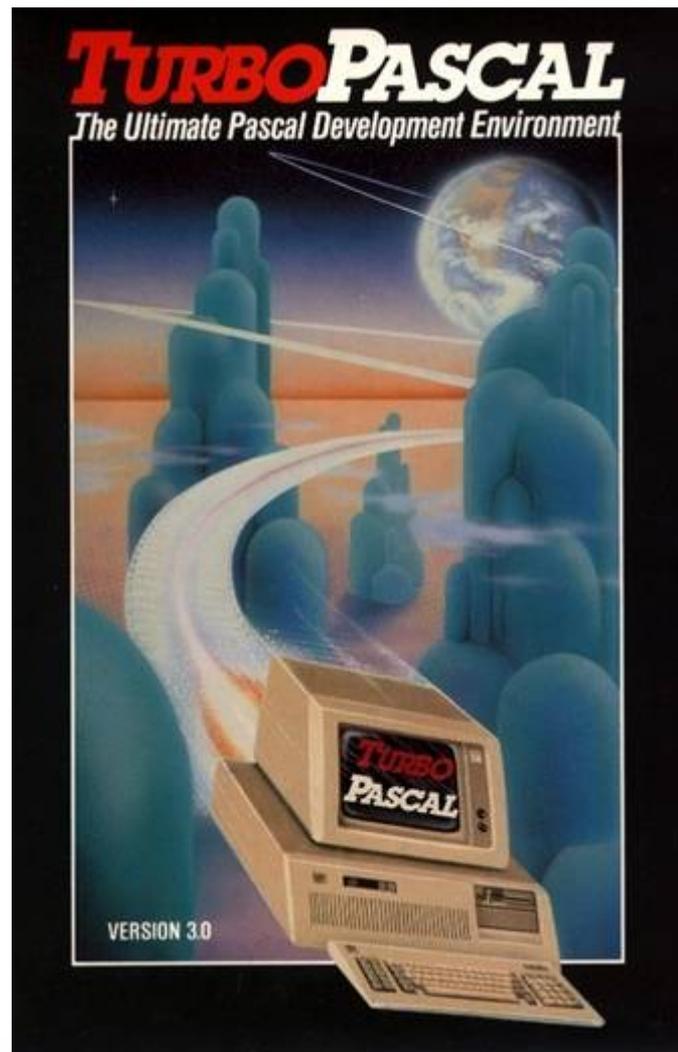
Программа на Паскале, вычисляющая среднее арифметическое  $n$  чисел

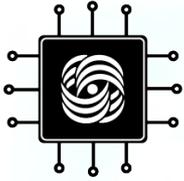


## Языки и системы программирования Pascal и его потомки



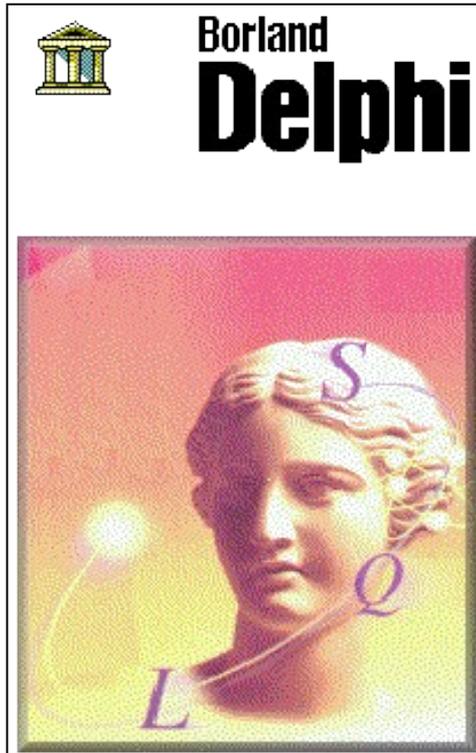
Новую жизнь языку Pascal дал Филипп Кан (Kahn, Philippe; р. 1938) – создатель компилятора Turbo Pascal для IBM PC и основатель компании Borland (1984 г.)





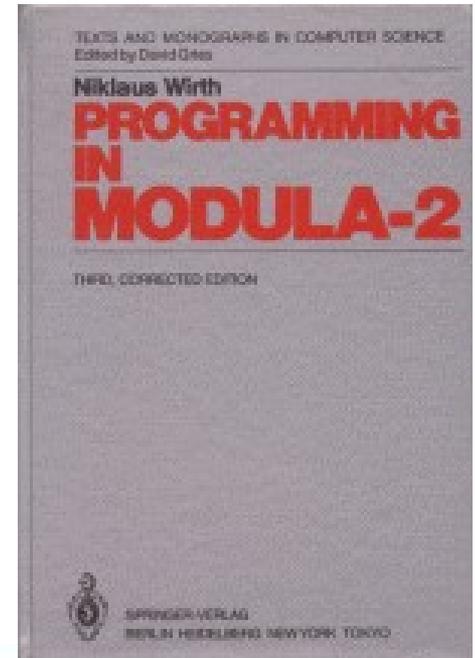
# Языки и системы программирования

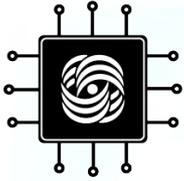
## Pascal и его потомки



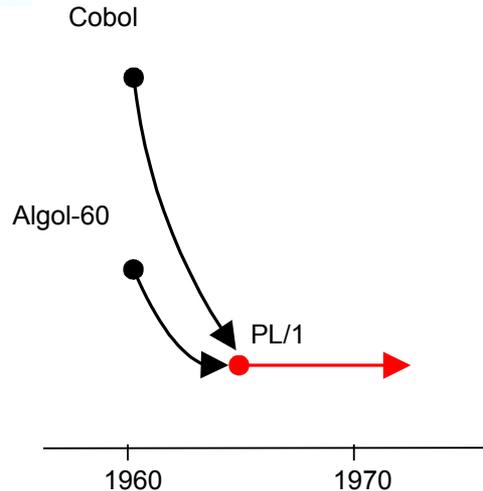
Среда разработки Delphi фирмы Borland объединила передовые достижения технологии программирования: объектное расширение языка Pascal, визуально-событийное проектирование, модульное структурирование и отдельная компиляция.

В отличие от учебного Паскаля, язык программирования Modula-2, предложенные Никлаусом Виртом, изначально предназначался для профессионального применения





# Языки и системы программирования

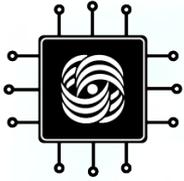


## PL/1 = Programming Language One

Язык PL/1 был частью амбициозного проекта IBM S/360, он создавался в спешке и представлял собой механическую смесь идей из многих языков. Критики сравнивали его с елкой со множеством украшений.

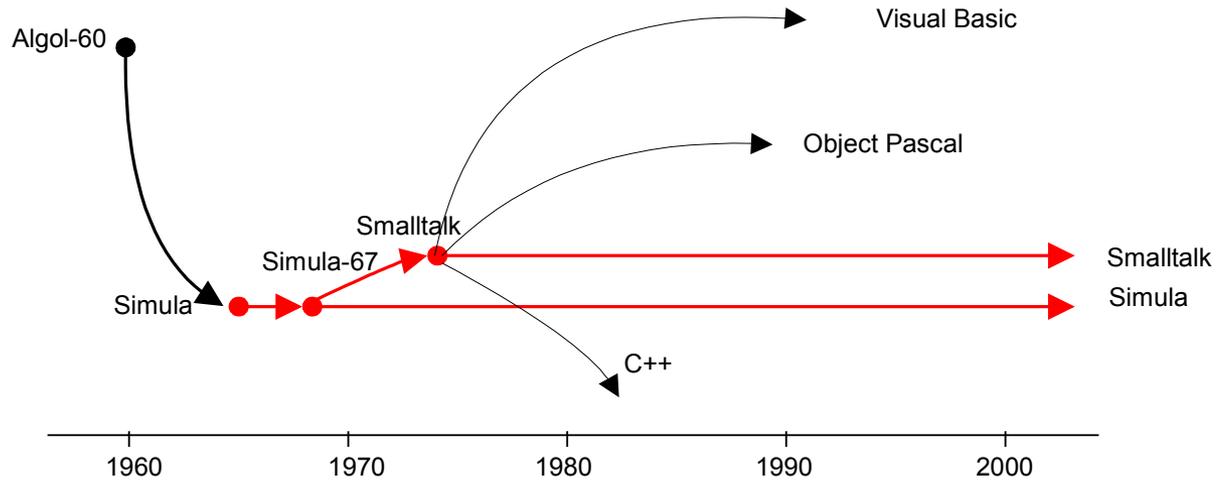
```
EXAMPLE: PROCEDURE OPTIONS (MAIN);
ON ENDFILE (SYSIN) GO TO ENDING;
P1:      GET LIST (A, B, C);
         D = B*B — 4*A*C;
         E = —B/(A+A);
         IF D<0 THEN DO;
           X1, X2 = E;
           Y1 = SQRT(—D)/(A+A);
         END;
         ELSE DO;
           R = SQRT(D)/(A+A);
         ...
         Y1 = 0;
         END;
         Y2 = —Y1;
         PUT LIST (X1, Y1, X2, Y2);
         GO TO P1;

ENDING;;
END EXAMPLE;
```



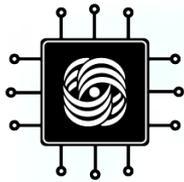
# Языки и системы программирования

## Simula и Smalltalk – революция в программировании



### Simula = SIMULAtion

За разработку языка Simula Кристен Нигорд (Nygaard, Kristen; 1926-2002), на снимке слева, и Оле-Йохан Дал (Dahl, Ole-Johan; 1931-2002) были удостоены высшей награды компьютерного сообщества – медали Тьюринга



## Языки и системы программирования Simula и Smalltalk – революция в программировании

```
|a|
```

```
a := Array new: 5.
```

```
1 to: 5 do: [:i | a at: i put:
```

```
(Prompter prompt: 'Введите элемент массива')  
asNumber].
```

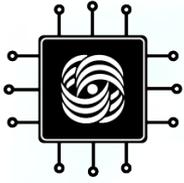
```
a := a asSortedCollection.
```

```
a do: [:i | Transcript putAll: i printString].
```

Простейшая программа  
на Smalltalk,  
вычисляющая среднее  
арифметическое пяти  
чисел

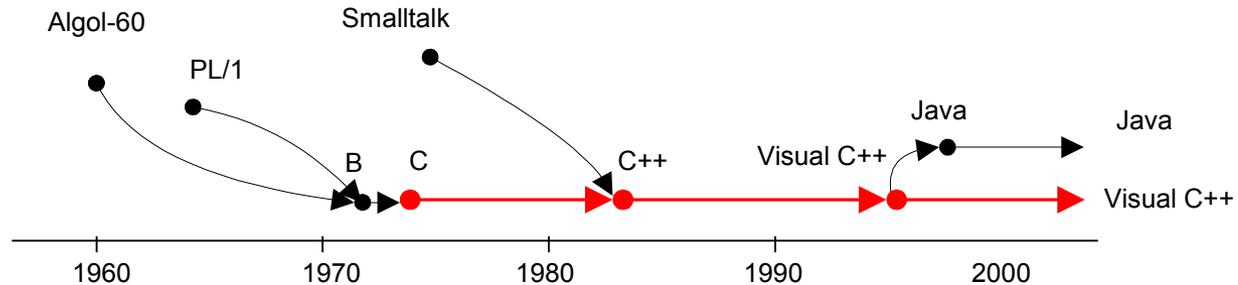
**Алан Кей**



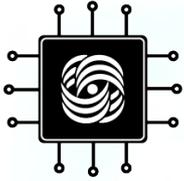


# Языки и системы программирования

## C – язык для профессионалов



Язык **Си (C)** был создан Деннисом Ричи (Ritchie, Dennis M.; р. 1941) в 1973 году в Bell Labs в ходе разработки операционной системы UNIX. Он развивал язык **Би (B)**, который основывался на созданном в Кембриджском университете языке **BCPL** (от **B**asic **C**ombined **P**rogramming **L**anguage), который в свою очередь был потомком Алгола-60

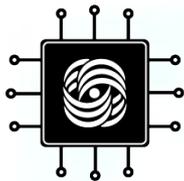


# Языки и системы программирования

## C – язык для профессионалов

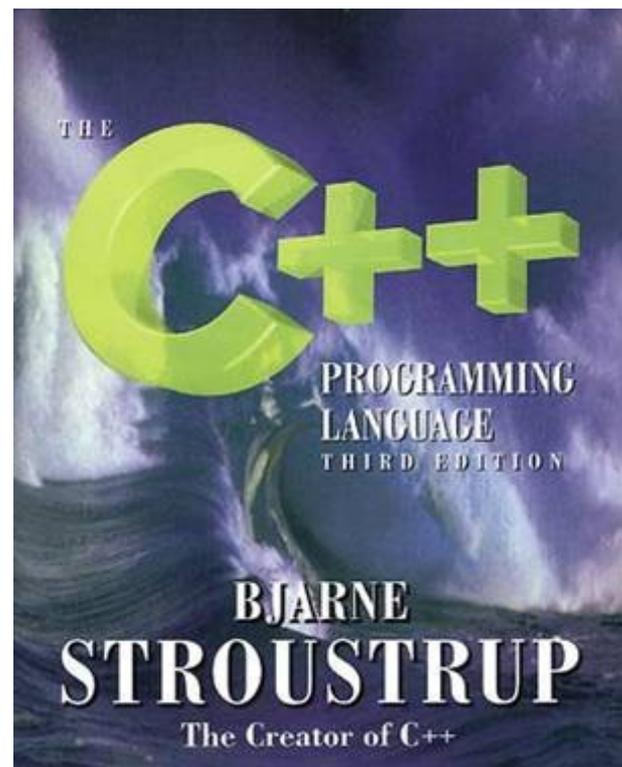
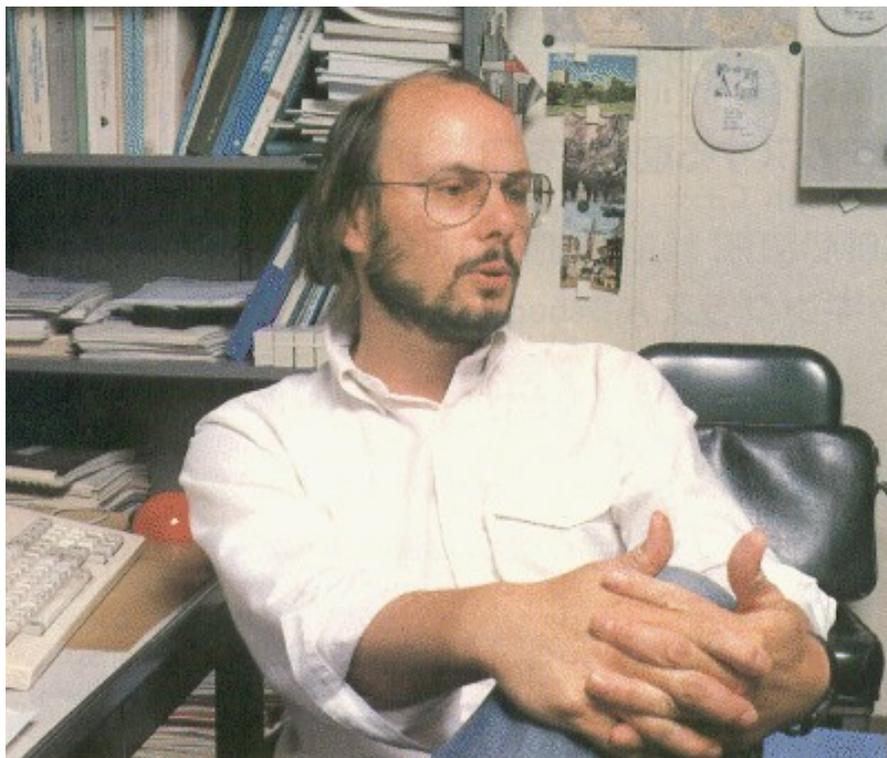
```
float A[5];
for(int i=0;i<5;i++)scanf("%f",&A[i]);
i=0;
while(i<4){
    if(A[i]<=A[i+1])i++;
    else{
        z=A[i];
        A[i]=A[i+1];
        A[i+1]=z;
        i=0;
    }
};
for(i=0;i<5;i++)printf("%f\n",A[i]);
```

Текст на языке C отличается лаконичностью

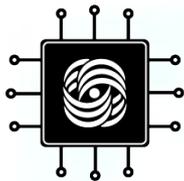


# Языки и системы программирования

## C – язык для профессионалов

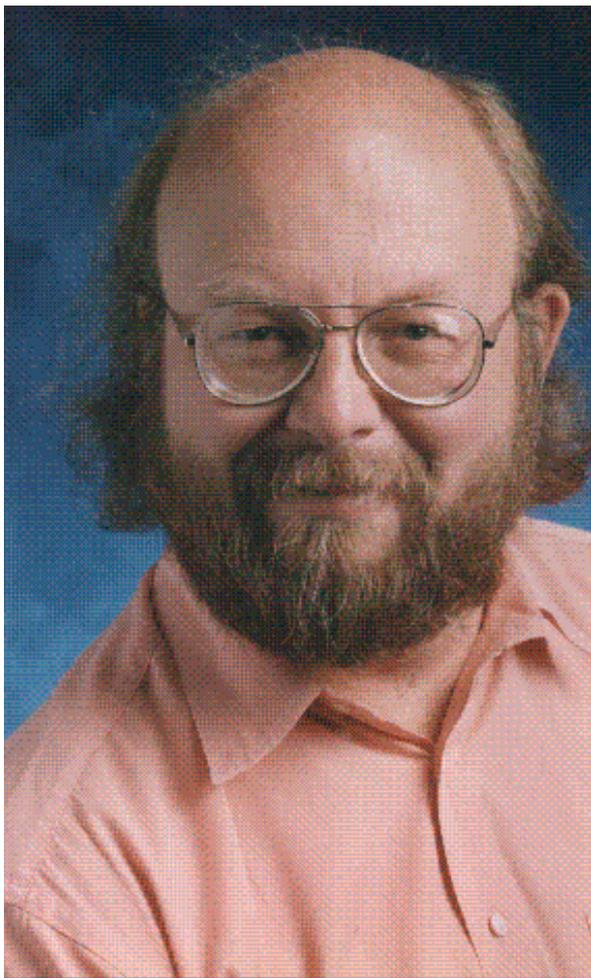


Бьярн Страуструп (Stroustrup, Bjarne; р. 1950) ввел в язык C объекты и превратил его в C++



# Языки и системы программирования

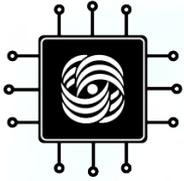
## Java – дитя интернета



В 1995 г. фирма Sun Microsystems представила язык **Java** для программирования в интернете.

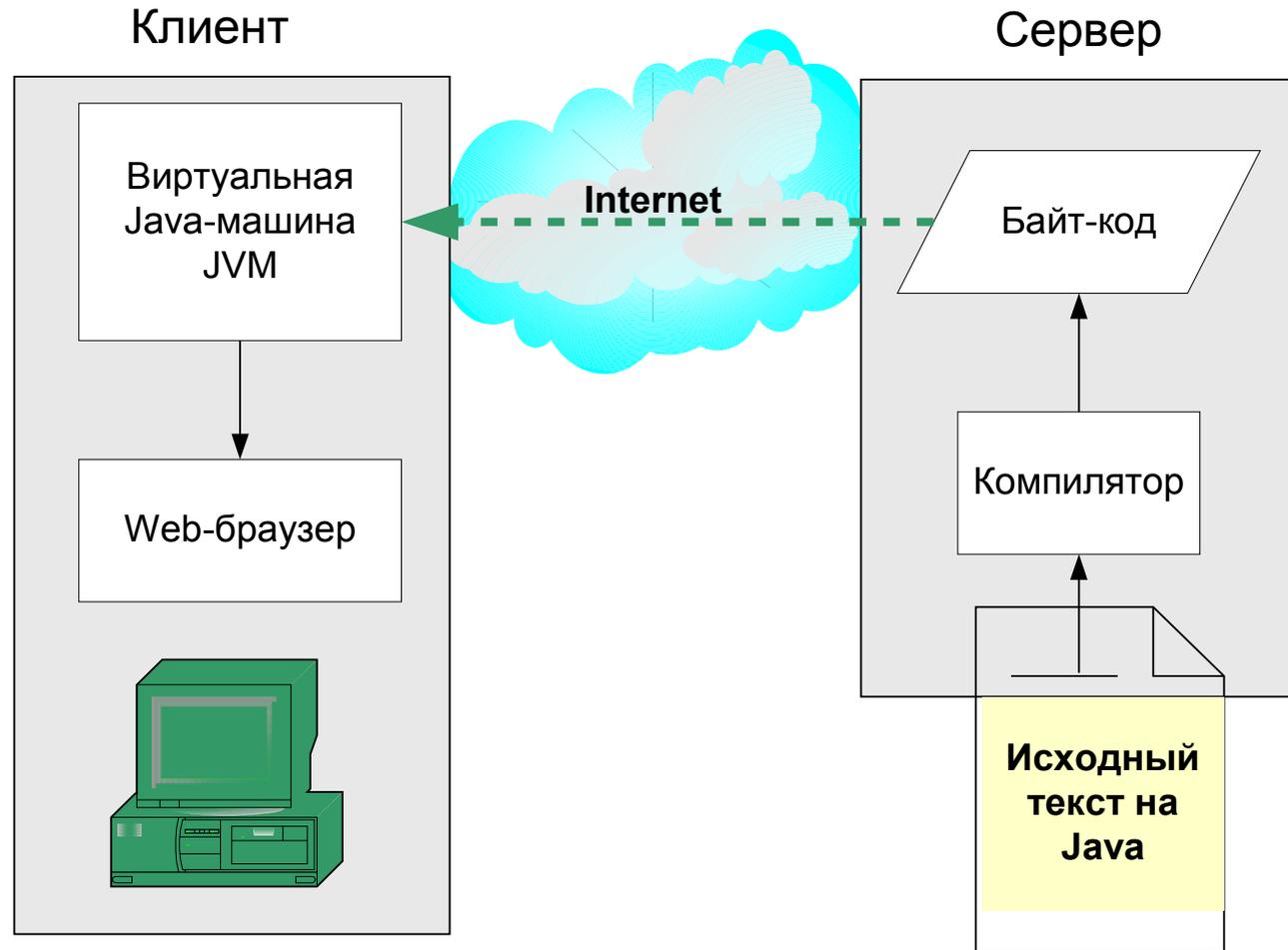
Он возник в ходе реализации проекта **Oak** («Дуб»), целью которого было создание системы программирования бытовых микропроцессорных устройств.

Джеймс Гослинг (Gosling, James) – автор Java.

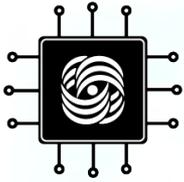


# Языки и системы программирования

## Java – дитя интернета



Java - технология



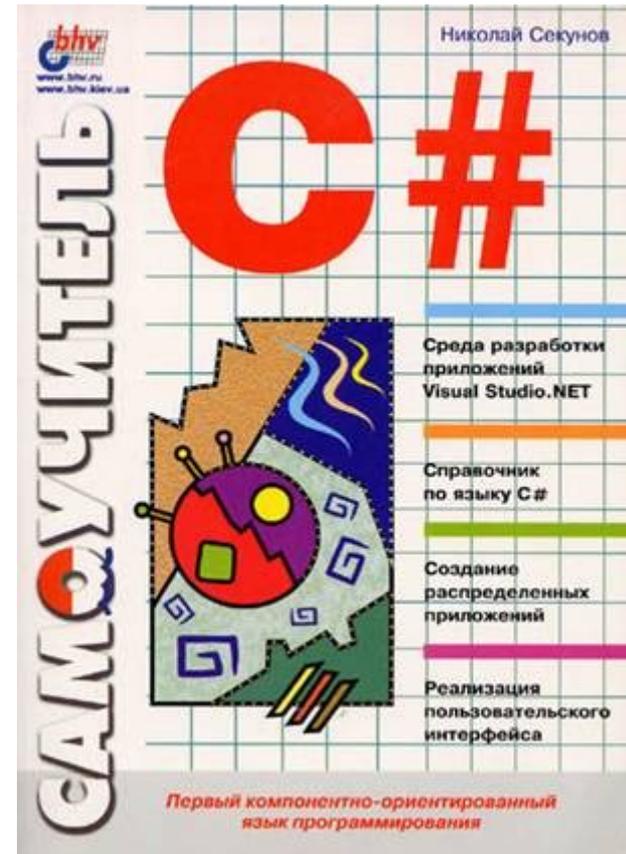
# Языки и системы программирования

## Java – дитя интернета

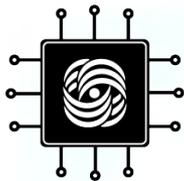
```
class test
{
    int i, n;
    float s;
    float x[n];

    public static void main( String
args[] )
    {
        n = 10;
        s = 0;
        for( i=1; i<=n; i++)
        {
            s = s + x[i-1];
            s = s / n;
        }
    }
}
```

Язык Java основан на C++

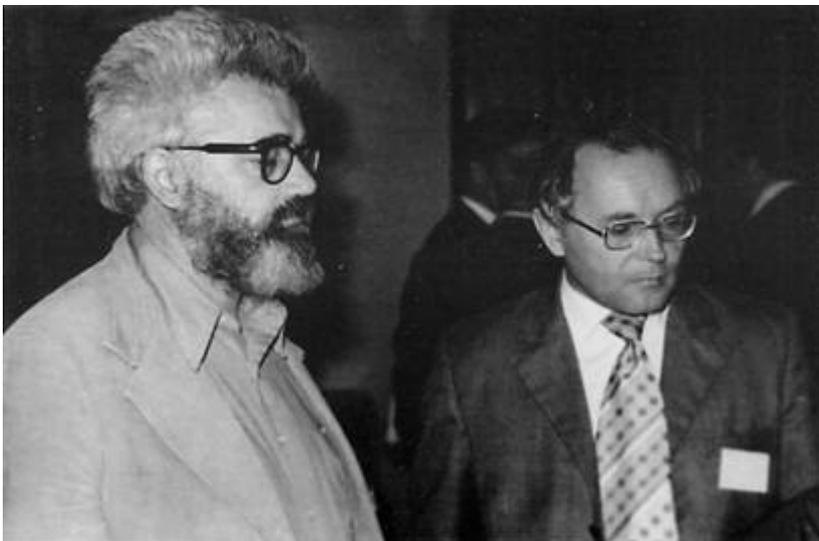
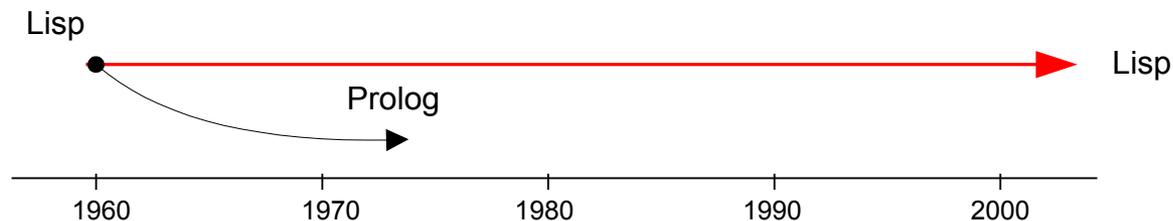


В качестве альтернативы Java  
корпорация Microsoft предложила  
язык C# (Си-шарп)



# Языки и системы программирования

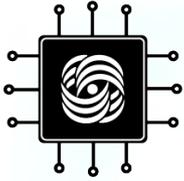
## Долгожитель Lisp – инструмент функционального программирования



Дж. Маккарти и А.П. Ершов  
Снимок 1975 г.

### **Lisp = LISt Processing**

Язык Lisp создан в 1960 году Джоном Маккарти (McCarthy, John; р. 1927 ) в Массачусетском технологическом институте на теоретическом фундаменте лямбда-исчисления, предложенного еще в 1930 году известным американским логиком Алонзо Черчем.



# Языки и системы программирования

## Долгожитель Lisp – инструмент функционального программирования

```
(setq L `(8 5 13 11 10))  
(defun sum (L)  
  (cond ((null L) '0)  
        (t (add (car L) (sum (cdr L)))))  
  )  
)  
  
(div (sum L) '5)
```

### **Примитивы:**

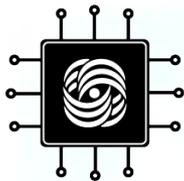
cond — условная функция, проверяющая с помощью функции null пустоту списка;

add — суммирование аргументов;

car — извлечение первого элемента из списка;

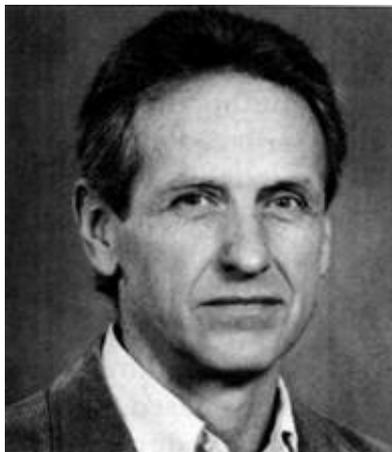
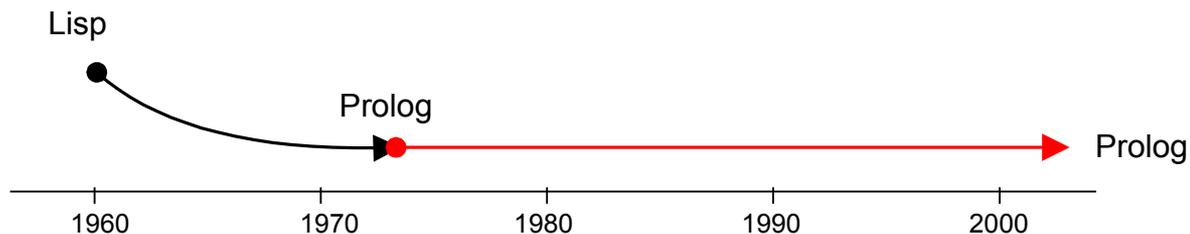
cdr — извлечение остатка списка (без первого элемента).

Программа на Lisp имеет специфический вид из-за обилия скобок. За это студенты прозвали его «Lots of Infuriating & Silly Parenthesis» - «Множество раздражающих и глупых скобок»



# Языки и системы программирования

## Prolog – несостоявшаяся мечта ЭВМ V поколения

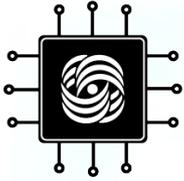


Prolog = PROgramming for LOGic

Теоретические основы языка были разработаны Робертом Ковальским (Kowalski, Robert) в Эдинбургском университете (Шотландия) в конце 1960-х годов

Первая практическая реализация языка осуществлена Аленом Кольмари (Colmerauer, Alain) в Марсельском университете (Франция) в 1972 г.





# Языки и системы программирования

## Prolog – несостоявшаяся мечта ЭВМ V поколения

### Факты:

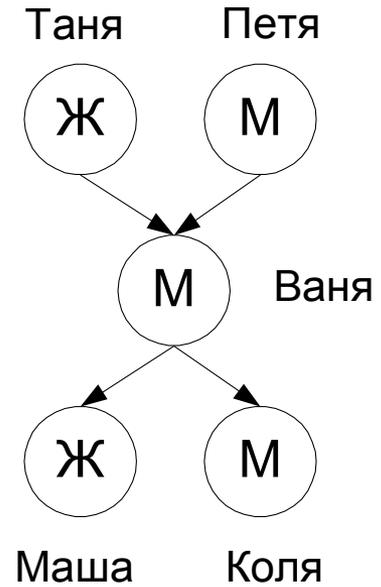
муж (петя), муж (ваня),  
муж (коля), жен (таня), жен (маша),  
мать (ваня, таня), отец (ваня, петя),  
отец (маша, ваня), отец (коля, ваня).

### Правила вывода:

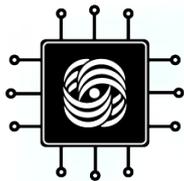
родитель (X, Y) :— отец (X, Y)  
родитель (X, Y) :— мать (X, Y)  
дед (X, Y) :— родитель (X, Z), отец (Z, Y)  
брат (X, Y) :— муж (Y), родитель (X, Z),  
родитель (Y, Z), X<>Y

### Примеры диалога:

GOAL> дед (коля, X) *Кто дед Коли?*  
X = Петя  
GOAL> брат (маша, X) *Кто брат Маши?*  
X = Коля

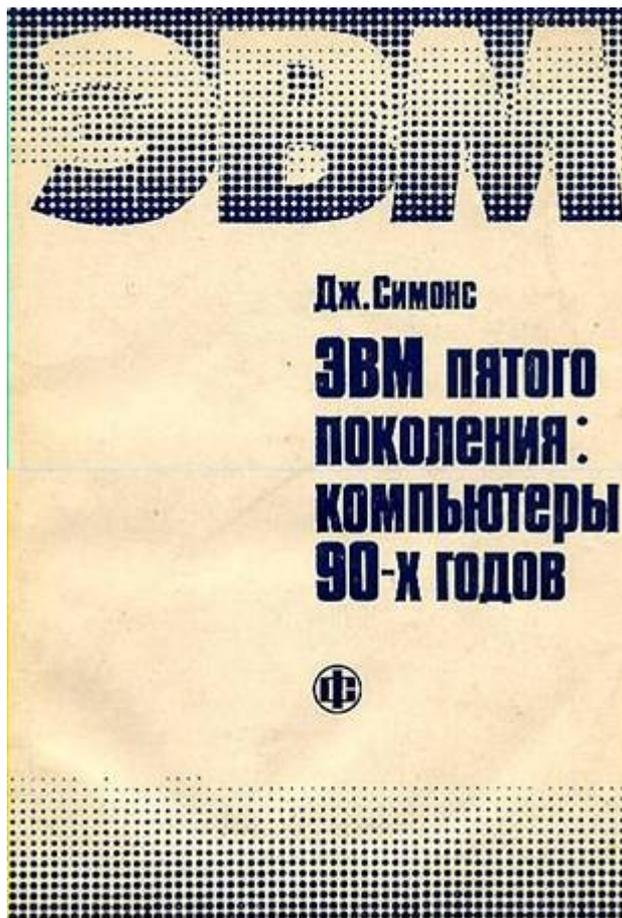


Описание  
предметной области  
семейных  
отношений на языке  
Prolog

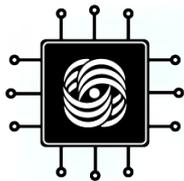


# Языки и системы программирования

## Prolog – несостоявшаяся мечта ЭВМ V поколения

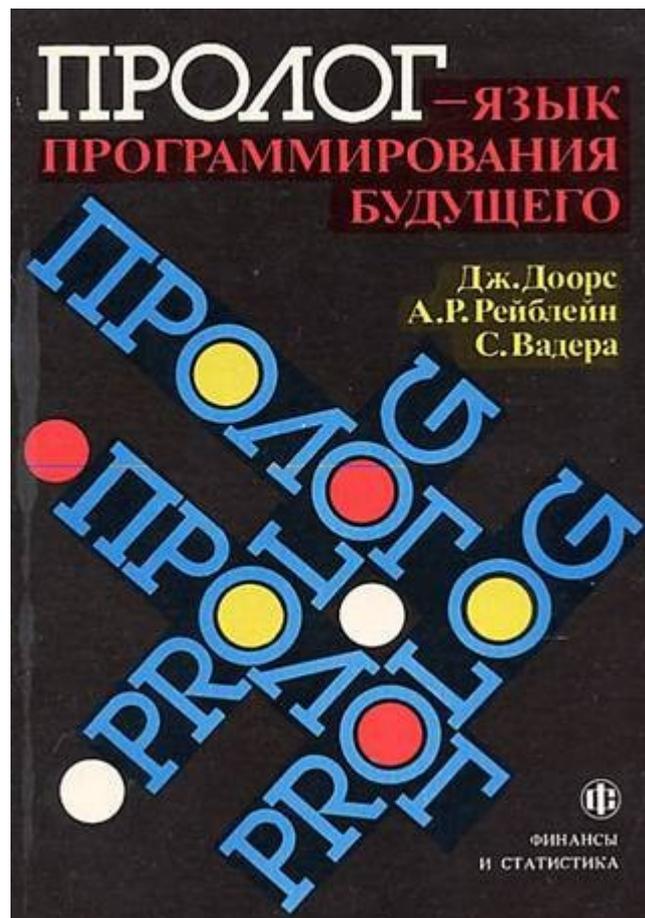


Проект ЭВМ V поколения – японский вызов мировой компьютерной индустрии, брошенный в начале 1980-х годов

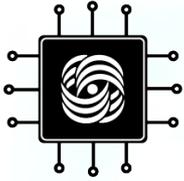


## 3.2. Языки и системы программирования

Prolog – несостоявшаяся мечта ЭВМ V поколения



В качестве основного языка ЭВМ V поколения предполагалось использовать Prolog



# Языки и системы программирования

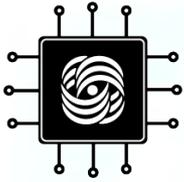
## Рефал – русский Пролог



Рефал = РЕкурсивных Функций  
АЛгоритмический

Теоретические основы языка были разработаны  
Валентином Турчиным в МФТИ в конце 1966  
году

```
Palindrom {  
  s.1 e.2 s.1 = <Palindrom e.2> ;  
  s.1 = True ;  
  = True;  
  e.1 = False ;  
} // Проверка палиндромности числа
```



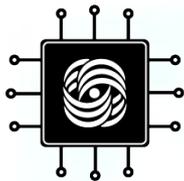
# РЕФАЛ

```
$ENTRY Go
```

```
{  
  = <Prout <Symb <FN <Numb <Card>>>>;  
}
```

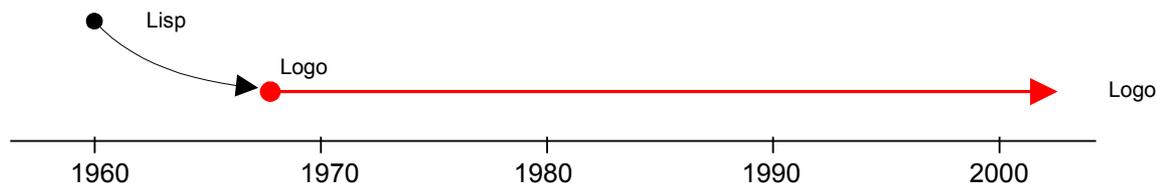
```
FN
```

```
{  
0 (s.1 s.2) = s.1;  
s.1 (s.2 s.3) = <FN <Sub s.1 1> (s.3 <Add s.2 s.3>)>;  
s.1 = <FN s.1 (0 1)>;  
} // Ввод N и вычисление числа Фибоначчи с  
  номером N
```



# Языки и системы программирования

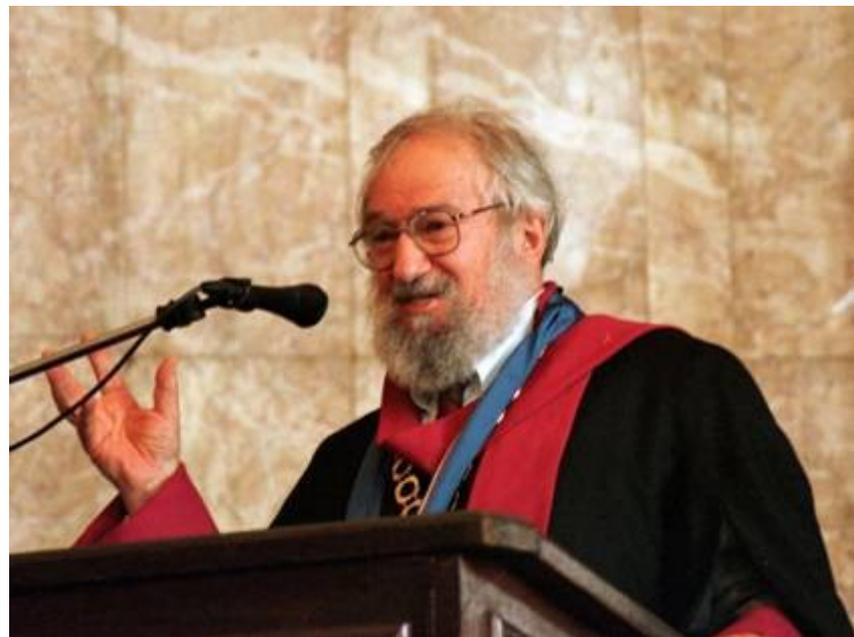
## Logo – язык для самых маленьких



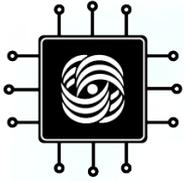
Язык **Logo**, изобретен в 1967 г. в MIT выдающимся математиком и педагогом Сеймуром Пейпертом (Papert, Seymour; р. 1928).

Пейперт в 1958-1963 годах работал в Женеве у знаменитого психолога Жана Пиаже (Piaget, Jean), где занимался детьми и природой их мышления.

Идейной основой Logo является язык Lisp



На фото: Сеймур Пейперт получает степень почетного доктора Софийского университета (1999 г.)



# Языки и системы программирования

## Logo – язык для самых маленьких

это дуга :шаг :число\_шагов

повтори :число\_шагов

[вперед :шаг направо 10]

Конец

это спираль :шаг

если :шаг < 1 [стоп]

дуга :шаг 18

спираль :шаг / 2

конец

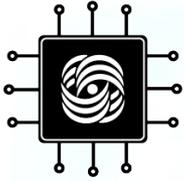
Рекурсия

Цикл



Процедура с  
параметром

Программа на Logo управляет черепашкой, оставляющей видимый след. С помощью зрительных образов интерпретируются все базовые структуры программирования

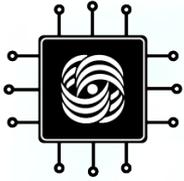


# Языки и системы программирования

## Уроки истории

### Основные парадигмы программирования:

- процедурное программирование (Fortran, Basic, Cobol, Algol, Pascal, Ada, C, Logo, FoxPro);
- объектно-ориентированное программирование (Simula, Smalltalk, Object Pascal, C++, Java, C#);
- визуально-событийное программирование (Visual Basic, Delphi, Visual C++, Visual Java, Visual FoxPro);
- функциональное программирование (Lisp, Рефал);
- логическое программирование (Prolog).



**Спасибо за внимание!**



# ИСТОРИЯ ПРИКЛАДНОЙ МАТЕМАТИКИ ЭВМ

## Лекция 13: *Биографии основных теоретиков Computer science*

ВМиК МГУ им. М.В. Ломоносова, Кафедра АСВК

Доцент Власов В.К.,

Ассистент Волканов Д.Ю.



## План лекции

- Алан Тьюринг
- Джон Фон Нейман
- Эдсгер Вайб Дейкстра
- Дональд Эрвин Кнут
- ЯП и их создатели



## Алан Тьюринг (1912 – 1954)



Для добавления текста  
щелкните мышью



## Основные результаты

- Тезис Чёрча-Тьюринга
- Машина Тьюринга
- Криптография
- Тест Тьюринга
- Колоссус

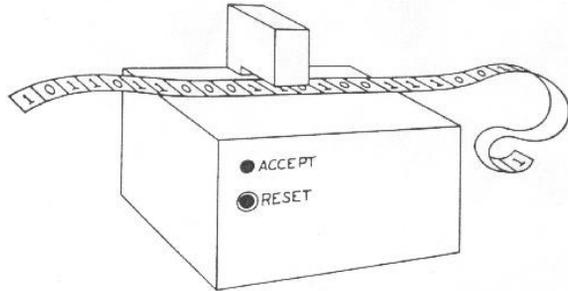


## Тезис Чёрча-Тьюринга

- Любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга



# Машина Тьюринга (1)





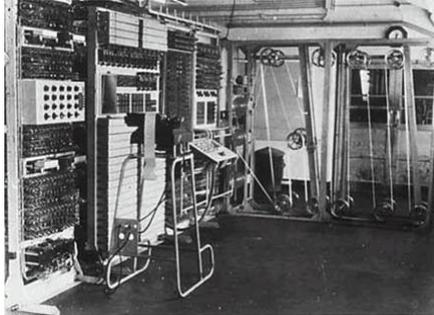
# Криптография

- Тьюринг помог взломать код Энигмы
- Построен первый программируемый компьютер Колоссус
- Базировался:
  - на его концепции универсальной машины 1936
  - потенциальной скорости и надёжности электронных технологий
  - неэффективность разностных машин для различных логических процессов
- Шифр-код был расшифрован в 1943
- Все компьютеры были разрушены по приказу Черчилля



# Colossus

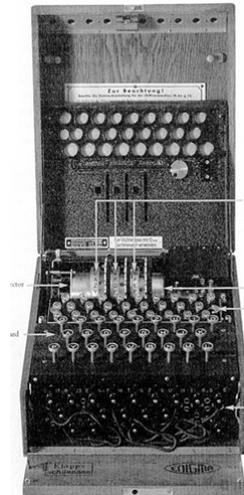
Под руководством выдающегося математика Алана Тьюринга была построена специализированная электронная вычислительная машина **Colossus**. Она насчитывала 2000 радиоламп и обрабатывала 25000 симв./с





# Enigma

В местечке Блечли-Парк (Bletchley Park) под Лондоном была организована сверхсекретная криптоаналитическая лаборатория для расшифровки немецких военных шифров, используемых в шифровальной машине [Enigma](#).





# Тест Тьюринга

- Опубликован в 1950 году
- Человек обменивается сообщениями на естественном языке с двумя собеседниками (человек и компьютер)
- Если человек не может определить кто есть кто, то считается что компьютер прошёл тест
- Переписка должна производиться через контролируемые промежутки времени
- Тьюринг оценил что программы в 2000 году пройдут тест
- Пока не подошли даже близко

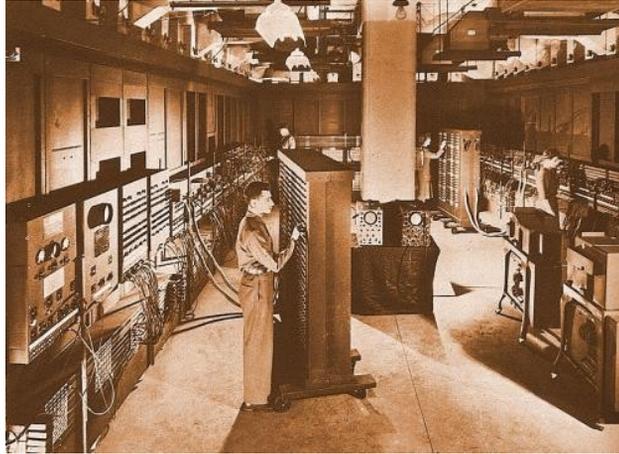


## Другие важные результаты

- Проблема самоприменимости МТ
- Универсальная МТ
- Основатель направления ИИ



## Первая ЭВМ ENIAC



Первая работающая ЭВМ **ENIAC (Electronic Numerical Integrator And Calculator)** была создана в 1945 г. в Пенсильванском университете. Длина 26 м, высота 6 м, масса 30 т. 18 000 ламп, 1500 реле, потребляемая мощность 150 квт.



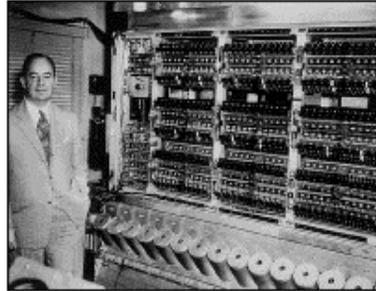
## Первая ЭВМ ENIAC



ЭВМ ENIAC. Вид сзади



## Проект фон Неймана и его вклад в архитектуру ЭВМ



Понятие «архитектура ЭВМ» связано с именем выдающегося математика XX столетия Джона фон Неймана (Neumann, John von; 1903-1957)



## Проект фон Неймана и его вклад в архитектуру ЭВМ

### ПРЕДВАРИТЕЛЬНОЕ РАССМОТРЕНИЕ ЛОГИЧЕСКОЙ КОНСТРУКЦИИ ЭЛЕКТРОННОГО ВЫЧИСЛИТЕЛЬНОГО УСТРОЙСТВА<sup>1)</sup>

*А. Беркс, Г. Голдстейн, Дж. Нейман*

#### **1. Основные компоненты машины**

**1.1.** Так как законченное устройство будет универсальной вычислительной машиной, оно должно содержать несколько основных органов, таких, как органы арифметики, памяти, управления и связи с оператором. Мы хотим, чтобы машина была полностью автоматической, т. е. чтобы после начала вычислений работа машины не зависела от оператора. Более полно это за-

**1.3.** Выше мы в принципе указали на два различных вида памяти — память чисел и память приказов. Если, однако, приказы машине свести к числовому коду и если машина сможет некоторым образом отличать число от приказа, то орган памяти можно использовать для хранения как чисел, так и приказов. Кодирование приказов в числовой форме рассматривается в п. 6. 3.

Фрагменты статьи фон Неймана с соавторами (русский перевод)



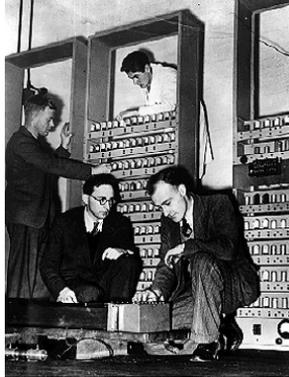
## Проект фон Неймана и его вклад в архитектуру ЭВМ

### Основные черты классической фон-неймановской архитектуры ЭВМ

2. Машина должна состоять из следующих основных блоков: арифметического устройства, оперативной памяти, устройства управления, устройства ввода, устройства вывода, устройства внешней памяти;
3. Команды программы должны храниться в оперативной памяти, откуда они последовательно выбираются и исполняются арифметическим устройством, система команд должна иметь операции условной и безусловной передачи управления. Команды должны рассматриваться как обычные данные, т.е. программа должна иметь возможность модифицировать себя в процессе вычислений;
4. Команды и данные должны храниться и обрабатываться в двоичной системе счисления.



## Проект фон Неймана и его вклад в архитектуру ЭВМ



Морис Уилкс у машины  
EDSAC. 3000 ламп, ОЗУ  
512 слов

Из-за разногласий в команде разработчиков реализация проекта фон Неймана в США затянулась.

Первая ЭВМ с хранимой программой **EDSAC (Electronic Delay Storage Automatic Calculator)** была построена в Англии в 1949 г. под руководством Мориса Уилкса (Wilkes, Maurice; р. 1913).

Английские ученые опирались на собственный опыт разработки электронных вычислительных устройств во время Второй мировой войны



## Другие значимые достижения

- Квантовая физика
- Функциональный анализ
- Теория множеств
- Создатель теории игр
- Создатель теории клеточных автоматов



# Эдсгер Вайб Дейкстра (1930 – 2002)



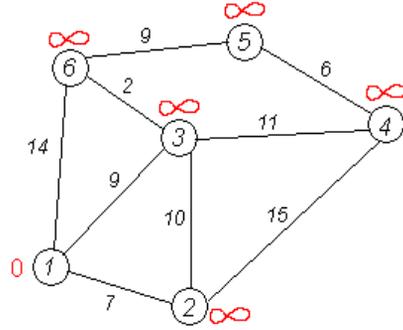
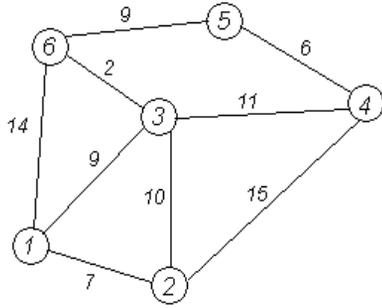


## Основные результаты

- Математическая логика
- Algol-60
- Концепция семафоров
- Алгоритм Дейкстра
- Борьба с оператором GOTO

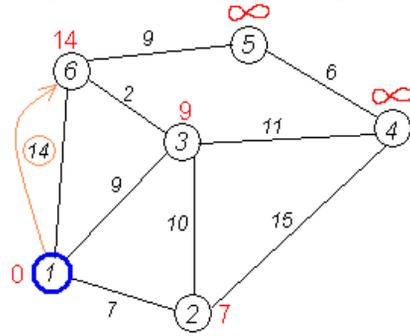
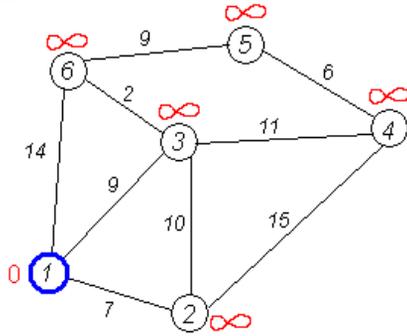


# Алгоритм Дейкстры (1)



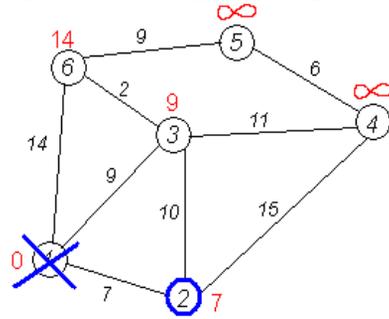
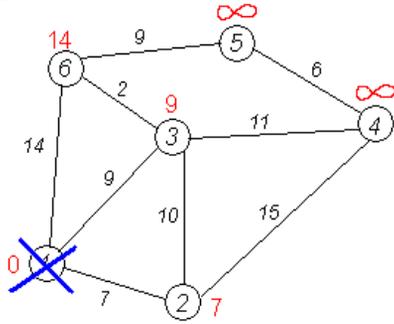


## Алгоритм Дейкстры (2)



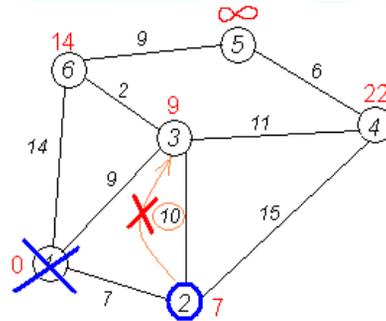
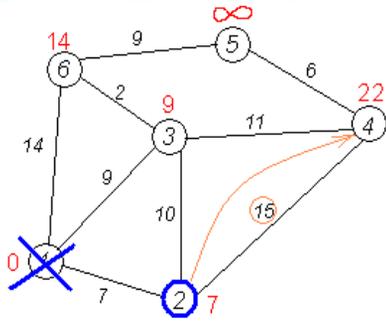


# Алгоритм Дейкстры (3)



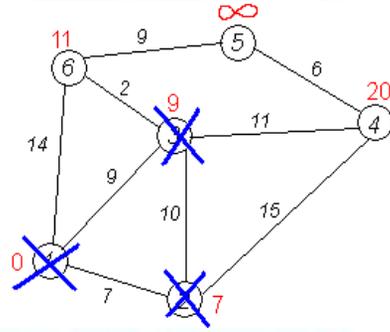
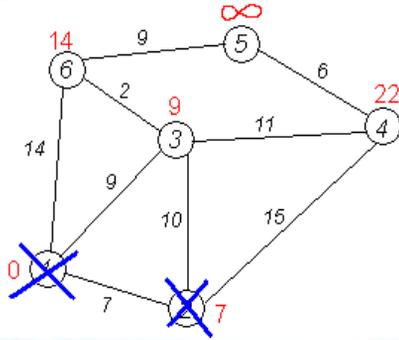


# Алгоритм Дейкстры (4)





# Алгоритм Дейкстры (5)





## Афоризмы (1)

- Студентов, ранее изучавших Бейсик, практически невозможно обучить хорошему программированию. Как потенциальные программисты они подверглись необратимой умственной деградации
- Вопрос «умеет ли компьютер думать» имеет не больше смысла, чем вопрос «умеет ли подводная лодка плавать»
- Проекты, предлагающие программирование на естественном языке, гибельны по своей сути

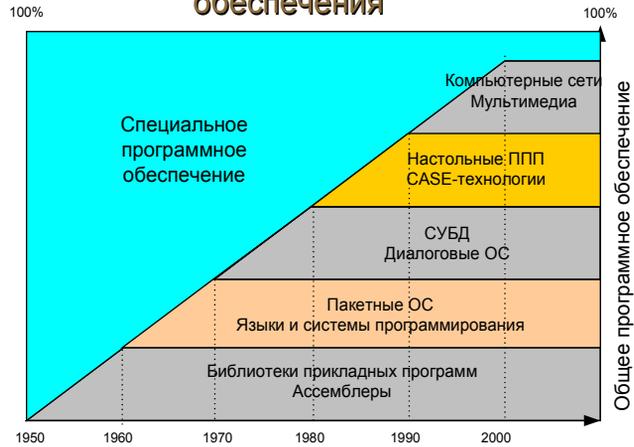


## Афоризмы (2)

- Дейкстра назвал модель IBM/360 (прообраз советской ЕС ЭВМ) — величайшей диверсией Запада против СССР
- На пустом диске можно искать вечно
- Если отладка — процесс удаления ошибок, то программирование должно быть процессом их внесения



## Классификация и эволюция программного обеспечения



Эволюция программного обеспечения. Подобно тому, как в океане из плавающей мути откладываются геологические пласты, из специального программного обеспечения с течением времени образуются слои общего ПО



## Языки и системы программирования Предыстория



Первая попытка создать высокоуровневый язык программирования принадлежит гениальному Конраду Цузе (конец 1940-х годов), разработавшему **Planalcul** (планировщик вычислений).

«Planalcul родился исключительно как результат теоретической работы, без всякой связи с тем, появится или нет в обозримом будущем машины, подходящие к программам на Planalcul».

Фрагмент рукописи Цузе с шахматной программой на языке Planalcul



## Языки и системы программирования

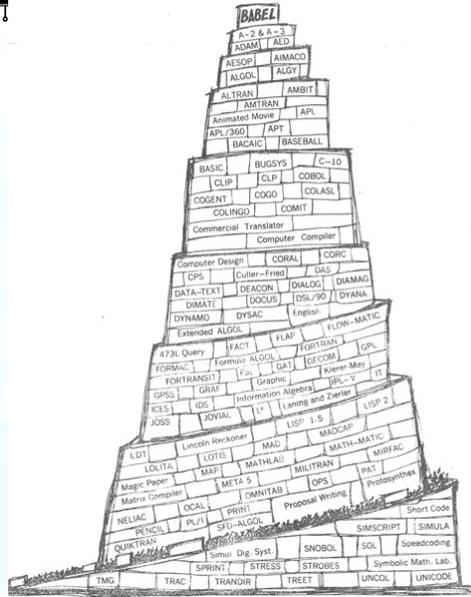
### Предыстория



Михаил Романович Шура-Бура и А.П. Ершов – создатели первых отечественных систем автоматизации программирования для ЭВМ «БЭСМ» и «Стрела» (1954-1956 годы)



## Языки и системы программирования



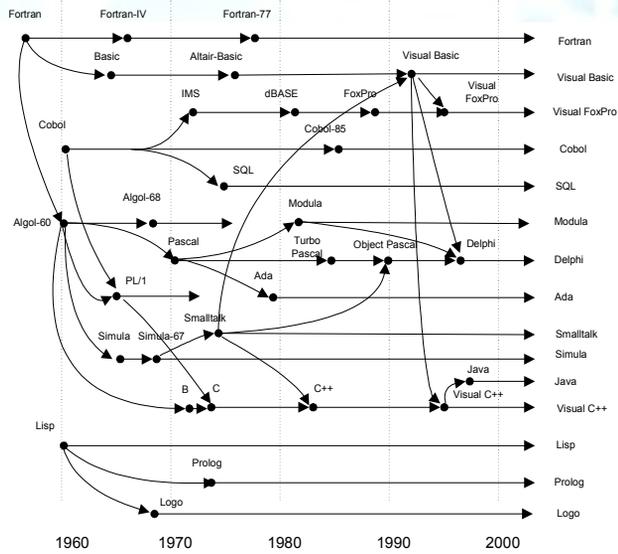
Наиболее активный период разработки языков и систем программирования приходится на 1960-е годы.

За это десятилетие в мире родилось более тысячи разнообразных языков, как универсальных, так и специализированных, но выжили и доросли до XXI века дожили немногие, в том числе бессмертные **Fotran**, **Basic**, **Algol**, **Cobol**, **Simula**, **Lisp** и их потомки.

На рисунке: «авилонская башня» языков программирования, созданных в 1960-е годы



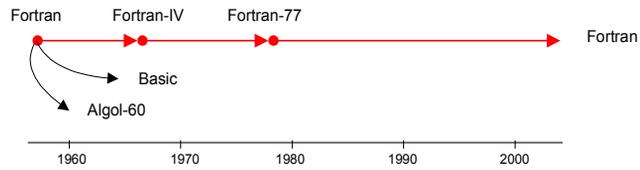
## Языки и системы программирования



Родословная основных высокоуровневых языков программирования



## Языки и системы программирования Бессмертный Fortran



### Fortran = FORMula TRANslator

Первый высокоуровневый язык программирования Fortran был разработан в фирме IBM под руководством Джона Бэкуса (Backus, John; р. 1924).

Работа над языком началась в 1954 г., первая реализация для IBM 704 в выполнена в 1957 г.



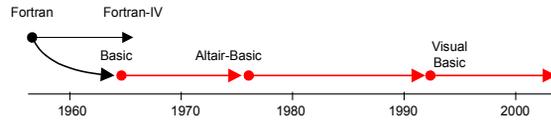
## Языки и системы программирования Бессмертный Fortran

```
C          MAIN PROGRAM
101      FORMAT(208)
102      FORMAT(//'N=',15, 5X, 'R=', 15
           1//6X, 'M', 5X, PROB)
103      FORMAT(18, F14.10)
201      READ(1,101) N, IR
           WRITE(3,102) N, IR
           IF(N) 202, 202, 203
202      STOP
203      IF(IR) 202, 202, 204
204      M=0
           P=COMBF(N,M)*COMBF(IR-1,N-M-1)
           1/COMBF(N+IR-1,IR)    ...
```

Фрагмент программы на языке Fortran



## Языки и системы программирования Basic – язык для начинающих



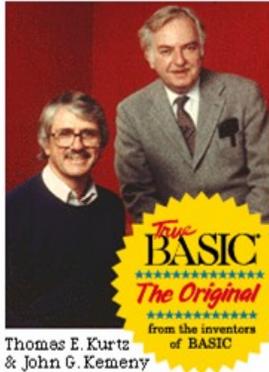
**BASIC = Beginners All-purpose Symbolic Instruction Code**

Язык Basic был разработан в 1964 г. в Дармутском колледже в г. Ханovere (Darmouth College, Hanover), штат Нью-Хемпшир



## Языки и системы программирования

### Basic – язык для начинающих



Авторы языка Basic.  
Стоит Джон Кемени  
(Kemeny, John G.; 1926-1993),  
сидит Томас Курц  
(Kurtz, Thomas E.; p. 1928)

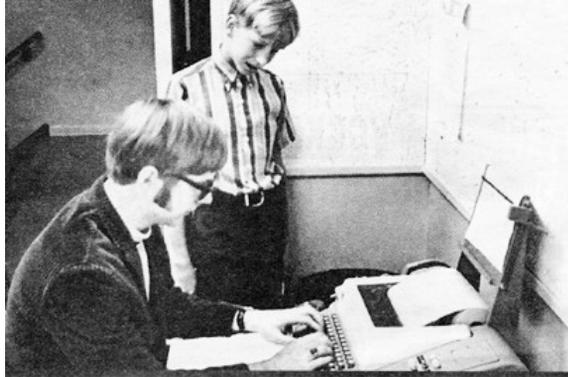
```
10 dim A(5)
20 for i=1 to 5
30 input A(i)
40 next i
50 if i=5 then goto 140
60 if A(i)<=A(i+1) then goto 90
70 i=i+1
80 goto 130
90 z=A(i)
100 A(i)=A(i+1)
110 A(i+1)=z
120 i=1
130 goto 50
140 for i=1 to 5
150 print A(i)
160 next i
```

Простейшая  
программа на  
языке Basic



## Языки и системы программирования

### Basic – язык для начинающих



Будущие создатели Microsoft Пол Аллен (Allen, Paul; р. 1954) и Билл Гейтс (Gates, William; р. 1955) познакомились с Бэйсиком, работая в компьютерном классе школы в Сиэтле (снимок 1968 г.)



## Языки и системы программирования

### Basic – язык для начинающих

Начав с Бэйсика, компания Microsoft превратилась в крупнейшую софтверную империю, а Билл Гейтс – стал самым богатым человеком на планете

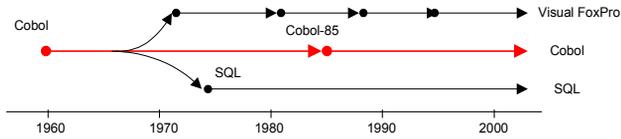


Штаб - квартира  
корпорации Microsoft в  
Редмонде (пригород  
Сиэтла)

30



## Языки и системы программирования Cobol – язык для бухгалтеров



### **COBOL = COmmon Business-Oriented Language**

На фото: разработчики языка Cobol у шуточного обелиска, присланного в их адрес в качестве намека на безнадежно медленную работу, способную похоронить саму идею. Справа внизу – Грейс Хоппер



## Языки и системы программирования

### Cobol – язык для бухгалтеров

Основные свойства языка Cobol:

- независимость программ от оборудования;
- независимость программ от данных;
- сложные структуры данных;
- синтаксис, приближенный к естественному английскому языку.



## 3.2. Языки и системы программирования

### Cobol – язык для бухгалтеров

#### 1010 IDENTIFICATION DIVISION.

1020 PROGRAM-ID "EXAMPLE".

#### 1030 ENVIROMENT DIVISION.

1040 INPUT-OUTPUT SECTION.

1050 FILE-CONTROL.

1060 SELECT CD ASSIGN TO "SYS010" UNIT-RECORD 2540R.

1070 SELECT TT ASSIGN TO "SYS009" UTILITY 2400.

#### 1080 DATA DIVISION.

1090 FILE SECTION.

1100 FD CD DATA RECORD IS C

1110 LABEL RECORDS ARE OMITTED.

1120 01 C.

1130 02 C1 PICTURE 9(4).

1140 02 C2 PICTURE 9.

1150 02 C3 PICTURE X(70).

...

Программа на Коболе  
(начало)

71



## 3.2. Языки и системы программирования

### Cobol – язык для бухгалтеров

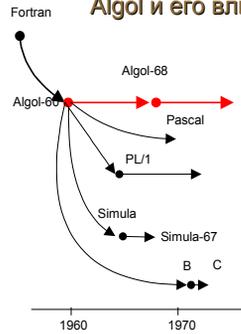
#### 1290 PROCEDURE DIVISION.

```
1300 P1.  OPEN INPUT CD, OUTPUT TT.  
1310 P2.  READ CD, AT END GO TO P3.  
1320          MOVE C1 TO D1.  
1330          MONE C2 TO D2.  
1340          MOVE C3 TO D3.  
1350          ADD C1, C2, GIVING D4.  
1360          WRITE T FROM D.  
1370          GO TO P2.  
1380 P3.  CLOSE SD, TT.  
1390          STOP RUN.
```

Программа на Коболе (окончание)



## Языки и системы программирования Algol и его влияние на языки программирования



### ALGOL = ALGORitmic Language

В 1958 году в Цюрихе (Швейцария) состоялась международная конференция, предложившая проект нового универсального международного языка программирования Algol-58. В 1960 году на парижской конференции была принята окончательная версия под названием Algol-60.

На снимке: участники парижской конференции голосуют за Алгол-60.



## Языки и системы программирования Algol и его влияние на языки программирования

Основные свойства языка Algol-60:

- машинная независимость;
- формальный синтаксис;
- описание переменных и блочная структура;
- рекурсия

Нормальная форма Бэкуса-Наура (БНФ)

$\langle \text{цифра} \rangle ::= 1|2|3|4|5|6|7|8|9|0$

$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \langle \text{целое без знака} \rangle$

TT



## Языки и системы программирования Algol и его влияние на языки программирования

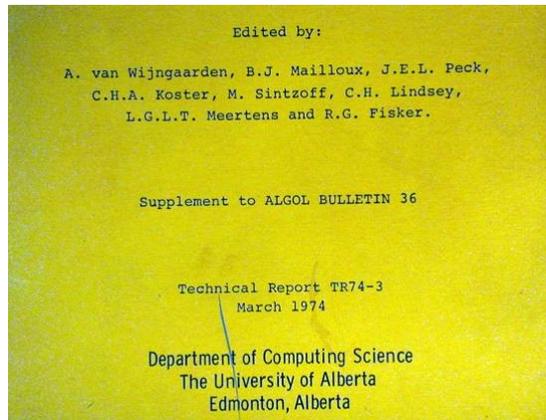
```
begin  
    integer i, n;  
    real s;  
    real array x[1:n];  
    s:=0;  
    for i:=1 step 1 to n do  
        s:=s+x[i];  
    s:=s/n  
end
```

Простейшая программа на Алголе-60, вычисляющая среднее арифметическое n чисел.

Синтаксис Алгола-60 сформировал стандарт для всех последующих языков программирования



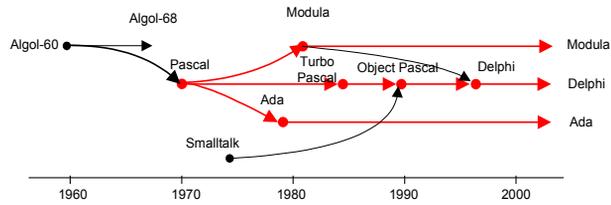
## Языки и системы программирования Algol и его влияние на языки программирования



В результате многолетней переработки Алгола-60 комитетом [IFIP](#) появился язык [Алгол-68](#) (пересмотренное сообщение под ред. А. ван Вейнгаардена (A. van Wijngaarden) и др. опубликовано в 1975 г.)



## Языки и системы программирования Pascal и его потомки



Член комитета по Алголу-68 Никлаус Вирт (Wirth, Niklaus; р. 1934) был против принятия переусложненного стандарта.

В знак доказательства своей правоты он разработал в 1971 г. простой и ясный алгоподобный язык, предназначенный прежде всего для обучения студентов в Федеральном техническом университете в Швейцарии. В честь изобретателя первой вычислительной машины Вирт назвал язык

**Паскалем.**





## Языки и системы программирования Pascal и его потомки

```
var
    i, n: integer;
    s: float;
    x: array[1..n] of real;

begin
    s:=0;
    for i:=1 to n do
        s:=s+x[i];
    s:=s/n

end.
```

Программа на Паскале, вычисляющая среднее  
арифметическое  $n$  чисел

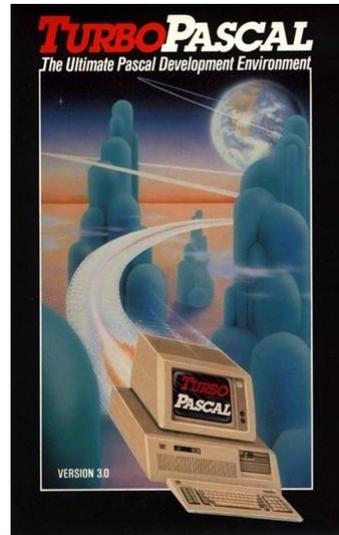
70



## Языки и системы программирования Pascal и его потомки

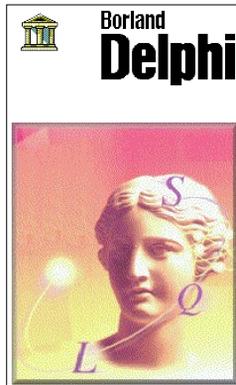


Новую жизнь языку Pascal дал Филипп Кан (Kahn, Philippe; р. 1938) – создатель компилятора Turbo Pascal для IBM PC и основатель компании Borland (1984 г.)



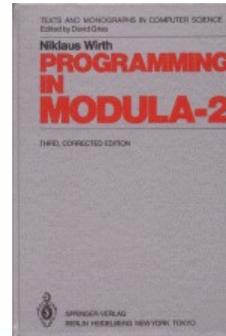


## Языки и системы программирования Pascal и его потомки



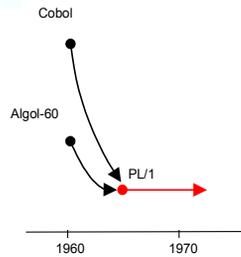
Среда разработки Delphi фирмы Borland объединила передовые достижения технологии программирования: объектное расширение языка Pascal, визуально-событийное проектирование, модульное структурирование и отдельная компиляция.

В отличие от учебного Паскаля, язык программирования Modula-2, предложенные Никлаусом Виртом, изначально предназначался для профессионального применения





## Языки и системы программирования



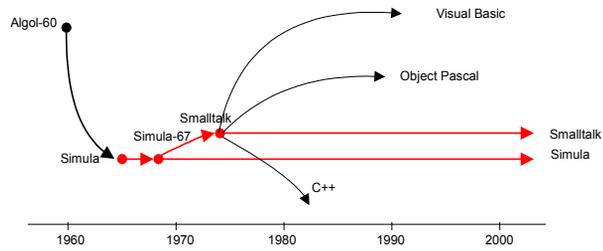
### PL/1 = Programming Language One

Язык PL/1 был частью амбициозного проекта IBM S/360, он создавался в спешке и представлял собой механическую смесь идей из многих языков. Критики сравнивали его с елкой со множеством украшений.

```
EXAMPLE: PROCEDURE OPTIONS (MAIN);
ON ENDFILE (SYSIN) GO TO ENDING;
P1:   GET LIST (A, B, C);
      D = B*B - 4*A*C;
      E = -B/(A+A);
      IF D<0 THEN DO;
        X1, X2 = E;
        Y1 = SQRT(-D)/(A+A);
      END;
      ELSE DO;
        R = SQRT(D)/(A+A);
        ...
        Y1 = 0;
      END;
      Y2 = -Y1;
      PUT LIST (X1, Y1, X2, Y2);
      GO TO P1;
ENDING;;
END EXAMPLE;
```



## Языки и системы программирования Simula и Smalltalk – революция в программировании



### Simula = SIMULAtion

За разработку языка Simula Кристен Нигорд (Nygaard, Kristen; 1926-2002), на снимке слева, и Оле-Йохан Дал (Dahl, Ole-Johan; 1931-2002) были удостоены высшей награды компьютерного сообщества – медали Тьюринга



## Языки и системы программирования Simula и Smalltalk – революция в программировании

```
|a|
```

```
a := Array new: 5.
```

```
1 to: 5 do: [:i | a at: i put:
```

```
(Prompter prompt: 'Введите элемент массива')  
asNumber].
```

```
a := a asSortedCollection.
```

```
a do: [:i | Transcript putAll: i printString].
```

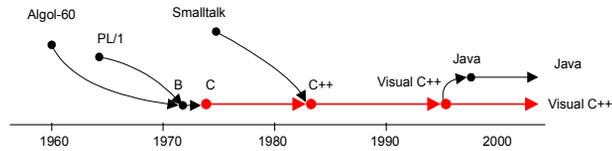
Простейшая программа  
на Smalltalk,  
вычисляющая среднее  
арифметическое пяти  
чисел



Алан Кей



## Языки и системы программирования С – язык для профессионалов



Язык **Си (C)** был создан Деннисом Ричи (Ritchie, Dennis M.; р. 1941) в 1973 году в Bell Labs в ходе разработки операционной системы UNIX. Он развивал язык **Би (B)**, который основывался на созданном в Кембриджском университете языке **BCPL (от Basic Combined Programming Language)**, который в свою очередь был потомком Алгола-60



## Языки и системы программирования

### C – язык для профессионалов

```
float A[5];
for(int i=0;i<5;i++)scanf("%f",&A[i]);
i=0;
while(i<4){
    if(A[i]<=A[i+1])i++;
    else{
        z=A[i];
        A[i]=A[i+1];
        A[i+1]=z;
        i=0;
    }
};
for(i=0;i<5;i++)printf("%f\n",A[i]);
```

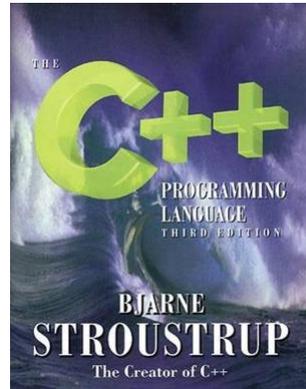
Текст на языке C отличается лаконичностью

»»



## Языки и системы программирования

C – язык для профессионалов



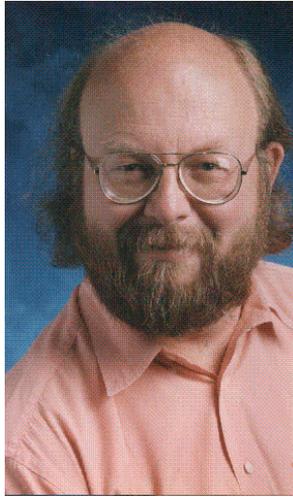
Бьярн Страуструп (Stroustrup, Bjarne; р. 1950) ввел в язык C объекты и превратил его в C++

50



## Языки и системы программирования

### Java – дитя интернета



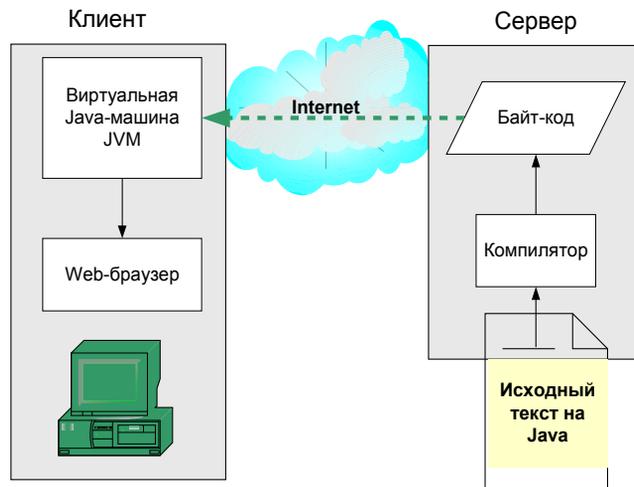
В 1995 г. фирма Sun Microsystems представила язык **Java** для программирования в интернете.

Он возник в ходе реализации проекта **Oak** («Дуб»), целью которого было создание системы программирования бытовых микропроцессорных устройств.

Джеймс Гослинг (Gosling, James) – автор Java.



## Языки и системы программирования Java – дитя интернета



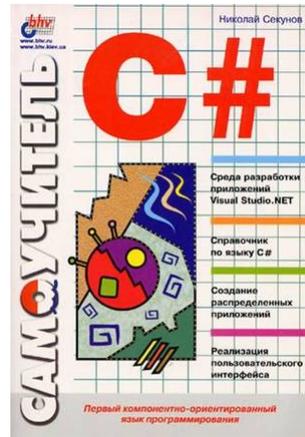
Java - технология



## Языки и системы программирования Java – дитя интернета

```
class test
{
    int i, n;
    float s;
    float x[n];
    public static void main( String
args[] )
    {
        n = 10;
        s = 0;
        for( i=1; i<=n; i++)
        {
            s = s + x[i-1];
            s = s / n;
        }
    }
}
```

Язык Java основан на C++

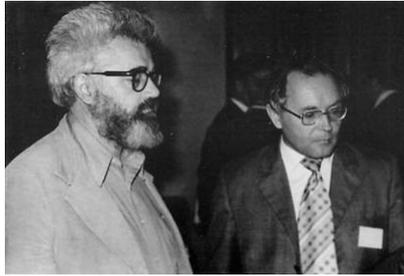
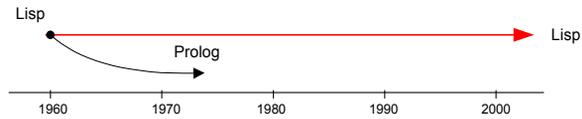


В качестве альтернативы Java  
корпорация Microsoft предложила  
язык C# (Си-шарп)



## Языки и системы программирования

### Долгожитель Lisp – инструмент функционального программирования



Дж. Маккарти и А.П. Ершов  
Снимок 1975 г.

#### Lisp = LISt Processing

Язык Lisp создан в 1960 году Джоном Маккарти (McCarthy, John; р. 1927 ) в Массачусетском технологическом институте на теоретическом фундаменте лямбда-исчисления, предложенного еще в 1930 году известным американским логиком Алонзо Черчем.



## Языки и системы программирования

### Долгожитель Lisp – инструмент функционального программирования

```
(setq L `(8 5 13 11 10))  
(defun sum (L)  
  (cond ((null L) '0)  
        (t (add (car L) (sum (cdr L)))))  
  )  
)  
(div (sum L) '5)
```

**Примитивы:**

cond — условная функция, проверяющая с помощью функции null пустоту списка;

add — суммирование аргументов;

car — извлечение первого элемента из списка;

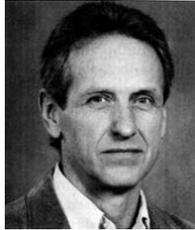
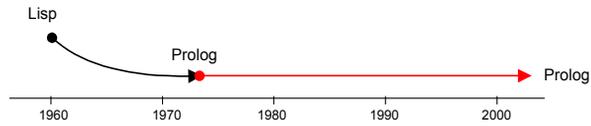
cdr — извлечение остатка списка (без первого элемента).

Программа на Lisp имеет специфический вид из-за обилия скобок.  
За это студенты прозвали его «Lots of Infuriating & Silly Parenthesis» -  
«Множество раздражающих и глупых скобок»



## Языки и системы программирования

### Prolog – несостоявшаяся мечта ЭВМ V поколения



Prolog = PROgramming for LOGic

Теоретические основы языка были разработаны Робертом Ковальским (Kowalski, Robert) в Эдинбургском университете (Шотландия) в конце 1960-х годов

Первая практическая реализация языка осуществлена Аленом Кольмари (Colmerauer, Alain) в Марсельском университете (Франция) в 1972 г.





## Языки и системы программирования

### Prolog – несостоявшаяся мечта ЭВМ V поколения

#### Факты:

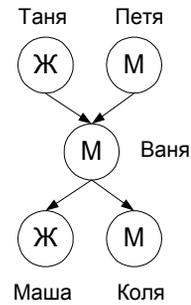
муж (петя), муж (ваня),  
муж (коля), жен (таня), жен (маша),  
мать (ваня, таня), отец (ваня, петя),  
отец (маша, ваня), отец (коля, ваня).

#### Правила вывода:

родитель (X, Y) :- отец (X, Y)  
родитель (X, Y) :- мать (X, Y)  
дед (X, Y) :- родитель (X, Z), отец (Z, Y)  
брат (X, Y) :- муж (Y), родитель (X, Z),  
родитель (Y, Z), X<>Y

#### Примеры диалога:

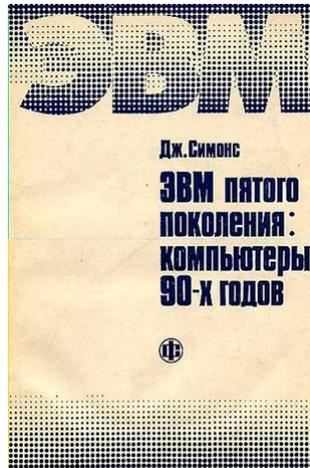
GOAL> дед (коля, X) *Кто дед Коли?*  
X = Петя  
GOAL> брат (маша, X) *Кто брат Маши?*  
X = Коля



Описание  
предметной области  
семейных  
отношений на языке  
Prolog



## Языки и системы программирования Prolog – несостоявшаяся мечта ЭВМ V поколения

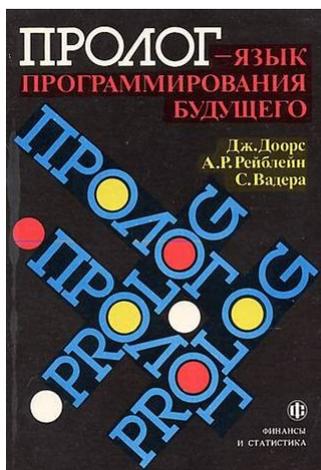


Проект ЭВМ V поколения – японский вызов мировой компьютерной индустрии, брошенный в начале 1980-х годов



### 3.2. Языки и системы программирования

Prolog – несостоявшаяся мечта ЭВМ V поколения



В качестве основного языка ЭВМ V поколения предполагалось использовать Prolog



## Языки и системы программирования

### Рефал – русский Пролог



Рефал = РЕкурсивных Функций  
АЛгоритмический

Теоретические основы языка были разработаны  
Валентином Турчиным в МФТИ в конце 1966  
году

```
Palindrom {  
    s.1 e.2 s.1 = <Palindrom e.2> ;  
    s.1 = True ;  
    = True ;  
    e.1 = False ;  
} // Проверка палиндромности числа
```



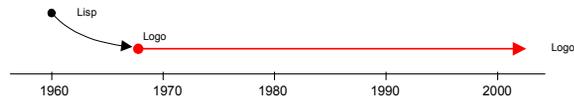
# РЕФАЛ

```
$ENTRY Go
{
  = <Prout <Symb <FN <Numb <Card>>>>;
}

FN
{
  0 (s.1 s.2) = s.1;
  s.1 (s.2 s.3) = <FN <Sub s.1 1> (s.3 <Add s.2 s.3>)>;
  s.1 = <FN s.1 (0 1)>;
} // Ввод N и вычисление числа Фибоначчи с
    номером N
```



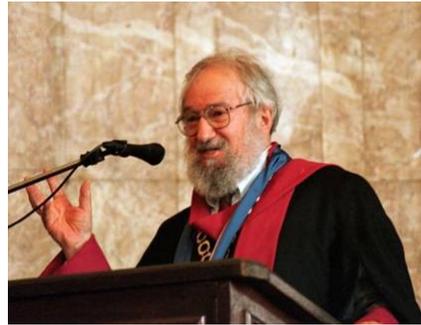
## Языки и системы программирования Logo – язык для самых маленьких



Язык **Logo**, изобретен в 1967 г. в МИТ выдающимся математиком и педагогом Сеймуром Пейпертом (Papert, Seymour; р. 1928).

Пейперт в 1958-1963 годах работал в Женеве у знаменитого психолога Жана Пиаже (Piaget, Jean), где занимался детьми и природой их мышления.

Идейной основой Logo является язык Lisp



На фото: Сеймур Пейперт получает степень почетного доктора Софийского университета (1999 г.)



## Языки и системы программирования Logo – язык для самых маленьких

```
это дуга :шаг :число_шагов  
  повтори :число_шагов  
    [вперед :шаг направо 10]  
Конец
```

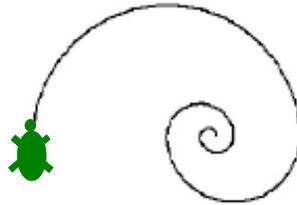
Цикл

```
это спираль :шаг  
  если :шаг < 1 [стоп]  
  дуга :шаг 18  
  спираль :шаг / 2  
конец
```

Процедура с  
параметром

Рекурсия

Программа на Logo управляет черепашкой, оставляющей видимый след. С помощью зрительных образов интерпретируются все базовые структуры программирования





## Языки и системы программирования

### Уроки истории

#### Основные парадигмы программирования:

- процедурное программирование (Fortran, Basic, Cobol, Algol, Pascal, Ada, C, Logo, FoxPro);
- объектно-ориентированное программирование (Simula, Smalltalk, Object Pascal, C++, Java, C#);
- визуально-событийное программирование (Visual Basic, Delphi, Visual C++, Visual Java, Visual FoxPro);
- функциональное программирование (Lisp, Рефал);
- логическое программирование (Prolog).



**Спасибо за внимание!**