

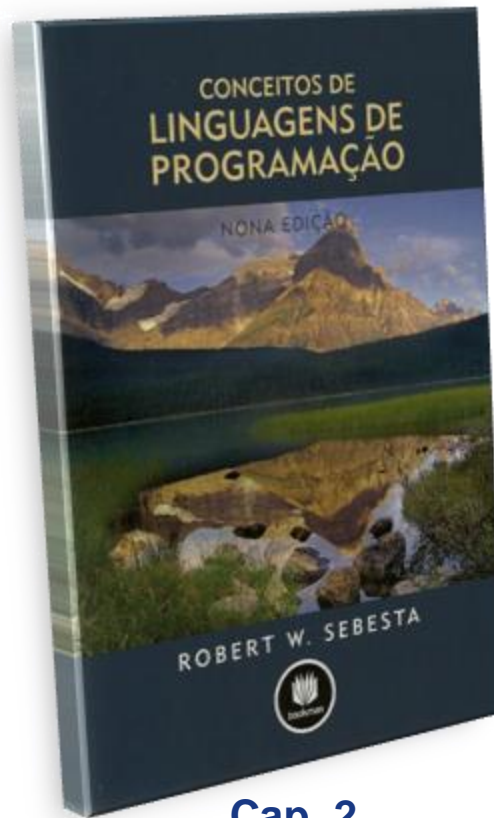


Evolução das Linguagens de Programação

História

Prof. Ausberto S. Castro V.
ascv@uenf.br

Bibliografia



Cap. 2

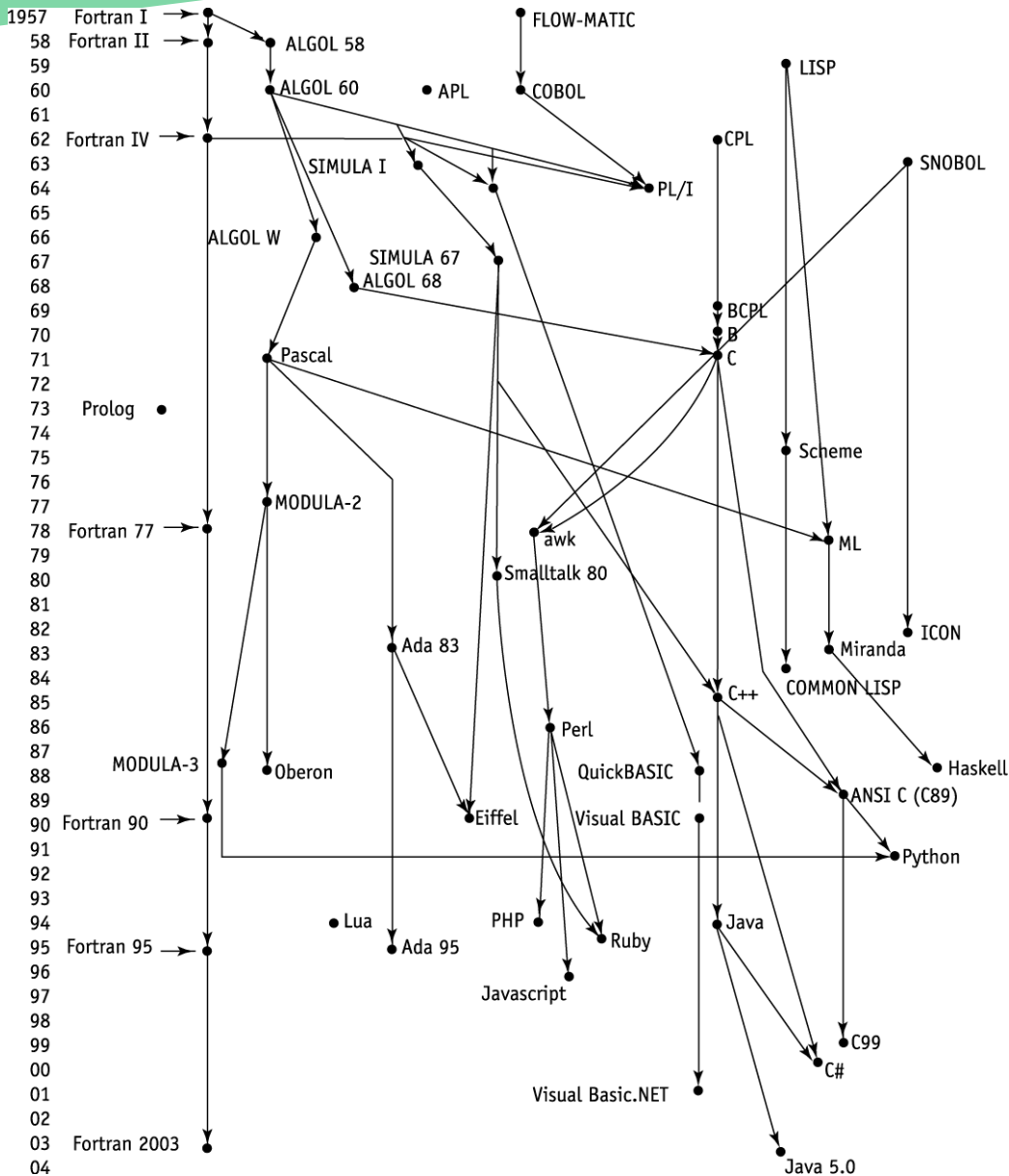
Evolução das Linguagens

- ❑ A primeira linguagem : Plankalkül
- ❑ Programação de Hardware Mínima: Pseudocódigos
- ❑ O IBM 704 e o FORTRAN
- ❑ Programação Funcional: LISP
- ❑ O primeiro passo rumo à sofisticação: ALGOL 60
- ❑ Informatizando registros comerciais: COBOL
- ❑ O início do Compartilhamento de Tempo: BASIC
- ❑ Tudo para Todos: PL/I
- ❑ Duas primeiras Linguagens Dinâmicas: APL e SNOBOL
- ❑ A origem da Abstração de Dados: SIMULA 67

Evolução das Linguagens

- ❑ **Projeto Ortogonal: ALGOL 68**
- ❑ **Programação baseada em Lógica: PROLOG**
- ❑ **O maior esforço de projeto da História: ADA**
- ❑ **Programação Orientada a Objetos: SMALLTALK**
- ❑ **Recursos Imperativos + Objetos: C++**
- ❑ **Programação Científica: MATLAB, Scilab, Fortran 90**
- ❑ **Programação Paralela: FORTRAN 90/95**
- ❑ **Programando a WWW: JAVA**
- ❑ **Outros:**
 - **Processamento de texto: LaTeX**
 - **Hipertexto: HTML**

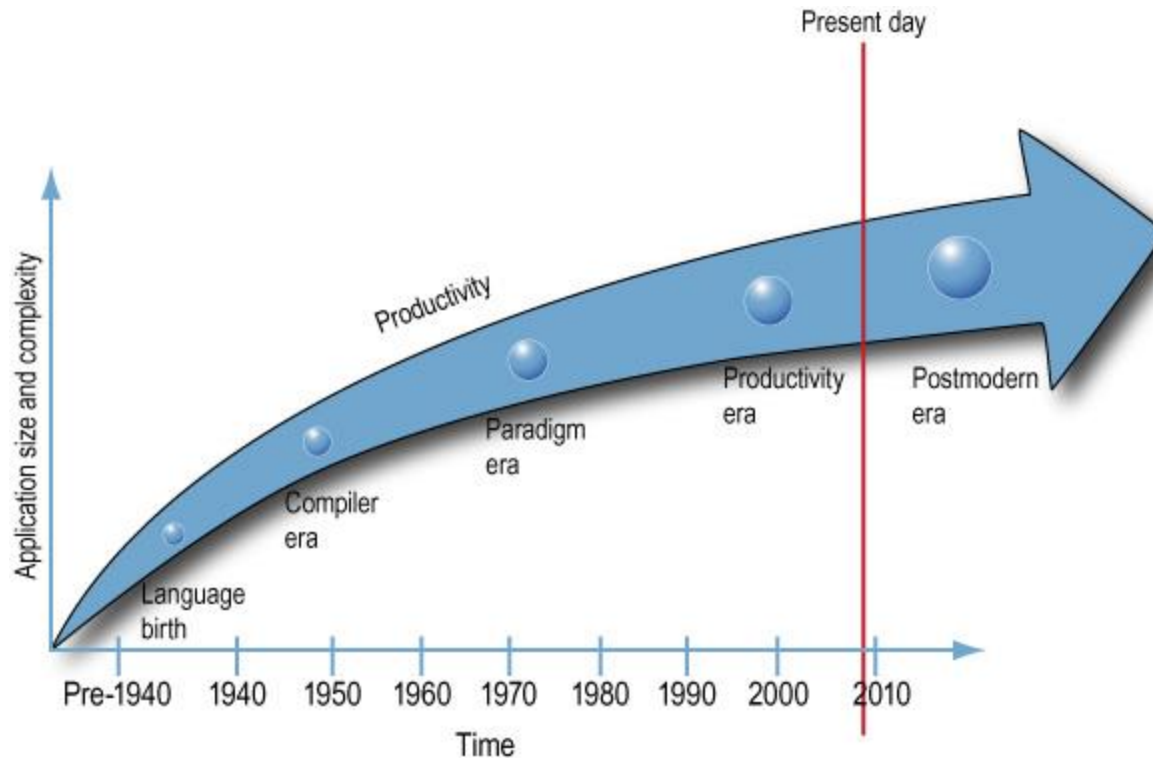
Genealogia abreviada



Linguagens mais influentes

- ❑ 1957 FORTRAN
- ❑ 1958 ALGOL
- ❑ 1960 LISP
- ❑ 1960 COBOL
- ❑ 1962 APL
- ❑ 1962 SIMULA
- ❑ 1964 BASIC
- ❑ 1964 PL/I
- ❑ 1966 ISWIM
- ❑ 1970 Prolog
- ❑ 1972 C
- ❑ 1975 Pascal
- ❑ 1975 Scheme
- ❑ 1977 OPS5
- ❑ 1978 CSP
- ❑ 1978 FP
- ❑ 1980 dBASE II
- ❑ 1983 Smalltalk-80
- ❑ 1983 Ada
- ❑ 1983 Parlog
- ❑ 1984 Standard ML
- ❑ 1986 C++
- ❑ 1986 CLP(R)
- ❑ 1986 Eiffel
- ❑ 1988 CLOS
- ❑ 1988 Mathematica
- ❑ 1988 Oberon
- ❑ 1989 HTML
- ❑ 1990 Haskell
- ❑ 1990 Python
- ❑ 1991 Java
- ❑ 1993 Ruby
- ❑ 1995 PHP
- ❑ 2000 C#

The New Era of Programming Languages



<http://techdistrict.kirrk.com/2009/06/17/the-new-era-of-programming-languages/>



1. A linguagem Plankalkül - 1945

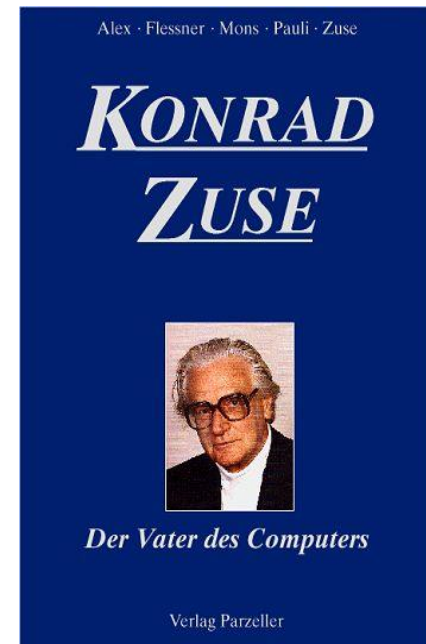
- ❑ 1936-1945, Konrad Zuse (Alemanha), construiu uma série de computadores a partir de relés mecânicos. A guerra destruiu todos, menos o Z4
- ❑ 1945 – proposta de dissertação Ph.D. , criou a linguagem PLANKALKÜL = cálculo de programa
- ❑ Nunca foi implementada
- ❑ Estruturas de dados avançadas
 - Ponto-flutuante, arrays e registros
- ❑ Notação: cada instrução, três linhas

A(8) := 3*B(6)

| 3 * B => A

V | 6 8 (subscritos)

S | 1:n 1:n (tipo de dado) 1:n inteiro de n bits



2. Pseudocódigo - 1949

- ❑ **Problemas com códigos de máquina (binária)**
 - Tediosa e com muitos erros
 - Códigos numéricos para instruções: 14 25 para ADD 25
 - Endereçamento absoluto
- ❑ **Primeira linguagem: Short Code, 1949**
 - Proposto por John Mauchly para o Computador BINAC
 - Implementado como interpretador por W.F. Schmitt e executado no UNIVAC I Serial 1 em 1950
 - Expressões codificadas, esquerda para direita
 - Implementada como um interpretador
 - Programação automática

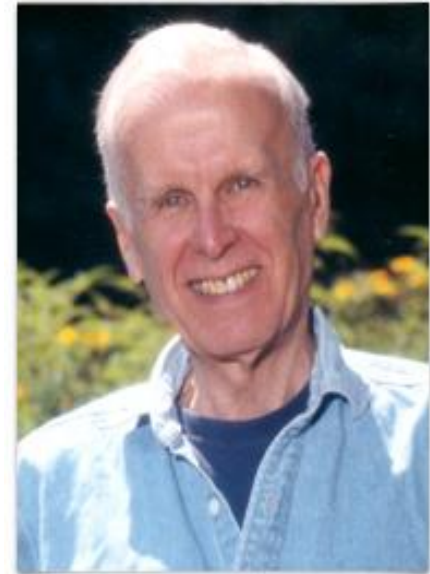
01 -	06 valor absoluto
02)	07 +
03 =	08 pause
04 /	09 (

x0 = sqrt(abs(y0))

00 x0 03 20 06 y0

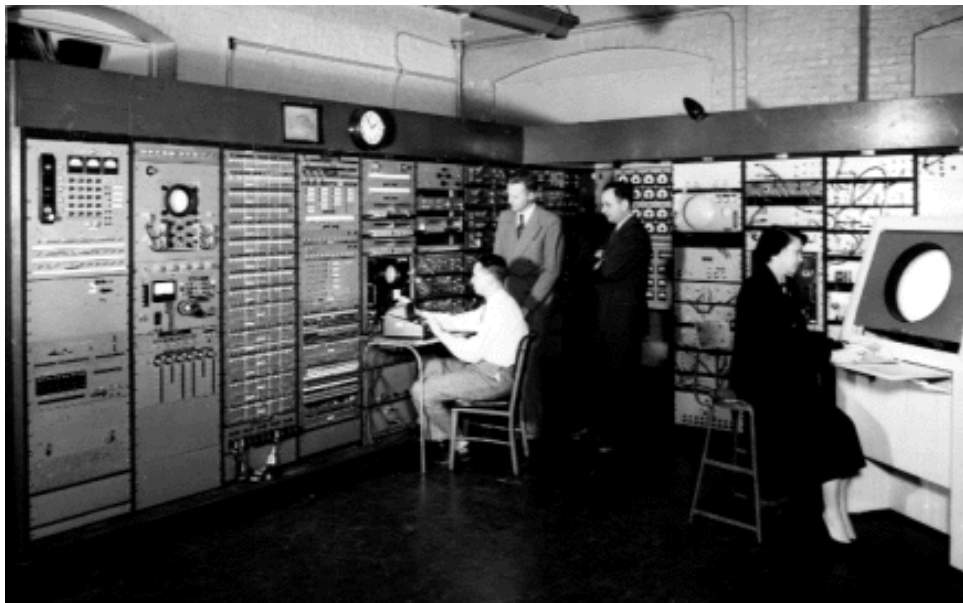
Pseudocódigo

- ❖ **Speedcoding, 1954, desenvolvido por John Backus para o IBM 701**
 - Pseudo-instruções para as operações e funções matemáticas
 - Ramificações condicionais e incondicionais
 - Auto-incremento dos **registradores** para acesso de arrays
 - Lento
 - Memória de 700 **palavras**
 - Problemas de 2 semanas em código de máquina, algumas horas em Speedcoding
- ❖ **Assemblers (compiladores) e Assembly (linguagem)**
 - 1951-1953 Sistemas “compiladores” : A-0, A-1 e A-2
 - Pseudocódigo para linguagem de máquina



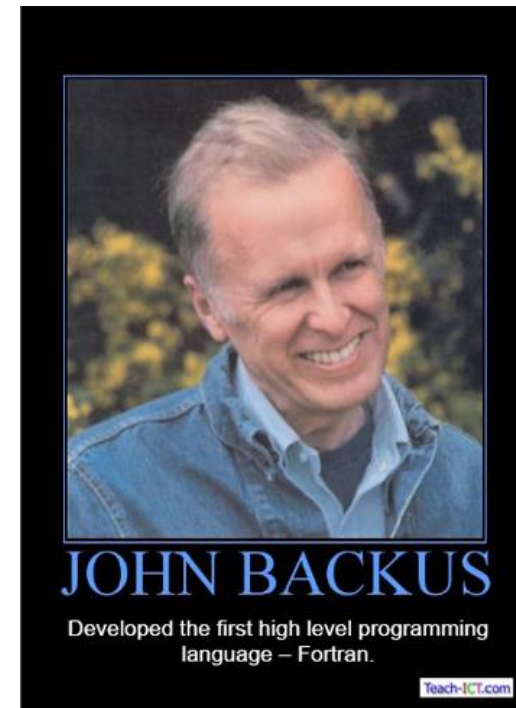
3. Laning e Zierler System - 1953

- ❖ **Implementado no computador MIT Whirlwind**
 - Primeiro sistema compilador “algébrico”
 - Implementado em 1952 e melhorado em 1953
 - Variáveis subscritas (arrays)
 - Chamadas de função
 - Tradução de expressões
- ❖ **Nunca foi implementado para outras máquinas**



4. FORTRAN I - 1957

- ❖ **FORTRAN 0 – 1954 – Não implementado**
 - “The IBM Mathematical **FOR**mula **TRAN**slating System: FORTRAN”
 - John Backus
- ❖ **Projetado para IBM 704**
- ❖ **Ambientes de desenvolvimento do FORTRAN:**
 - Computadores pequenos, lentos e pouco confiáveis
 - Uso para computações científicas
 - Nenhuma metodologia de programação e ferramentas
 - Eficiência da máquina o mais importante: velocidade do código-objeto gerado
- ❖ **Primeira Linguagem de alto nível compilada**



4. FORTRAN

1954-1957 – FORTRAN I

1958 - FORTRAN II

1958 - FORTRAN III (*Nunca publicado*)

1961 - FORTRAN IV

1972 - FORTRAN 66

1980 - FORTRAN 77

1991 - FORTRAN 90

1997 - FORTRAN 95

2003 - FORTRAN 2003

2010 – FORTRAN 2008

2015 – Fortran 2015

2017 – **Fortran 2018** (ISO/IEC JTC 1/SC 22/WG 5/N2146)

2020 – Fortran 2020



4. FORTRAN I

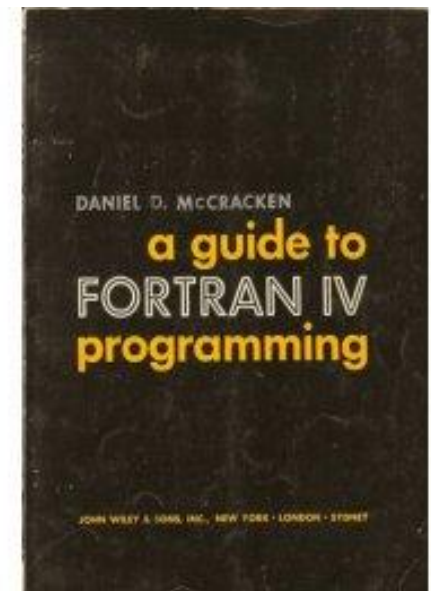
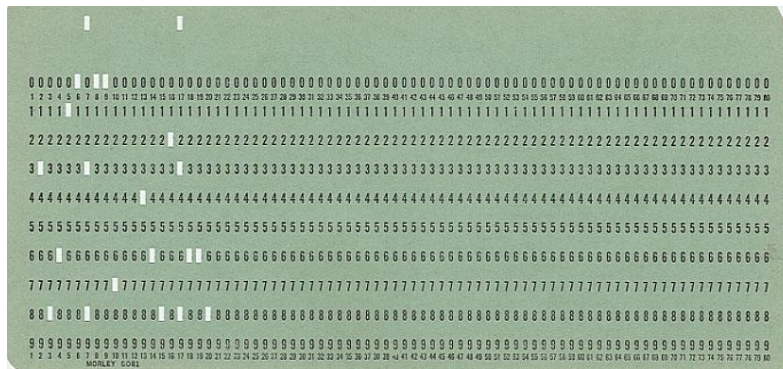
- ❖ Inventado por John Backus, IBM, em 1954
- ❖ Primeira implementação (compilador), abril 1957:
 - Nomes até 6 caracteres
 - Pós-teste do contador de laços (DO)
 - Entrada/saída formatada
 - Sub-programas definidos pelo usuário
 - Seleção em três parâmetros **IF (expr-lógica) n1, n2, n3**
 - Nenhuma declaração sobre tipos de dados
 - Programas de mais de 400 linhas raramente eram compiladas corretamente
 - Código foi muito rápido (execução)
 - Rapidamente foi usado por muitos programadores

5. FORTRAN II - 1958

- ❖ **Distribuído na primavera de 1958**
- ❖ **Compilação independente de sub-rotinas**
- ❖ **Corrigido alguns erros (bugs)**
- ❖ **Programas mais longos podiam ser compilados**

6. FORTRAN IV – 1960-62

- ❖ FORTRAN III desenvolvido, porém jamais distribuído
- ❖ FORTRAN IV: versão padrão até 1978
- ❖ Declarações de tipo explícitas para variáveis
- ❖ Construção lógica IF
- ❖ Capacidade de passar sub-programas como parâmetros a outros programas
- ❖ Padronizado pela ANSI em 1966

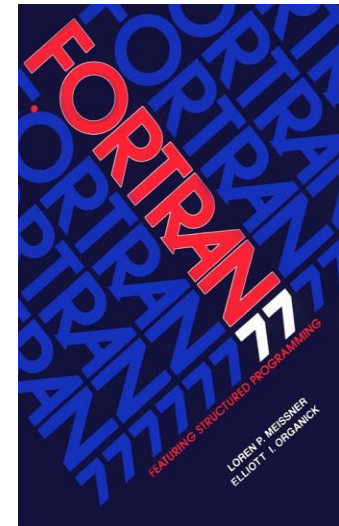
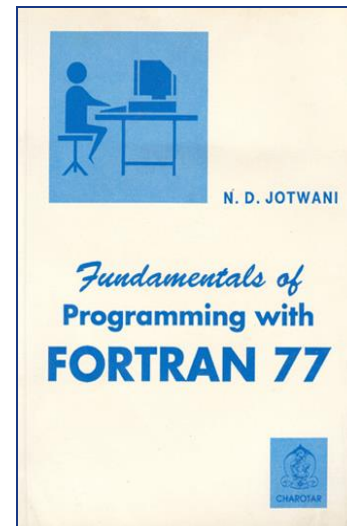


FORTRAN IV



7. FORTRAN 77 - 1978

- ❖ Os mesmos recursos de FORTRAN IV
- ❖ Manipulação de cadeias de caracteres
- ❖ Instruções lógicas de controle de laços
- ❖ Declaração IF com cláusula opcional ELSE
- ❖ Utilizada por supercomputadores



8. FORTRAN 90 - 1990

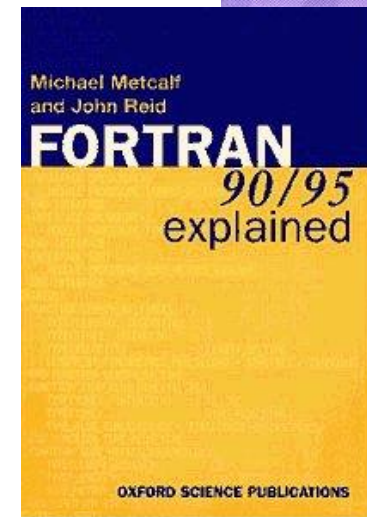
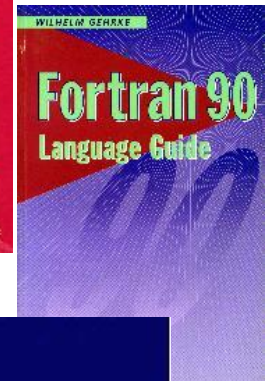
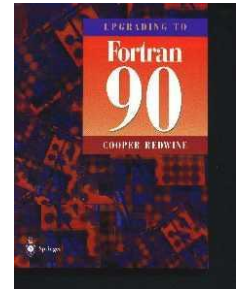
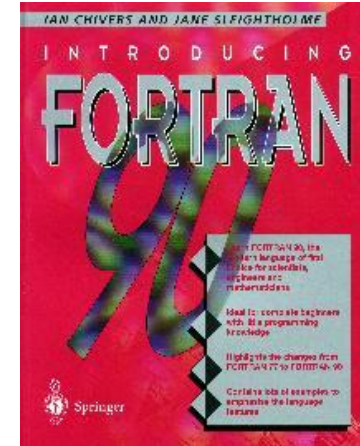
❖ **Drasticamente diferente de FORTRAN 77**

❖ **Novas construções:**

- Módulos
- Arrays dinamicamente alocados
- Ponteiros
- Recursão para sub-programas
- Declaração CASE (seleção múltipla)
- Do-While
- Declarações EXIT, e CYCLE
- Formato livre (independente da coluna)

❖ **Nova versão: FORTRAN 95**

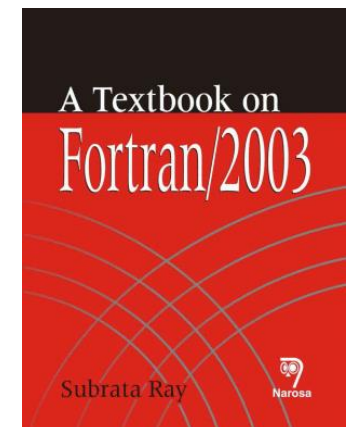
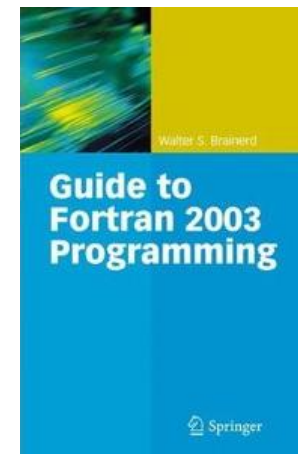
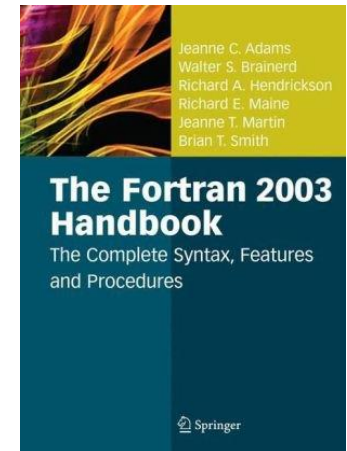
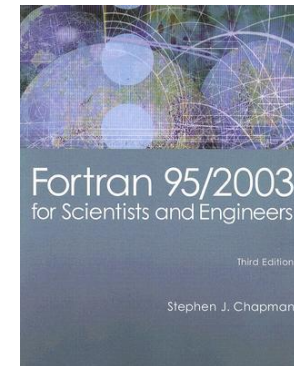
- Fortran 90 melhorado



FORTRAN 2003



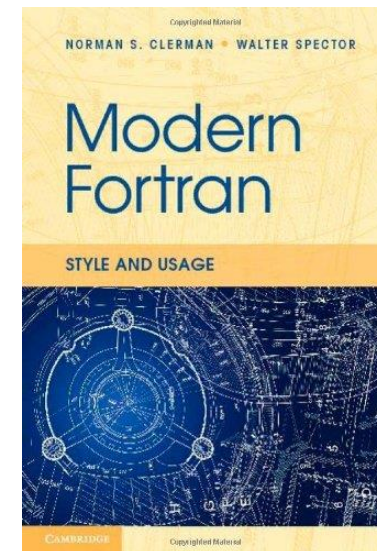
- ❖ **Revisão de Fortran 95**
- ❖ **Final draft: 3 de maio de 2004 (N1601.pdf)**
- ❖ **Características:**
 - Tipos derivados parametrizados
 - Programação Orientada a Objetos
 - Ponteiros para procedimentos
 - Interoperabilidade com C
 - Integração com o Sistema Operacional
 - Argumentos na linha de comandos
 - Variáveis do ambiente
 - Aritmética IEEE (ponto flotante)
 - Melhoria de alocação



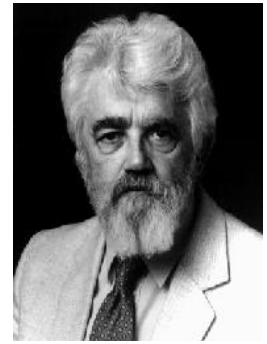
FORTRAN 2008



- ❖ **Final draft: 7 de junho de 2010 (N1830.pdf)**
- ❖ **Características:**
 - Submódulos
 - Coarrays
 - Procedimentos internos passados como argumentos reais
 - Funções matemática com argumentos complexos
 - Funções especiais: trigonométricas inversas
 - Funções de Bessel
 - Função Gamma
 - Distância euclidiana, norma
 - Operações entre bits: shift, máscara, merge, etc.

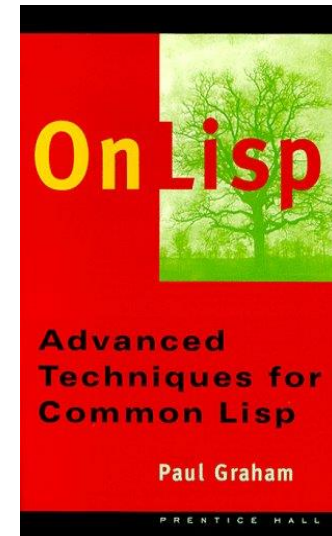


9. LISP - 1959



- ❖ Necessidades de IA: dados simbólicos
- ❖ Incapacidade do FORTRAN
- ❖ ILP I (Information Processing Language I), 1956 para processar listas. Nunca foi implementada
- ❖ Depois IPL II implementado para computador Rand Corporation Johnniac
- ❖ Verão 1958: **John McCarthy**, MIT => IBM, computações simbólicas
- ❖ Outono 1958: McCarthy e Marvin Minsky: MIT AI Project: Desenvolvimento de um sistema de processamento de listas: **LISt Processing Language**

LISP



❖ AI necessitava uma linguagem que:

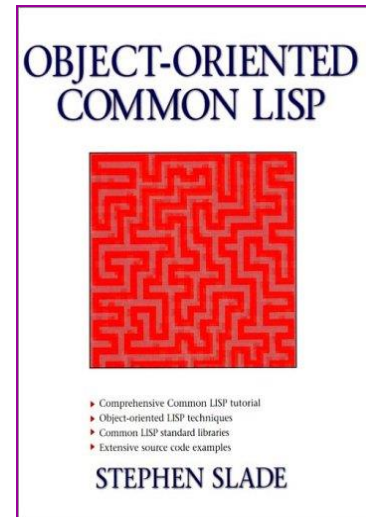
- processe dados em listas (não arrays)
- execute computações simbólicas (não numéricas)

❖ LISP

- Linguagem de programação funcional: funções com argumentos
- Somente dois tipos de dados:
 - Átomos: símbolos (x) ou números (8)
 - Listas (a b c d)
- Sintaxe baseada em lambda calculus

❖ COMMON Lisp (1984) e Scheme (1996)

- Linguagens compiladas: código executável mais rápido



10. ALGOL 58 - 1958

- ❖ **Comitê de estudos ACM e GAMM, 1958, Zurique, projetar uma linguagem universal:**
 - Sintaxe mais próximo da notação matemática
 - Boa para descrever algoritmos
 - Programas mecanicamente traduzíveis para linguagem de máquina
- ❖ **Resultado: IAL (International Algorithmic Language)**
 - 1959: ALGOL (ALGO^{rithmic} L^{anguage})
 - Sucessor do FORTRAN
 - Generalização de recursos
 - Adição de novos conceitos e construções

ALGOL 58

❖ Características

- Conceito de tipo foi formalizado
- Nomes devem ter qualquer tamanho
- Array devem ter qualquer número de subscritos
- Parâmetros foram separados em entrada e saída
- Subscritos foram colocados em colchetes **X[5]**
- Instruções compostas **Begin ... End**
- Separador de instruções: Ponto e Vírgula
- Operador de atribuição :=
- Cláusula ELSE-IF

❖ **ALGOL : como um conjunto de idéias (USA)**

- Novas linguagens: MAD, NELIAC, JOVIAL
- JOVIAL: linguagem científica oficial da Força Aérea USA

11. ALGOL 60 - 1960

- ❖ **Meeting em París, janeiro de 1960**
- ❖ **Novos conceitos**
 - Estrutura de blocos
 - Dois métodos de passar parâmetros
 - Recursão de sub-programas
- ❖ **Sucesso:**
 - Padrão nos próximos 20 anos
 - Base para todas as linguagens imperativas
 - Primeira linguagem independente de máquina
 - Primeira linguagem com sintaxe definida formalmente
- ❖ **Falhas: jamais teve uso generalizado nos EE.UU.**
 - Falta de instruções de Entrada-Saída
 - Fácil de usar, difícil de implementar

13. COBOL - 1960

- ❖ **Background histórico – Linguagens comerciais**
 - FLOW-MATIC, 1957, UNIVAC: linguagem compilada para aplicações comerciais
 - AIMACO, Força Aérea USA
 - COMTRAN (COMercial TRANslation), IBM
- ❖ **Dez. 1959, DoD - Grupo de discussão: Nova linguagem**
 - CBL (Common Business language)
 - Usar o inglês o máximo possível
 - Fácil de usar (menos poderosa)
 - Não influenciada pelos compiladores

COBOL - 1960

❖ Contribuições

- DEFINE: primeira macro em linguagens de alto nível
- Estruturas hierárquicas de dados (registros)
- Nomes até 30 caracteres com hífen
- Declarações de seleção aninhadas
- Data Division

❖ Comentários

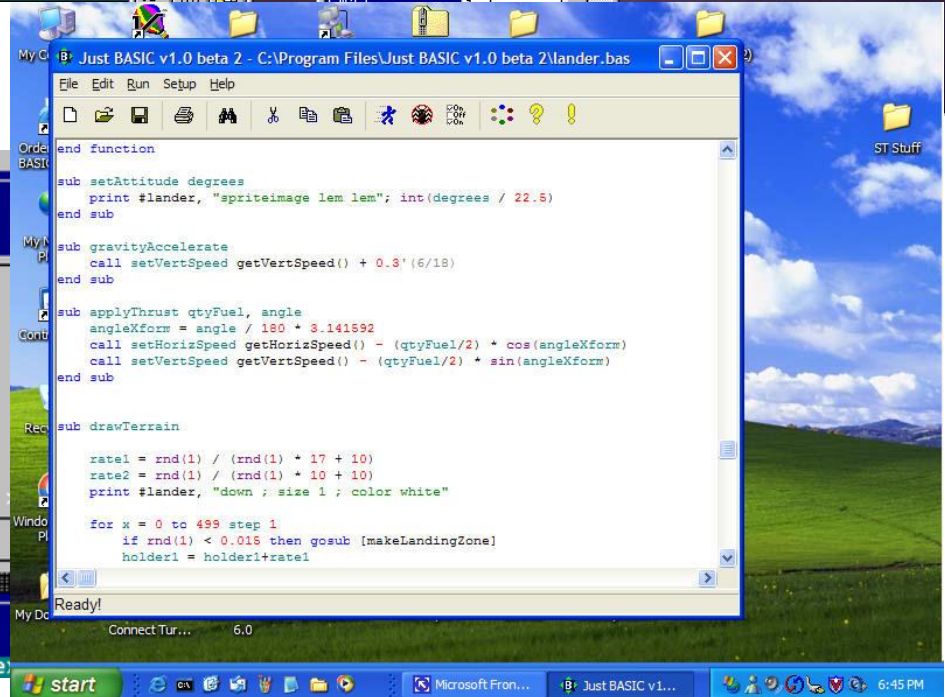
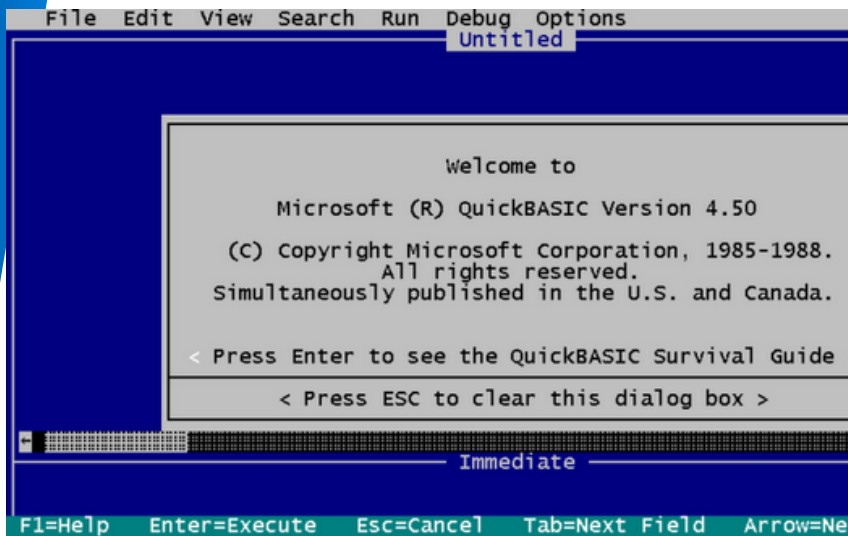
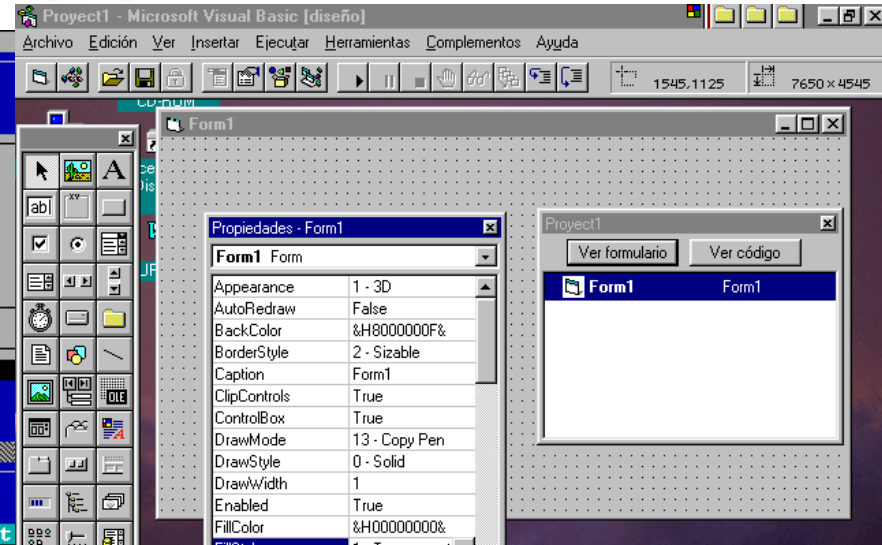
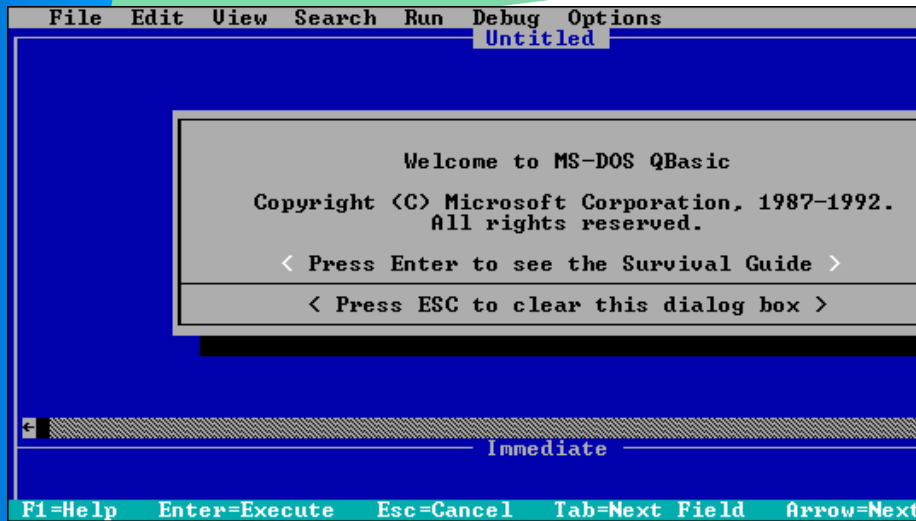
- Sucesso: Uso obrigatório no DoD
- Uso: mecanização eletrônica da contabilidade
- Padronizada pelo ANSI em 1968

14. BASIC - 1964

- ❖ **BASIC = Beginner's All-purpose Symbolic Instruction Code**
- ❖ **Projetado por dois matemáticos: T. Kurtz & J.Kemeny na Darmouth University, em 1963**
- ❖ **Metas do projeto:**
 - Fácil de ser aprendida e usada por estudantes “não-científicos”
 - Agradável e amigável
 - Oferecer rápido retorno para trabalho de casa
 - Permitir acesso livre e privado
 - O mais importante o tempo do usuário do que o tempo do computador
- ❖ **Usado em maio de 1963: acesso remoto a um computador GE225**
- ❖ **Outubro de 1963: 20 terminais no sistema**
- ❖ **ANSI, 1978: padrão Minimal BASIC**
- ❖ **DEC – BASIC PLUS: sistema operacional para PDP-11, RSTS**
- ❖ **Versões contemporâneas: QuickBASIC e MS Visual BASIC**



BASIC



BASIC



Microsoft Visual Basic Version 1.0

Copyright © 1987-1991 Microsoft Corp.
Portions of this product were developed
for Microsoft by Cooper Software, Inc.

This product is licensed to:

Serial number:

Warning: This computer program is protected by
copyright law and international treaties.
Unauthorized reproduction or distribution of this
program, or any portion of it, may result in
severe civil and criminal penalties, and will be
prosecuted to the maximum extent possible
under law.



Microsoft Visual Basic Version 2.0

Copyright © 1987-1992 Microsoft Corp.
Portions of this product were developed
for Microsoft by Cooper Software, Inc.

This product is licensed to:

Serial number:

Warning: This computer program is protected by
copyright law and international treaties.
Unauthorized reproduction or distribution of this
program, or any portion of it, may result in
severe civil and criminal penalties, and will be
prosecuted to the maximum extent possible
under law.



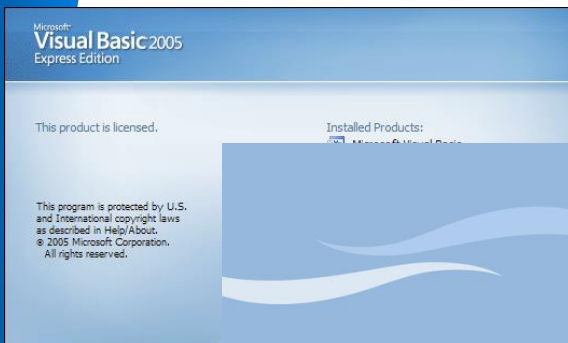
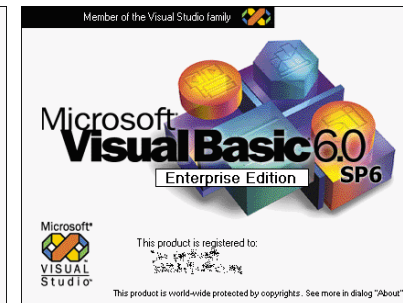
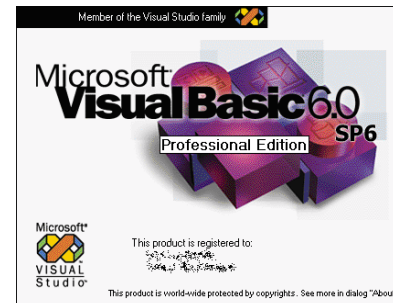
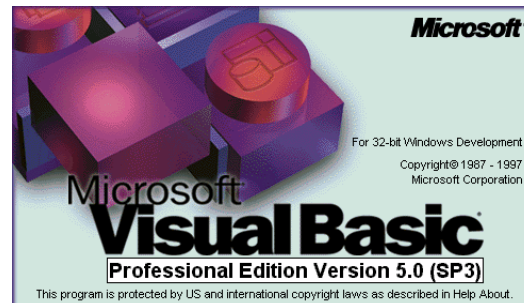
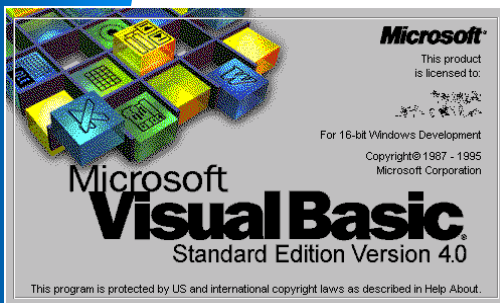
Microsoft Visual Basic Version 3.0

Copyright © 1987-1993 Microsoft Corp.
Portions of this product were developed
for Microsoft by Cooper Software, Inc.

This product is licensed to:

Serial number:

Warning: This computer program is protected by
copyright law and international treaties.
Unauthorized reproduction or distribution of this
program, or any portion of it, may result in
severe civil and criminal penalties, and will be
prosecuted to the maximum extent possible
under law.



15. PL/I

- ❖ **FORTRAN (ciências), LISP (IA), COBOL (negócios)**
- ❖ **PL/I: primeira tentativa de uma linguagem para um amplo espectro de áreas de aplicação**
- ❖ **Na IBM**
 - Dois grupos: computadores de grande e pequena escala
 - Duas linguagens: FORTRAN e COBOL
 - Cientistas (grupo SHARE) IBM 7090 e 1620
 - Grandes arquivos de dados a serem processados
 - Necessidade de I/O sofisticadas e eficientes
 - Aplicações comerciais (grupo GUIDE) IBM 7080 e 1401
 - Uso de análise de regressão para sistemas de informação gerencial
 - Necessidade de um computador universal:
 - IBM System 360

- ❖ **IBM + SHARE: Advanced Language Development Committee, 1963, Comitê 3↔3**
- ❖ **Objetivos:**
 - Nome: FORTARN VI
 - Extensão do FORTRAN IV
- ❖ **Até 1965: NPL=New Programming Language**
 - NPL (National Physical Laboratory) => PL/I
- ❖ **Contribuições**
 - Programas criassem tarefas executadas concorrentemente
 - Detectar e manipular 23 diferentes tipos de exceções
 - Procedimentos usados recursivamente e não-recursivamente
 - Ponteiros como tipos de dados
 - Seções transversais de array referenciados como vetores

16. Linguagens Dinâmicas

- ❖ **Caracterizadas pela tipificação dinâmica e alocação de armazenagem dinâmica**
- ❖ **Não tiveram influencia de outra linguagem nem influenciaram para novas linguagens**
- ❖ **APL (A Programming Language), 1962, IBM**
 - Projetada como linguagem de descrição de hardware
 - Grande número de operadores
 - Programas são difíceis de ser lidos
- ❖ **SNOBOL, 1964, Bell Laboratories**
 - Projetada para processamento de texto (strings)
 - Usada para escrever editores de texto

17. SIMULA 67 - 1967

- ❖ **Norwegian Computing Center, K. Nygaard & O-J. Dahl**
- ❖ **SIMULA I : 1962-64**
- ❖ **Projetada para simulação de sistemas e implementada em UNIVAC 1107**
- ❖ **Contribuição:**
 - Co-rotinas (sub-programas)
 - Implementada na base de estruturas: classes
 - Classe: estrutura de dados + funcionalidade
 - Idéias básicas para Abstração de Dados

18. ALGOL 68

- ❖ **Sucessora de ALGOL 60**
- ❖ **Projeto baseado no conceito de ortogonalidade**
 - Conjunto relativamente pequeno de construções primitivas pode ser combinado em um número relativamente pequeno de maneiras para construir estruturas de controle e de dados
- ❖ **Contribuições**
 - Estruturas de dados definidas pelo usuário
 - Arrays dinâmicos (flex): não especifica limites de subscrito
 - Uso de uma meta-linguagem elegante e concisa (?)
 - Forte influência sobre Pascal, C e ADA
 - Destinada a aplicações científicas

19. PASCAL - 1971

- ❖ N. Wirth & C.A.R. Hoare, Grupo 2.1 IFIP, proposta para modificar ALGOL 60
- ❖ ALGOL-W, 1966, Stanford University
- ❖ Projetado para o ensino universitário de programação estruturada
- ❖ Popularidade baseada no conceitos de simplicidade e expressividade (operadores muito poderosos)
- ❖ Problema: impossibilidade de descrever um sub-programa que tome como parâmetro um array de tamanho variável
- ❖ Ainda é uma linguagem relativamente segura
- ❖ Base do Delphi (Borland)



N. Wirth



C.A.R. Hoare

20. C - 1972

❖ Precursoras

- **CPL**: Univ. Cambridge
- **BCPL**: linguagem de sistemas, M. Richards, 1969
- **B**: primeira linguagem de alto nível implementada em UNIX, 1970, sem verificação de tipos

❖ **C: projetada e implementada por Dennis Ritchie, Bell Laboratories, 1972, baseada na linguagem B**

❖ **Nome Original: NB (New B?)**

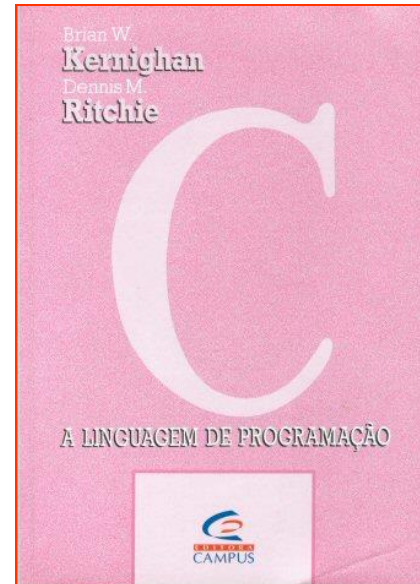
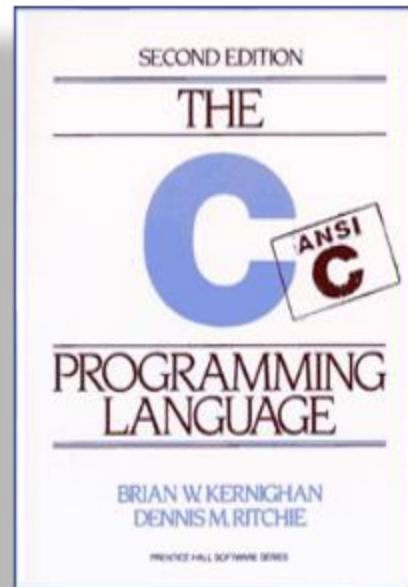
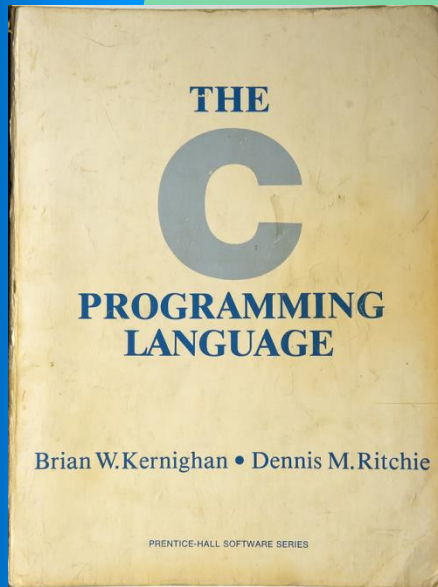
❖ **Padronizado**

- Único: o livro de Kernighan e Ritchie, 1978
- C (ANSI), 1989

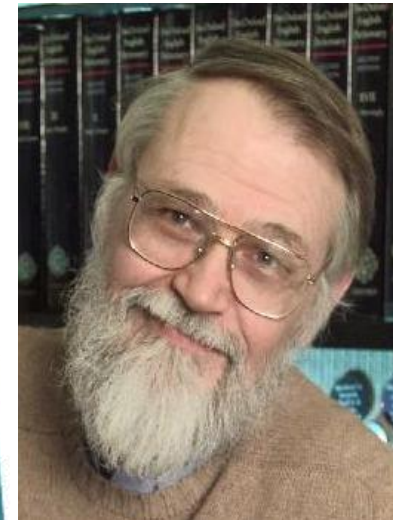
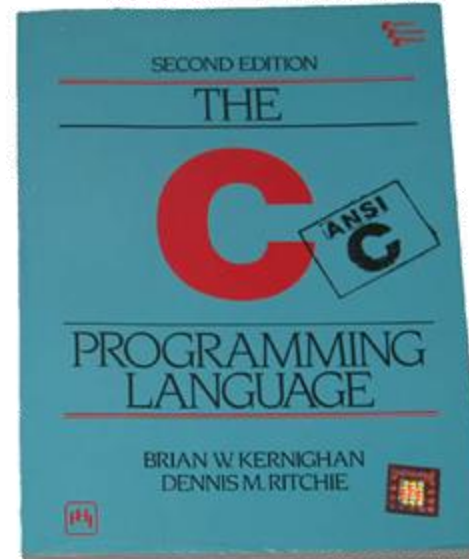
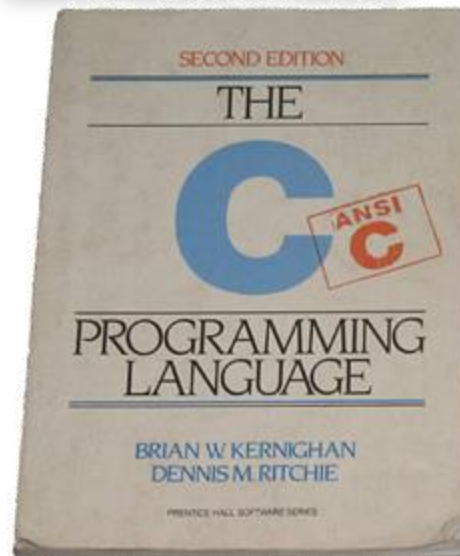
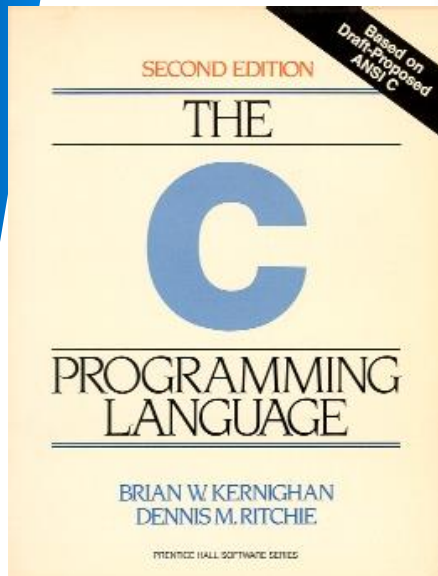
❖ **C é muito apreciada e detestada:**

- Completa ausência de verificação de tipos
- Faz parte do sistema UNIX: compilador gratuito
- Muito flexível: alto e baixo nível de programação

20. C - 1972



Dennis Ritchie
1941-2011



Brian W. Kernighan

20. C - 1972



21. Descendentes de ALGOL

- ❖ **MODULA 2, N. Wirdth, 1976: concorrência**
 - Projetada para o computador Lilith
 - Pascal + módulos
- ❖ **MODULA 3, SRC of DEC & Olivetti, 1989, L. Cardelli**
 - Baseada nas linguagens Modula-2, Mesa, Cedar e Modula-2+
 - Adiciona: classes, manipulação de exceções, coleta de lixo e concorrência
- ❖ **OBERON, N. Wirth,**
 - Modula-2 + OO
 - Menos complexa que C++ e Ada 95
- ❖ **Delphi, Borland**
 - Pascal + OO
 - Interface Gráfica

22. PROLOG - 1972

❖ **Início da década do 70: Projeto Prolog**

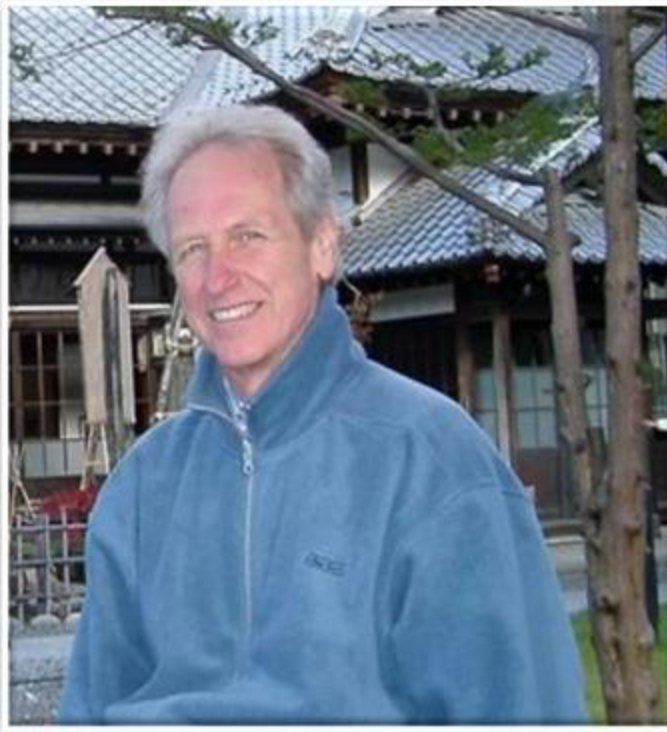
- Colmerauer & Roussel, Artificial Intelligence Group, Univ. Aix-Marseille
- R. Kowalski, Dpto of Artificial Intelligence, Univ. Edinburgo

❖ **Primeiro compilador, 1972, Marseille**

❖ **Características**

- Baseado na lógica formal (cálculo de predicados + resolução)
- Não procedural
- É usado na forma de um banco de dados inteligente com dois tipos de instruções: fatos e regras
- Eficiente somente para algumas pequenas áreas de aplicação (sistemas de gerenciamento de BD, IA)

22. PROLOG - 1972



Kowalski



Alain Colmerauer



P. Roussel

23. ADA - 1983

Wooden=palha
Tin=lata
Iron=ferro
Steel=aço
Stone=pedra

❖ O maior esforço de projeto da História

- DoD: 1974: sistemas embarcados (hardware)
- Mais de 450 linguagens de programação em uso no DoD
- 1975, High-Order language Group, Exército, Marina e FA
 - Identificar requisitos de uma nova linguagem
 - Avaliar as linguagens existentes
 - Recomendar adoção ou implementação de um conjunto mínimo de LP
- Resultado: Projetos Woodenman (1975) e Tinman (1976)
- Participantes: + 200 pessoas e + 40 organizações
- 1977, Projeto Ironman (contrato de desenvolvimento) => Ada
- 1978, Projeto Steelman (doc. Requisitos)
 - 1979, nome Ada: Augusta Ada Byron (1815), Condessa de Lovelace, matemática e primeira programadora do mundo. Trabalhou com Babbage
- 1980, Projeto Stoneman (Especificação de requisitos)
- 1980, MIL-STD 1815: Ada Language Reference Manual
- 1983, padronizado pela ANSI

❖ Contribuições

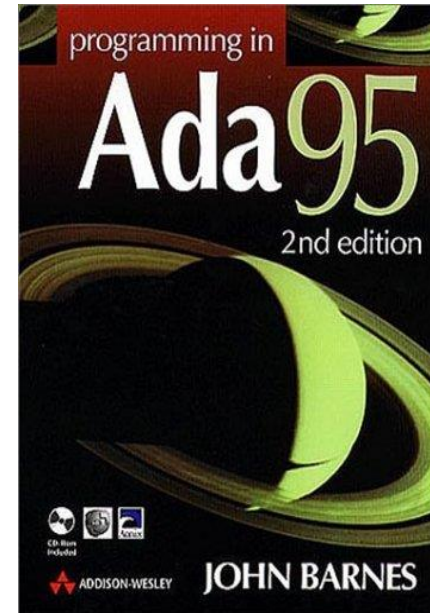
- Pacotes: suporte para abstração de dados
- Manipulação de exceções e erros
- Unidades de Programa Genéricas
- Concorrência baseada no modelo de tarefas

❖ Comentários

- Projeto competitivo
- Inclui tudo sobre engenharia de software e projeto de linguagens
- Primeiros compiladores: cinco anos depois

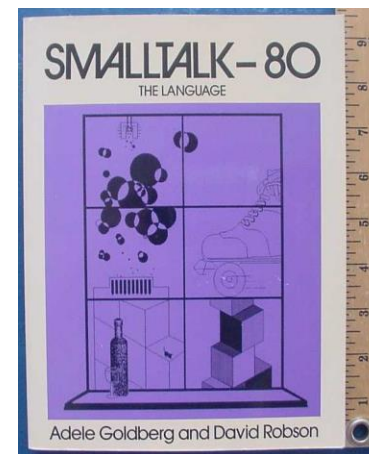
❖ Ada 95, 1988 – Projeto Ada 9X

- Suporte para OO
- Melhores mecanismos de controle para dados compartilhados
- Bibliotecas mais flexíveis
- Capacidade de interface



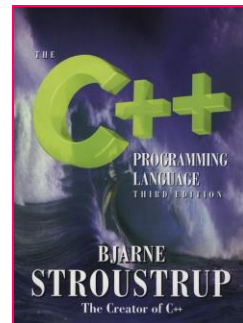
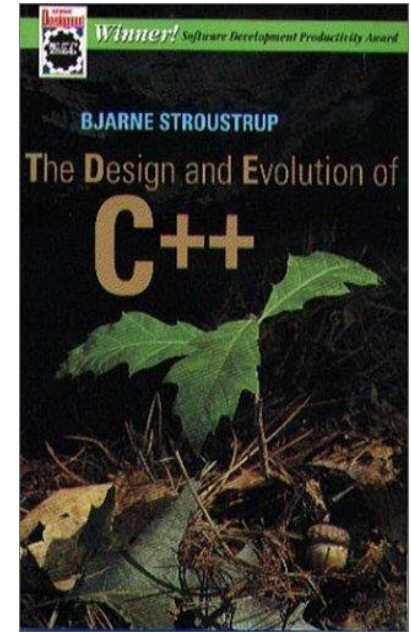
24. SMALLTALK – 1972-80

- ❖ **Baseado no trabalho de dissertação Ph.D. de Alan Kay, Univ. Utah, 1969**
 - Projeto original de Kay: Dinabook = processador de informações gerais. Baseado na linguagem Flex
 - Xerox PARC: Learning Research Group
 - Resultado: hardware Xerox Alto e software Smalltalk-72
 - Desenvolvimento final: Adele Goldberg
- ❖ **Primeira implementação completa: Smalltalk-80**
 - Abstração de dados (objetos)
 - Herança
 - Ligação dinâmica de tipos
 - Sistema de Janelas Gráficas



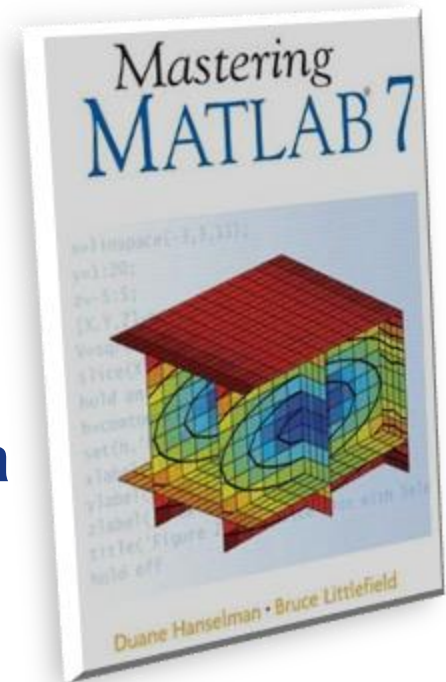
25. C++ - 1985

- ❖ **B.Stroustrup, 1980, Bell Labs**
- ❖ **Combina recursos imperativos e OO**
- ❖ **Desenvolvido a partir de C e SIMULA 67**
 - Facilidades OO
 - Manipulação de exceções
 - Tipos parametrizados
 - Linguagem grande e complexa
- ❖ **1985, primeira implementação: sistema Cfront**
 - Traduz código C++ em C
- ❖ **Release 3.0 em 1990**
- ❖ **Padrão ANSI: 1997**
- ❖ **Linguagem popular: compiladores bons e baratos**



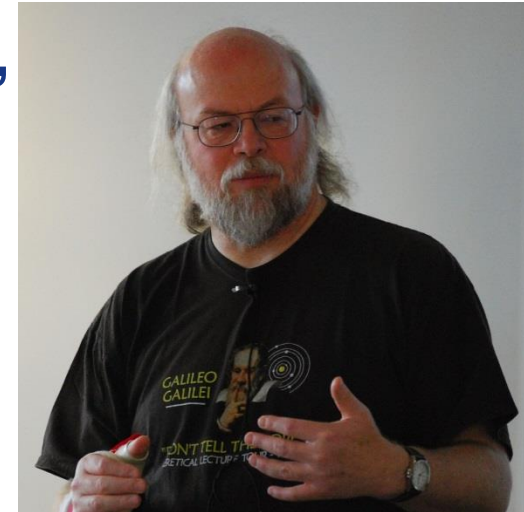
26. MATLAB

- ❖ Desenvolvido na University New Mexico e Stanford University por Cleve Moler, como um sistema de computação matricial: **MAT**ricial **LAB**oratory
- ❖ Baseado no FORTRAN e no Pascal
- ❖ Ambientes de programação
 - Linguagem matricial interpretada e compilada
 - Editor dirigido por sintaxe
 - Janelas Gráficas Interativas
 - GUI - Graphic User Interface
 - Utiliza toolbox's (bibliotecas especializadas)
- ❖ Linguagem para Ciências e Engenharia



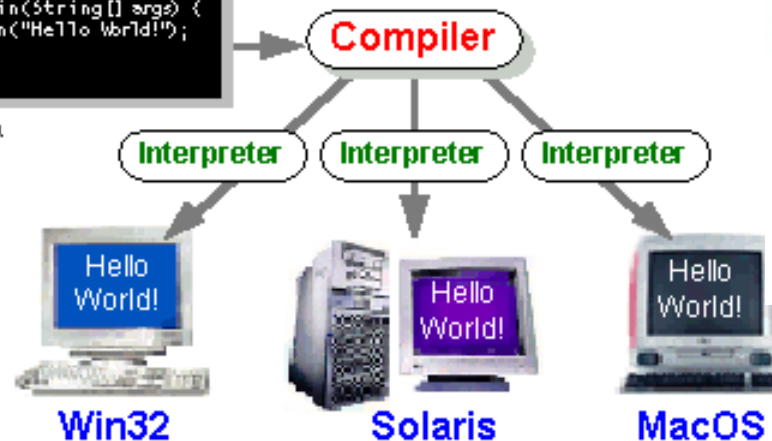
27. JAVA - 1995

- ❖ Desenvolvida na SUN Microsystems,
- ❖ 1990, James Gosling
- ❖ Baseada em C++
 - Altamente simplificada
 - Suporte somente a OO
 - Têm referências, não ponteiros
 - Suporte por applets (programas no cliente) e concorrência



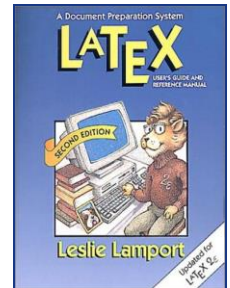
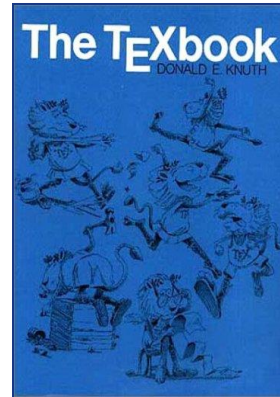
Java Program

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



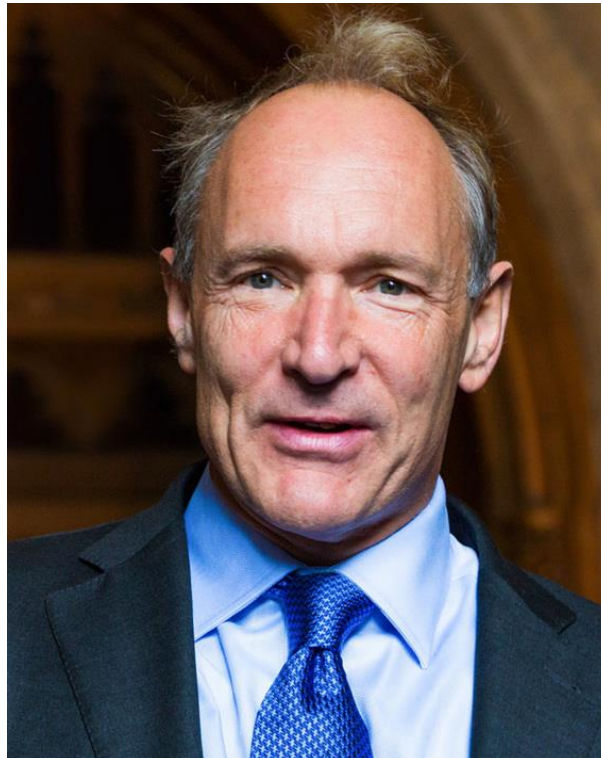
28. LaTeX

- ❖ Desenvolvido por D. Knuth, 1977: sistema TeX
- ❖ 1980, L. Lamport: sistema LaTeX
- ❖ 1989, TeX Users Group Meeting, Stanford
- ❖ Características
 - Linguagem baseada no sistema de marcas genéricas
 - Linguagem de Processamento de texto automático
 - Arquivo fonte totalmente portátil (hardware, software): texto ASCII
 - Programação em alto e baixo nível
 - Composição incremental (baseado em pacotes)
 - Direcionado para documentos complexos, fórmulas matemáticas, e Engenharia



29. HTML

- ❖ Linguagem de marcas, utilizado na web
- ❖ Baseado no conceito de HiperTexto
- ❖ Interface: Visualizador (browser)
- ❖ Timothy John Berners-Lee, 1991



30. PHP, Perl, ...

❖ **PHP (Personal Home Page)**

- Rasmus Lerdorf (Grupo Apache), 1994

❖ **Phyton**

- Guido van Rossum, Holanda, 1990
- Orientada a objetos

❖ **Perl**

- Larry Wall,
- sh+awk

❖ **Ruby**

- Yukihiro Matsumoto, Japão, 1996
- Orientada a objetos

❖ **Javascript (LiveScript)**

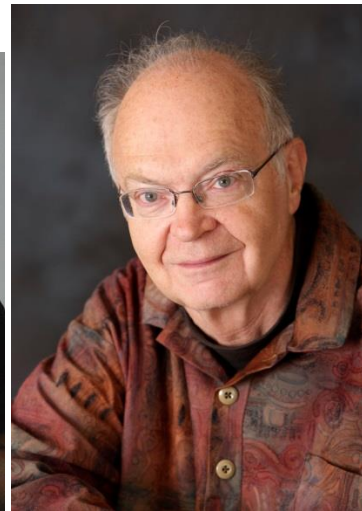
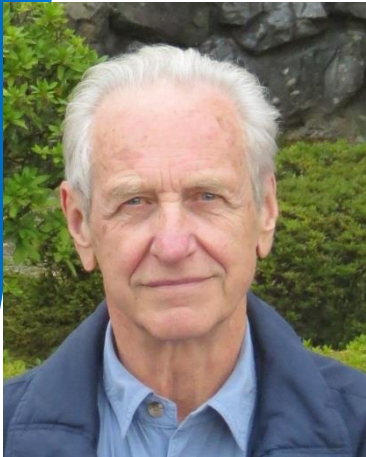
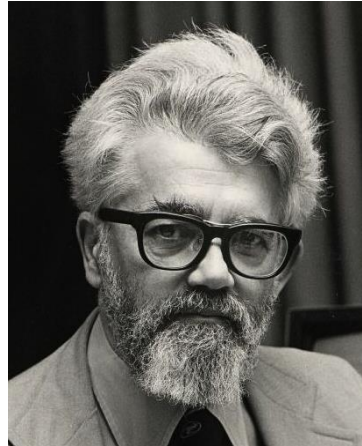
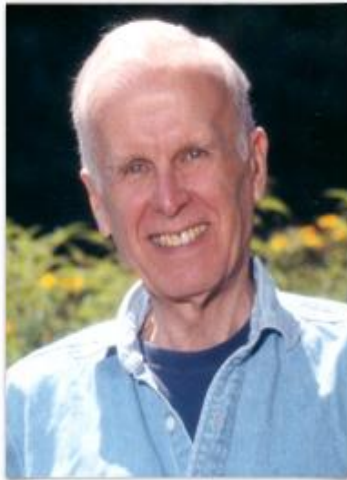
- Netscape (Sun), 1995
- Uso com HTML



1. **Plankalkul**
2. **FORTRAN**
3. **LISP**
4. **Pascal**
5. **C**
6. **Prolog**
7. **Smalltalk**
8. **Java**
9. **La-TeX**
10. **HTML**

1. **Adele Golberg, 1980**
2. **John McCarthy, 1958**
3. **Donald Knuth, 1977**
4. **Konrad Zuse, 1945**
5. **Niklaus Wirth, 1971**
6. **Timothy John Berners-Lee, 1991**
7. **James Gosling, 1995**
8. **Dennis Ritchie, 1972**
9. **Robert Kowalsky, 1972**
10. **John Backus, 1957**

10++



Programming Language Explorations

Ray Toal • Rachel Rivera
Alexander Schneider • Eileen Choe



 **CRC Press**
Taylor & Francis Group
A CHAPMAN & HALL BOOK

September 22, 2016

Referências na Internet

- ❑ <http://www.levenez.com/lang/>
- ❑ http://oreilly.com/news/languageposter_0504.html
- ❑ http://vernix.org/marcel/programming_language_authors/
- ❑ http://www.kcl.ac.uk/kis/support/cit/fortran/john_reid_new_2003.pdf
- ❑ [http://www.everything2.com/index.pl?node_id=858421.](http://www.everything2.com/index.pl?node_id=858421)
- ❑ <http://merd.sourceforge.net/pixel/language-study/diagram.html>
- ❑ http://oreilly.com/news/languageposter_0504.html
- ❑ <http://www.webdesigndev.com/programming/30-most-influential-people-in-programming>
- ❑ <http://www.flazx.us/>

Perguntas para a prova

- ❖ Qual foi a primeira linguagem de programação?
- ❖ Quais são as 10 pessoas mais influentes em linguagens de programação?
- ❖ Qual foi o maior esforço de projeto de uma linguagem de programação na história?
- ❖ Qual foi a linguagem orientada para estudantes?
- ❖ Qual foi a linguagem projetada para um amplo espectro de aplicações?
- ❖ Qual foi a primeira linguagem de programação total e completamente orientada a objetos?
- ❖ Qual é o significado original de PHP?



Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

ascv@uenf.br
ausberto.castro@gmail.com

