

PASCAL

É uma linguagem de programação estruturada que recebeu este nome em homenagem ao matemático Blaise Pascal. Foi criada em 1970 pelo suíço Niklaus Wirth, tendo em mente encorajar o uso de código estruturado.

O próprio Niklaus Wirth diz que Pascal foi criada simultaneamente para ensinar programação estruturada e para ser utilizada em sua fábrica de software. Simultaneamente, a linguagem reflete a liberação pessoal de Wirth das restrições impostas pelo após seu envolvimento com a especificação de ALGOL 68, e sua sugestão para essa especificação, o ALGOL W.

A linguagem é extremamente bem estruturada e muito adequada para ensino de linguagens de programação. É provavelmente uma das linguagens mais *bem resolvidas* entre as linguagens estruturadas, e certamente um dos exemplos de como uma linguagem especificada por uma pessoa pode ser bem melhor do que uma linguagem especificada por um comitê.

Pascal é normalmente uma das linguagens de escolha para ensinar programação, junto com Scheme, C e Fortran.

Comercialmente, a linguagem foi sucedida pela criação da linguagem Object Pascal, atualmente utilizada nas IDEs Borland Delphi, Kylix e Lazarus. Academicamente, seus sucessores são as linguagens subsequentes de Niklaus Wirth: Modula-2 e Oberon

A partir da versão 2005, o Delphi passou a se referir a sua linguagem de programação como Delphi Language.

Assim como a Linguagem C, que é padronizado pela ANSI (Ansi C), o Pascal possui padrões pela ISO, como o Pascal Standard e o Advanced Pascal.

O primeiro compilador Pascal foi desenvolvido em Zurique para a família de computadores CDC 6000, sendo lançado em 1970. Também em 1970 foi desenvolvido o primeiro compilador Pascal norte americano, na Universidade de Illinois por Donald B. Gillies, que gerava código de máquina nativo para o mini-computador PDP-11.

Pensando-se em propagar rapidamente o uso da linguagem, foi criado, em Zurique, um "kit de conversão" que incluía um compilador que gerava código intermediário, e um simulador para ele. Esse kit foi batizado de p-System, e foi utilizado, entre outras coisas, para criar um sistema operacional para mini-computadores chamado UCSD p-System, desenvolvido pelo Instituto de Sistemas de Informação da Universidade da Califórnia em San Diego. Segundo o próprio Niklaus Wirth, o p-System e o UCSD foram instrumentais na popularização do Pascal.

Nos anos 80, Anders Hejlsberg desenvolveu o compilador Blue Label Pascal o Nascom-2. Depois, ele foi trabalhar na Borland e reescreveu seu compilador transformando-o no Turbo Pascal para a plataforma IBM PC (e também CP/M 80), que era vendido a US\$ 49,95, muito mais barato do que o Blue Label. Uma característica muito importante é que o Turbo Pascal é uma linguagem compilada, que gera código de máquina real para a arquitetura Intel 8088, tornando-a muito mais rápida do que as linguagens interpretadas.

Por ser mais barato, o Turbo Pascal passou a ter uma grande influência na comunidade Pascal, que começou a se concentrar na plataforma IBM PC no fim dos anos 80. Muitos usuários de PC da época migraram para o Turbo Pascal, em busca de uma linguagem estruturada que não fosse interpretada, para substituir, por exemplo, o BASIC. Pode se afirmar que o sucesso comercial de Turbo Pascal foi definitivo para a ampla divulgação da linguagem Pascal entre os usuários de micro-computador.

A linguagem **Pascal** foi criada para incentivar a programação modular e estruturada, facilitando a criação de procedimentos com baixo acoplamento e alta coesão. Um programa em **Pascal** é composto de constantes e variáveis globais, procedimentos e funções re-entrantes e um programa principal.

Procedimentos não retornam valores, funções sim. Tanto em procedimentos quanto em funções os parâmetros podem ser passados por nome ou por valor. É possível passar vetores e matrizes com o tamanho, mas não a quantidade de dimensões, especificado no tempo de execução.

Procedimentos e funções podem conter, dentro de seu escopo, novos procedimentos e funções. Dentro de qualquer parte do programa também podem ser criados blocos com os comandos **BEGIN** e **END**, que também possuem seu próprio escopo. Nas versões originais, as variáveis só podiam ser declaradas em posições específicas e não ao decorrer do programa, o que limitava a regra de escopo.

O conjunto de procedimentos e funções pré-definidos é fixo e inclui as funções read, readln, write e writeln, para realizar E/S.

Referências

- Wikipedia, disponível em <http://pt.wikipedia.org> acessado em 22 de Fevereiro de 2008