

计算机常用的几种语言：PASCAL语言 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/261/2021\\_2022\\_\\_E8\\_AE\\_A1\\_E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E5\\_c67\\_261348.htm](https://www.100test.com/kao_ti2020/261/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E5_c67_261348.htm) PASCAL The computer language PASCAL was invented in 1970 by Professor Nucleus Wirth of Zurich. It was named after the French mathematician, Blaze Pascal , who invented one of the earliest known calculators--a forerunner of the modern computer. As with all computer languages PASCAL was designed to meet a particular objective. In this case the objective was to relate a language, which would be better suited to teaching programming than any existing language. Although PASCAL was invented in 1970, this is not to s 叮 that it came out of the blue and had no similarity to computer languages that already existed PASCALs ancestors include the programming languages ALGOL and PL I . PASCAL took much of the best material from these two languages and as a result it is a better and simple language than either . PASCAL, like BASIC, also is a high level language. However, before a program written in PASCAL can be run on a computer it must be trans-asted into machine code. This is different from BASIC . It means that running a PASCAL program takes at least three operations : (1) The program must be entered into the computer as text. (2) It must be compiled into machine code (3) It must then be run,i.e.the computer must begin obeying its instructions. The first stage usually involves the use of a pro-gram called an editor, which allows text of any sort to be entered, altered and saved for later use. The sec-and stage then uses

the output from the editor and converts it to machine code. Finally this machine code is loaded into the computer and is allowed to take control. We can visualize this process as :

program written on paper    Editor    program text in computer    Compiler

program in machine code form    program loaded and running

We start with a program written on paper and end with the equivalent machine code program in control of the computer's actions . In developing a program it is rare for it to work first time . In this case we have to discover the reason for the failure and correct it. This involves going back to the text of the program and using the editor to make the correction. The amended program must then be re-compiled and once again tested by running. This forms a cycle that may have to be repeated several times . If you have used a computer with a language such as BASIC you may be puzzled as to why PASCAL requires so many steps to get from a written program to running program. It is fairly common to find implementations of BASIC where the program is typed in, edited and run without any obvious moving from editor to compiler, catch. The reason for this is that an interpreter most often implements BASIC. An interpreter is a program that serves the same purpose as a compiler, in that it allows instructions written in a high level language to control a computer that understands only machine code, but it achieves this in a very different way . An interpreter is a program that 's in the machine at the same time as the text of your program. When you type RUN , or whatever word indicates that the computer should carry out your program, the interpreter looks at the first line of your program and

makes the machine carve out actions that are implied by its meaning. You can think of this as the interpreter converting the current line of your program to machine code and running it whereas compiler converts the entire program before you can run it. It is obvious that for learning a language it is better to use an interpreter or some similar system. Its a sad fact that nearly all implementations of PASCAL use compilers and therefore not ideal foreigners

. However, if you need a program to bedewed at full speed then use of a compiler is clearly-.he better choice. This is because an interpreter trans- . Ate the program every time it is run but a compiler4ocs it once only. Thus, for a program that uses an-interpreter, the translation time is added to the time Nat the program takes to carry out its task. Therefore, hale interpreters are easy to use, compilers produce sat programs . The picture that we have given so far is a little too simple to account for the way that all versions of PASCAL are implemented. The truth is that a complier is a program that translates from one computer language to another--not just to machine code . It would for example be quite possible to write a compiler to change PASCAL into BASIC

. Indeed it must be an advantage because if we translated PASCAL into BASIC we could make use of the BASIC interpreter that is almost always supplied with a computer. What we have just proposed is a method that issued by many versions of PASCAL although instead of translating to BASIC the language that the compiler translates to is known as Plods. In other words , many PASCAL compilers do not translate PASCAL into machine code:

Instead they produce another language called P Code. A P-Code interpreter then interprets this P-Code program. This might sound like a silly way to run a program because it combines the difficulty of a compiler in the first stage with the slowness of an interpreter in the second, However, it is the method used by a number of PASCAL implementations and the reason is that it makes it easier to produce versions for different machines . If a compiler produces machine code then moving it to a new machine means virtually rewriting it. If a compiler produces P-Code then moving it to a new machine is a matter of writing a new P-Code interpreter--which is a considerably easier task! To summarize, there are two popular ways to implement PASCAL : Compile to machine code, or compile to P-Code and then interpret it. Of course any one version of PASCAL will only use one of these two methods but it should help you to understand the different steps that you have to go through to run your PASCAL programs and why other versions are different. 翻译:PASCAL语言 PASCAL这种计算机语言是在1970年由苏黎士的NiklausWirth教授提出的。它是以世界上最早发明计算器（现代计算机的前身）的法国数学家Blaise Pascal的名字而命名的。与所有的计算机语言一样，PASCAL也是为满足特定的目标而设计的一种语言。这个目标就是创造出一种语言，使之比其他任何一种已有的计算机语言更适于编程教学。虽然PASCAL语言早在1970年就被提出来了，但并不是说那时它已走出蓝图阶段，与其他已有的计算机语言没有任何相似之处。实际上，PASCAL语言的前身包括了ALGOL和PLI编程语言。PASCAL语言汲取了这两种语言的精华，从而成为比两者

一中任何一种都更好、更简单的语言。PASCAL与Basic一样，都属高级程序设计语言。不过，用PASCAL语言编写的程序必须先转换成机器代码才能在计算机执行。这一点与Basic不一样。这意味着运行一个PASCAL程序至少要经过三个步骤：(1)将程序以文本的形式输入计算机。(2)将程序编译成机器代码。(3)运行该程序。即计算机开始按指令行事。在进行第一步操作时，要用到编辑程序，该程序允许对任何一科文本文件进行输入、修改和查询以供日后使用。第二步则把从编辑程序处取得的输出转换成机器代码。最后，将机器代码调进计算机并运行。这个过程可视为：

程序	编辑程序	计算机中的程序文本	编译程序	机器代码形式的程序	程序的调用和运行
----	------	-----------	------	-----------	----------

这样，我们就从写在纸上的程序开始，到计算机操作控制下对等的机器代码结束。开发程序很少能一次就运行成功。这时，我们必须找出其失败的原因并修正之。修改后程序必须重新编译之后再运行。这样反复循环，可能要经很多次才能最后交功。如果你曾经采用过Basic语言编程，也许会对PASCAL从写程序到程序运行何以需要经过这么多的步骤感到困惑不解。一般地，在BASIC中，程序从键入、编辑到运行都没有明显的从编辑程序到编译程序的转换等。原因在于BASIC系统大多带有解释程序。解释程序的功能与编译程序相同，即使高级语言编写的指令为只能识别机器代码的计算机所接受，但达到这一目标的途径却不一样。解释程序既存在于机器中，同时又是你所编写的程序文本。当你敲运行键或其他命令指示计算机执行程序时，解释程序将描述你所编程序的第一行并使计算机按该行程序暗示的意思行动。可以这样来理解：解释程序

是将程序的当前行转换为机器代码后立即执行该行指令，而编译程序则是将整个程序转换后再运行。显然在学习一门语言时，最好选用解释程序或其他类似的系统。可惜的是，几乎所有的PASCAL都是采用编译程序运行方式，这对初学者来说不是很理想。但如果你需要程序全速运行，显然，编译程序是较好的选择。这是因为程序每运行一次，解释程序都要解释一次；而编译程序只需编译一次后，程序即可多次运行。因此，采用解释语言编程的程序，解释的时间增加了程序执行的时间。简而言之，解释程序使用起来方便，而编译程序则可使程序运行速度加快。以上我们所描述的只PASCAL中很少的、最简单的部分，实际应用中PASCAL有很多复杂的分支，在此不能一尽述。事实上，编译程序是指将一种计算机语言转换成另一种计算机语言，而不仅仅是转换成机器代码。比如，可以写一个编译程序，实现从PASCAL语言到Basic语言的转换。这样有利于将所有计算机带有的Basic解释系统尽量利用起来。刚才我们所建议的方法已被许多PASCAL本所利用，但编译程序转换成的，是一种被称为P代码的语言，而不是BASIC语言。换句话说，大多PASCAL编译程序不是把PASCAL转换成机器代码；而是转换为另一种名叫P计算机语言。P程序再经P解释程序解释成能接受的机器代码即可。用这种方法运行程序看起来似乎很愚蠢，因为这样做既带有第一步编译程序使用上的难度，又带有第二步解释程序执行速度慢低缺点。尽管如此，然而在PASCAL运行中，这种方法仍用得相当多：这主要是因为便于针对不同的机器采用不同的版本的缘故。如果利用译程序生成的是机器代码，那病将编译程序移植到新机器中就意翎着最后重写

程序。如果编译程序生成的是P代码，那么将其移植到目标机器内实质上就是编写出一种新的代码解释程序来——这样做要容易得多！总之，有两种常用的方式运用PASCAL程序：一种是将其编译成机器代码，另一种是编译成P代码再解释之。当然，任何一种PASCAL系统都只会采用这两种运行方式中的一种。介绍这两种方法的目的是帮助用户理解：在运行PASCAL程序时将会经历哪些步骤，以及为什么其他文本的运行步骤有所不同。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)