

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PASCAL

Treball Dirigit de Competències Transversals

Hash TD: 1455

23 de Desembre de 2021

Llenguatges de Programació



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índex

Introducció	1
Propòsit del llenguatge	2
Paradigma de programació	2
Programació Imperativa, Procedural i Estructurada	2
Programació Orientada a Objectes	3
Sistema d'execució	4
Sistema de tipus	5
Principals aplicacions	6
Història del LP	6
Exemples de codi	7
Característiques particulars	9
Conclusió i visió personal	9
Bibliografia	10
Descripció de les fonts d'informació	11
Avaluació de la qualitat de la informació trobada	11

1. Introducció

Aquest document tractarem d'una banda els elements i característiques que conformen el llenguatge de programació Pascal, la seva història i alguns exemples de programes escrits en Pascal. D'altra banda tractarem la descripció de les fonts usades així com la seva qualitat.

Tanmateix, aquest document complementa una presentació gravada del llenguatge de programació Pascal.

2. Propòsit del llenguatge

Pascal és un llenguatge de programació de propòsit general i d'alt nivell que originalment va desenvolupar Niklaus Wirth a principis dels anys 70. Va ser molt utilitzat per ensenyar a programar de forma disciplinada i per a crear programes eficients i confiablès.

Pascal està basat en Algol i per tant inclueix moltes funcionalitats d'aquest. Tanmateix, Pascal ofereix molts tipus de dades i estructures, a més és fàcil d'entendre i mantenir.

3. Paradigma de programació

Pascal és un llenguatge imperatiu, procedural i estructurat, encara que també suporta programació orientada a Objectes, per exemple a partir d'una extensió de Pascal anomenada Object Pascal.

3.1. Programació Imperativa, Procedural i Estructurada

Pascal és un llenguatge imperatiu i per tant el seu codi és una seqüència de comandes que s'executen una darrera l'altre i que canvien l'estat del programa. A més el llenguatge és procedural, el que vol dir que pot englobar instruccions en una funció o en un procediment per tal de cridar-la quan es necessiti. Tanmateix, Pascal és un llenguatge estructurat, per tant fa un ús extensiu d'estructures de control de selecció i repetició, estructures de bloc i subrutines. Gràcies a aquest paradigma de programació, Pascal és fàcil de llegir.

```

program HelloWorld;

(* Here the main program block starts *)
begin
  writeln('Hello, World!');
  readkey;
end.

```

3.2. Programació Orientada a Objectes

Object Pascal és una extensió de Pascal que otorga programació orientada a Objectes. Aquesta extensió va ser desenvolupada per Apple Computer amb l'ajuda del creador de Pascal, Niklaus Wirth.

Object Pascal proporciona dos tipus de dades estructurals que permeten representar un objecte de la vida real:

- Objectes

Els objectes en Pascal es declaren de la següent forma:

```

type object-identifier = object
  private
    field1 : field-type;
    field2 : field-type;
    ...
  public
    procedure procl;
    function fl(): function-type;
  end;
var objectvar : object-identifier;

```

A més els objectes de Pascal poden tenir tres tipus de visibilitat i accessibilitat:

- Public
- Private
- Protected

D'altra banda els objectes també tenen els mètodes de construcció i destrucció, a aquests, normalment, se'ls anomena `init` i `done` respectivament.

- Classes

En Pascal també existeixen classes, i al contrari dels objectes, les classes sí que suporten els polimorfismes. Les classes es defineixen com un pointer a un objecte. Veiem un exemple:

```
type class-identifier = class
  private
    field1 : field-type;
    field2 : field-type;
    ...

  public
    constructor create();
    procedure procl;
    function fl(): function-type;
end;
var classvar : class-identifier;
```

Les classes disposen de les següents visibilitats i accessibilitats:

- Public
- Private
- Strict Private
- Protected
- Published

També hi ha la possibilitat de implementar herència múltiple mitjançant interfaces. Les quals es defineixen de la següent forma:

```
type
  myInterface = interface
    procedure DoThis (value: integer);
  end;
```

A més, a Pascal també existeixen les classes abstractes, les classes abstractes son aquelles que no poden ser instanciades, només heredades. I es defineixen així:

```
type
  Shape = ABSTRACT CLASS (Root)
    Procedure draw; ABSTRACT;
    ...
  end;
```

4. Sistema d'execució

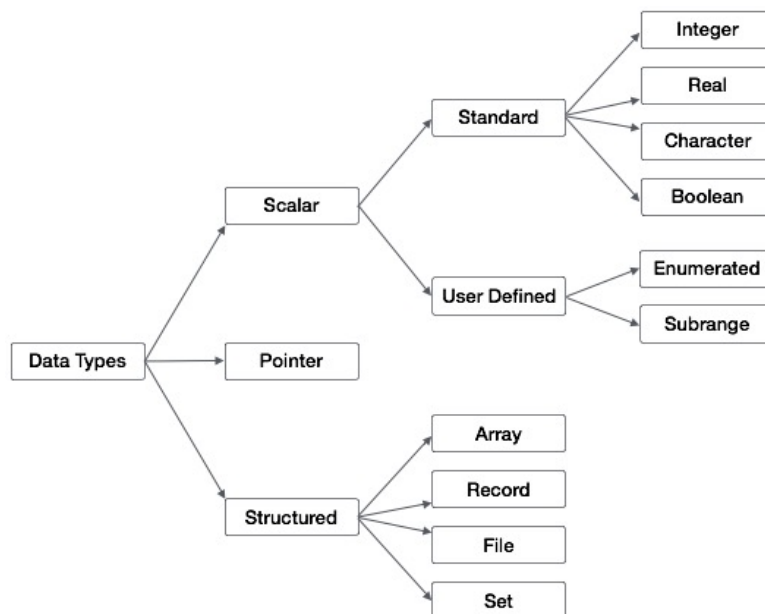
Pascal és un llenguatge compilat i interpretat. On en el seu origen era compilat, però poc a poc van anat sortint intèrprets raonablement eficients. Avui dia existeixen molts intèrprets i compiladors per Pascal i altres implementacions seves. Entre d'altres podem trobar compiladors com:

- Delphi
- Free Pascal
- Turbo51
- Oxygene
- Kylix
- GNU Pascal

A més de intèrprets com p-code per a UCSD Pascal, que va ser utilitzat per Apple.

5. Sistema de tipus

Pascal és un llenguatge de programació fortament tipat, i per tant un tipus de dades no pot ser convertit en un altre sense conversions explícites. Principalment, els tipus de dades existents a Pascal es poden resumir en aquest diagrama:



Les constants a Pascal es defineixen així:

```
const  
Identifier = constant_value;
```

A més, a Pascal es poden definir tipus enumerats i tipus amb un subrang.

Per definir variables a Pascal fem:

```
var  
variable_list : type;
```

En cas que es vulgui inicialitzar un valor en la mateixa declaració es pot fer fent un igual a continuació del tipus i fican't el valor desitjat. Si es vol fer per separat es pot fer així:

```
variable_name := value;
```

6. Principals aplicacions

Pascal és un llenguatge imperatiu i orientat a objecte, a més és fortament tipat. Això permet definir estructures de dades complexes i construir estructures de dades recursives, com per exemple llistes, arbres i graphs.

Per tot l'esmentat, Pascal és un bon llenguatge de propòsit general que permet programar de forma estructurada i sistemàtica, a més de fer programes eficients i entenedors. És per això que es va utilitzar tant com llenguatge d'aprenentatge. Amb les extensions que permeten fer programació orientada a objectes, Pascal, es converteix en un llenguatge ideal per dissenyar GUI.

7. Història del LP

A finals del anys 70, Niklaus Wirth va voler fer millores al ja existent ALGOL 60, i va presentar el ALGOL W el qual no va contentar als desenvolupadors de ALGOL, així que va crear Pascal al 1970 basant-se en el ALGOL 60 i en honor al matemàtic francès Blaise Pascal.

Pascal va tenir bastant èxit ja que era un llenguatge de programació amb moltes estructures de dades i funcionalitats que el feien molt rígid, no destacava en res però tot el que feia ho feia notablement. És per això que es va tornar molt popular com a un llenguatge de programació d'aprenentatge, sobretot en universitats dels Estats Units.

Al desenvolupament de Lisa, un dels primer computadors personals que presentava una GUI fet per Apple, Larry Tesler va fer una extensió de Pascal anomenada Clascal al 1983 que permetia fer programació orientada a objectes.

A mesura que Lisa anava desapareixent, Apple va llençar Macintosh, que estava programat en Object Pascal, una versió més avançada de Clascal. Object Pascal es va convertir en el llenguatge de programació principal de Apple als anys 90.

Amb els anys Object Pascal va servir de base per a fer Delphi, un sistema de desenvolupament amb un IDE integrat que serveix per fer aplicacions web, per mòbil, per consoles i d'escriptori. I actualment encara es fan aplicacions de Windows amb Delphi o altres versions més modernes.

D'altra banda, als anys 80 i a principis dels anys 90, amb l'aparició de C i posteriorment de C++, i gràcies a que UNIX està fet amb C, Pascal va anar desapareixent i va ser substituït per aquests llenguatges de programació més moderns i més complets.

8. Exemples de codi

A Pascal existeixen moltes formes de estructurar un programa i especificar condicions. On per exemple tenim la possibilitat de fer declaracions anidades:

```
1  program nested_ifelseChecking;
2  var
3      { local variable definition }
4      a, b : integer;
5
6  begin
7      a := 100;
8      b := 200;
9
10     (* check the boolean condition *)
11     if (a = 100) then
12         (* if condition is true then check the following *)
13         if ( b = 200 ) then
14             (* if nested if condition is true then print the following *)
15             writeln('Value of a is 100 and value of b is 200' );
16
17     writeln('Exact value of a is: ', a );
18     writeln('Exact value of b is: ', b );
19 end.
```

O fer declaracions per casos (com si fos el switch de C++):


```

1 program checkCase;
2 var
3   grade: char;
4 begin
5   grade := 'A';
6
7   case (grade) of
8     'A' : writeln('Excellent! ');
9     'B', 'C': writeln('Well done' );
10    'D' : writeln('You passed' );
11    'F' : writeln('Better try again' );
12  end;
13
14  writeln('Your grade is ', grade );
15 end.

```

I per fer loops també hi ha moltes estructures de control com podria ser el repeat-until loop:

```

1 repeat
2   sum := sum + number;
3   number := number - 2;
4 until number = 0;

```

D'altra banda, a Pascal existeixen dos tipus de mòduls de subprogrames. Les funcions i els procediments, els quals permeten executar ordres fora del mòdul principal. La diferència entre funcions i procediments es que els procediments no retornen cap valor directament i les funcions sí.

Per veure un exemple de funció veiem una funció que retorna el màxim entre dos nombres:

```

1 (* function returning the max between two numbers
2 function max(num1, num2: integer): integer;
3
4 var
5   (* local variable declaration *)
6   result: integer;
7
8 begin
9   if (num1 > num2) then
10    result := num1
11
12   else
13    result := num2;
14   max :=

```

Ara veiem un programa que escriu els nombre de fibonacci de manera recursiva utilitzant procediments:

```

1  program recursiveFibonacci;
2  var
3      i: integer;
4  function fibonacci(n: integer): integer;
5
6  begin
7      if n=1 then
8          fibonacci := 0
9
10     else if n=2 then
11         fibonacci := 1
12
13     else
14         fibonacci := fibonacci(n-1) + fibonacci(n-2);
15 end;
16
17 begin
18     for i:= 1 to 10 do
19
20         write(fibonacci (i), ' ');
21 end.

```

9. Característiques particulars

1. A Pascal, excepte els noms reservats per definir quelcom, no importa si una variable, constant, tipus, etc, té el nom en majúscula o minúscula, Pascal ho ignora per tant una variable que sigui "VARIABLE" i una altre que sigui "variable" són el mateix.
2. Per poder comentar en Pascal, es pot utilitzar parèntesi i asterisc "(* ... *)" per un comentari multilínia i brackets per un comentari d'una sola línia.
3. A Pascal hi ha els anomenats Records que permeten definir tipus de dades que contenen diferents ítems de diferents tipus de dades.
4. Els Variants permeten definir variables que poden contenir diferents tipus de dades.
5. Per defecte, Pascal utilitza crida per valor per passar els valors als subprogrames. Encara que també es poden passar els valors per referència.

10. Conclusió i visió personal

Per concloure, podem afirmar que Pascal ha tingut una gran influència en molts llenguatges de programació i és l'antecessor dels coneguts i altament utilitzats C i C++. Pascal és un llenguatge molt ben estructurat i fàcil de llegir i permet desenvolupar grans programes de forma eficient. A més, amb

les seves extensions per poder fer programació orientada a objectes, Pascal també és molt bo per a construir interfícies gràfiques.

Pel que fa a la meva opinió, Pascal és un llenguatge una mica pesat a l'hora de definir estructures i condicions, i entenc que avui dia C++ i C tinguin molta més rellevància. Tot i això crec que és un bon llenguatge de programació que podria ensenyar-se a les universitats d'avui en dia per la seva estructura tant forta i la possibilitat de definir qualsevol estructura de dades.

11. Bibliografia

[1] Viquipèdia de Pascal. [En línia]. [Consulta: 22 de desembre 2021]. Disponible a: <[https://en.wikipedia.org/wiki/Pascal_\(programming_language\)](https://en.wikipedia.org/wiki/Pascal_(programming_language))>

[2] Pascal Programming info. [En línia]. [Consulta: 22 de desembre 2021]. Disponible a: <<http://www.pascal-programming.info/index.php>>

[3] Tutorials Point Pascal. [En línia]. [Consulta: 22 de desembre 2021]. Disponible a: <<https://www.tutorialspoint.com/pascal/index.htm>>

[4] Was Pascal Influential? [En línia]. [Consulta: 23 de desembre 2021]. Disponible a: <<https://www.quora.com/Was-the-programming-language-Pascal-influential-If-so-then-why>>

[5] Marco Cantù's Essential Pascal Web. [En línia]. [Consulta: 23 de desembre 2021]. Disponible a: <<https://www.marcocantu.com/epascal/English/default.htm>>

[6] Marco Cantù's Essential Pascal Book. [En línia]. [Consulta: 23 de desembre 2021]. Disponible a: <<https://www.marcocantu.com/epascal/EssentialPascal.zip>>

[7] The Pascal Family of Programming Languages. [En línia]. [Consulta: 24 de desembre 2021]. Disponible a: <<https://safeswisscloud.com/en/blog/pascal-family-programming-languages/>>

12. Descripció de les fonts d'informació

A l'hora de buscar informació de Pascal, he fet una primera cerca per internet per poder veure per on començar el treball de recerca. El primer recurs que he trobat és la viquipèdia en anglès ja que disposa de força informació de Pascal i a priori sembla de qualitat.

A partir de viquipèdia he anat trobant pàgines web antigues (totes elles en anglès), que donaven nocions de Pascal. Amb aquestes pàgines, com per exemple TutorialsPoint o Pascal Programming info, he après i conegut el funcionament en detall de Pascal. A més he trobat un llibre de Marco Cantù's sobre els essencials de Pascals que m'ha servit per aprofundir en algunes àrees del llenguatge. Aquest llibre està disponible en línia a la seva pàgina web.

D'altra banda, per entendre el context del llenguatge i quina influència va tenir als anys 70 i 80 he recurrit a fòrums i blogs, on gent que va viure l'època discuteix sobre que va suposar Pascal en la seva època. A l'hora de trobar història crec que la font principal ha estat viquipèdia.

13. Avaluació de la qualitat de la informació trobada

Per la meva sorpresa, molta informació que circula per internet sobre Pascal, sembla bastant acurada. Tot el que té a veure amb codi i funcionament del llenguatge, totes les fonts consultades diuen el mateix. Per tant no he tingut cap problema per entendre les peculiaritats de Pascal.

D'altra banda, pel que fa a història i context de Pascal, m'ha costat una mica més formar una idea pròpia i consensuada, ja que a cada web visitada, es deien unes coses i es deixaven de dir altres. Això és totalment comprensible ja que el contexte i la història tenen un punt de subjectivitat que el llenguatge en sí no té. És per això que els apartats que tenen a veure amb això poden ser menys acurats.

Per concloure, encara que Pascal és un llenguatge de programació antic, tinc la sensació que trobar bona informació sobre ell és més fàcil que trobar informació bona de llenguatges actuals. Això segurament es deu a que poques persones tenen interès en Pascal i la gent que publica informació ho fa per passió i gust, sense buscar cap interès propi.