John Wilder Tukey

Resolución de Problemas y Algoritmos

Clase 4
Estructura de control condicional.

Dr. Alejandro J. García
http://cs.uns.edu.ar/~ajg

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Conceptos de las clases anteriores

- **Algoritmo. Primitiva. Traza.**
- **Lenguaje de programación. Programa. Código fuente.**
- **Compilador. Código ejecutable.**
- **Lenguaje de programación Pascal:**
  - **Palabras reservadas e identificadores**
  - **Constantes y variables**
  - **Tipos de datos. Tipos predifinidos.**
  - **Primitiva de asignación (:=)**
  - **Primitivas para interacción con el usuario (read & write)**
  - **Expresiones. Operaciones y funciones predefinidas.**

¿Preguntas?

Concepto: lenguaje de programación

Un ***lenguaje de programación*** es un lenguaje artificial creado para expresar procesos que pueden ser llevados a cabo por computadoras. (Ej: Pascal)

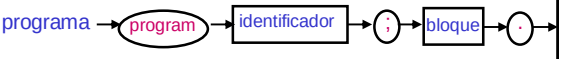
Está definido por un conjunto de símbolos, **reglas sintácticas** que definen su estructura, y reglas semánticas que definen el significado de sus elementos.

¿Cómo puedo conocer el conjunto de las reglas sintácticas de Pascal Estándar?

Conceptos: Sintaxis y Diagrama Sintáctico

- **La sintaxis de un lenguaje es un conjunto de reglas que indica como escribir programas.**
- **Un diagrama sintáctico es una descripción gráfica de la sintaxis de un lenguaje de programación. Permite describir sin ambigüedad la sintaxis de un lenguaje de una manera simple y formal.**
- **Está compuesto por cuatro tipos de elementos.**

Ejemplo:



(p)

Elementos de un diagrama sintáctico

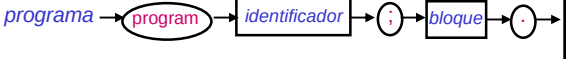
nombre → (1) **Un nombre y flecha** indican el comienzo de un diagrama para la definición de nombre.

texto — (2) **Las figuras "redondeadas"** indican que texto se debe **incluir tal cual** como aparece.

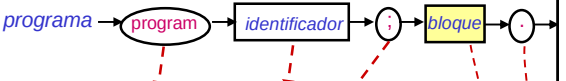
nombre — (3) Los **rectángulos** indican que nombre está **definido en algún otro diagrama** sintáctico.

→ (4) Las **flechas** indican el **orden** de lectura en el diagrama.

Ejemplo:

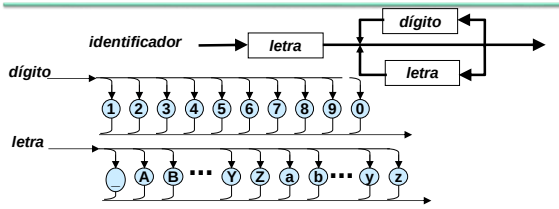


Diagramas sintácticos




```
PROGRAM AreaCirculo;
CONST pi = 3.1416;
VAR area,radio: REAL;
BEGIN
    read(radio);
    area := pi * radio * radio;
    write("Area es", area);
END.
```

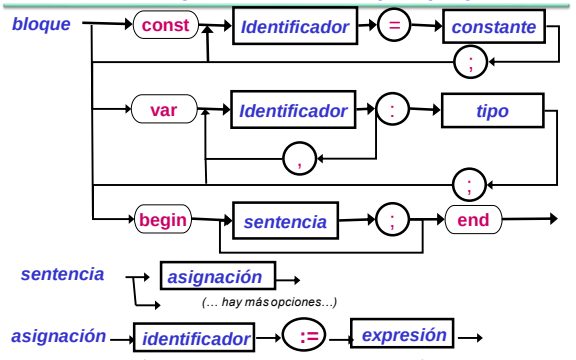
Diagramas sintácticos para "identificador"



- En letra se puede utilizar el símbolo "_" (underscore), las mayúsculas: ABCDEFGHIJKLMNOPQRSTUVWXYZ y las minúsculas: abcdefghijklmnopqrstuvwxyz
- No pueden utilizarse por ejemplo: á, ó, Ú, ñ, Ñ, -.

IMPORTANTE: Los diagramas sintácticos completos pueden encontrarse en el material de la página web de la materia.

Parte de los diagramas sintácticos para programa (p)



Uso de punto y coma

Para separar sentencias en Pascal se utiliza el símbolo ";" **(punto y coma).**

Observación: puede habar varias sentencias en un mismo renglón y una sentencia puede tener varios renglones.

```
i := 1;  j:=2;  k:= 3;
B:= (i>0) and (k = j);
```

equivalente

```
i := 1;
j:=2;
k:= 3;
B:= (i>0)
and (k=j);
```


Sentencias en Pascal

Las sentencias en Pascal pueden ser simples o compuestas. Ejemplos:

Sentencia o proposición

- **simple**: `a:=1`
- **compuesta**:

```
BEGIN
  PrecioBase := 200;
  Iva:= Precio * 0.20;
  PrecioFinal:= PrecioBase+ iva;
END
```


Sentencia Compuesta en Pascal

Una sentencia compuesta comienza con **BEGIN** y termina con **END** **y permite definir una secuencia de sentencias como si fuera una única sentencia.**

Por ejemplo, la siguiente es una sentencia (compuesta, a su vez, por tres sentencias simples)

```
BEGIN
  PrecioBase := 200;
  Iva:= Precio * 0.20;
  PrecioFinal:= PrecioBase+ iva;
END
```

Las **sentencias** que forman una sentencia compuesta se separan una de otra con **punto y coma**

Problema simple propuesto

Problema: Escriba un programa en Pascal para obtener el valor absoluto de un número.

Solución:

Si el número es positivo, el valor absoluto es el mismo número y sino es el número multiplicado por -1.

Algoritmo:

Leo el número Num
Si Num > 0
entonces: val_abs es Num
de lo contrario: val_abs es Num * -1
Muestro val_abs en pantalla

Verificación:

ejemplos significativos 3, 0 y -3

Estructura de control condicional (1)

```
IF expresión boolean
THEN Sentencia 1
   (simple o
   compuesta)
ELSE Sentencia 2
   (simple o
   compuesta)
```

Ejemplo (valor absoluto de un número):

```
Write ('Ingrese…'); read(numero);
IF numero > 0
THEN val_abs := numero
ELSE val_abs := (-1) * numero;
writeln(' Valor absoluto: ', val_abs);
```

Obs: no lleva ";" antes del ELSE

Sintaxis: ver el diagrama sintáctico.
Semántica: Si la evaluación de la expresión lógica da verdadero, entonces se ejecuta solamente la sentencia que sigue al "THEN". Si la evaluación de la expresión lógica da falso, entonces se ejecuta sólo la sentencia que sigue al "ELSE"

Estructura de control condicional (2)

```
IF expresión boolean
THEN
   Sent. 1 (simple o
   compuesta)
```

```
program leer_char;
var   ch: char;
begin
write('Ingrese un caracter:');
readln(ch);
IF  (ch >= 'A') and (ch <= 'Z')
   then writeln('mayúscula.');
…
```

- **Es un caso particular del IF-THEN-ELSE.**
- **Semántica: Si la evaluación de la expresión lógica da verdadero, entonces se ejecuta solamente la sentencia que sigue al "THEN". Si da falso, sigue con la sentencia siguiente.**

Problema: Escriba un programa que lea un caracter (CHAR) y diga si se trata de una letra mayúscula, o una letra minúscula.

Solución: de 'A' a la 'Z' es una mayúscula, de 'a' a la 'z' es una minúscula.

Algoritmo:

- **leer el caracter**
- **Si está entre 'A' y 'Z' entonces es una mayúscula**
- **Si está entre 'a' y 'z' entonces es una minúscula**

Verificación:

ejemplos significativos 'G', 'g', '3' , '$'

Un programa posible…

```
program leer_char;
var   ch: char;
{Este programa permite distinguir mayúsculas y
   minúsculas}
begin
write('Ingrese un caracter:');
readln(ch);
IF  (ch >= 'A') and (ch <= 'Z')
   then writeln(ch, ' es una mayúscula.');
IF  (ch >= 'a') and (ch <= 'z')
   then writeln(ch, ' es una minúscula.');
end.
```

Hacer una traza
¿qué ejemplos usa?

Problema propuesto

Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, o un símbolo distinto a los anteriores.
por ejemplo:
'G', es una mayúscula
'g', es una minúscula
'3' , es otro símbolo
'$', es otro símbolo

Copie del pizarrón las soluciones propuestas y los programas incorrectos (y porque lo son)

Condicionales "anidados"

```
IF <exp. boolean>
THEN
   Sentencia
   (simple o
   compuesta)
ELSE
   Sentencia
   (simple o
   compuesta)
```

```
IF  <exp. Boolean E1 >
THEN
   IF < exp. boolean E2 >
   THEN <sentencia s. o c. s1>
   ELSE <sentencia s. o c. s2>
ELSE
   IF < exp. Boolean E3 >
   THEN <sentencia s. o c. s3>
   ELSE <sentencia s. o c. s4>
```

Conceptos: Hardware [Wikipedia]

Hardware es una palabra inglesa (literalmente: partes duras); como no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: *«Conjunto de los componentes que integran la parte material de una computadora».*

Hardware corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes, periféricos de todo tipo y cualquier otro elemento físico involucrado;

contrariamente, al soporte lógico que es intangible y es llamado ***software***.

Conceptos: Software [Wikipedia]

Software es una palabra inglesa (literalmente: partes blandas o suaves); como en español no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: *«Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora ».*

Se conoce como ***software*** *al equipamiento lógico o soporte lógico* de un sistema informático; comprende el conjunto de los componentes **lógicos** necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados ***hardware***.

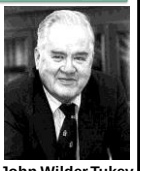
Software [Wikipedia]

Existen varias definiciones similares aceptadas para software, pero probablemente la más formal sea la siguiente: « Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación» [Extraído del estándar 729 del IEEE]

Considerando esta definición, el concepto de software va más allá de los programas de computación; también su documentación, los datos a procesar e incluso la información de usuario forman parte del software: es decir, *abarca todo lo intangible*, todo lo «no físico» relacionado.

Historia [Wikipedia]

- John W. Tukey usó el termino "Software de Computación" (Computer Software) en un artículo de 1958 en el *American Mathematical Monthly*, aparentemente fue el primer uso de este término.
- Además, mientras trabajaba con John von Neumann en los primeros diseños de las primeras computadoras, Tukey introdujo la palabra "bit" como contracción de las palabras *Binary Digit* ("Dígito binario")



John Wilder Tukey

Continuará