

INTRODUCCIÓN A LA PROGRAMACIÓN

PRACTICA 3

Objetivos

- ✓ Presentar las primeras estructuras de control de Pascal: estructuras de selección **if** y **case**.
- ✓ Estudiar los operadores de Pascal para la composición de expresiones booleanas
- ✓ Aplicar las estructuras de control vistas para hacer programas sencillos.

1. Análisis de Casos en Pascal

Notación SP	Pascal
SI <i>cond</i> ENTONCES <i>S</i> FIN_SI	IF <i>cond</i> THEN BEGIN <i>S</i> END
SI <i>cond</i> ENTONCES <i>S</i> SI_NO <i>R</i> FIN_SI	IF <i>cond</i> THEN BEGIN <i>S</i> END ELSE BEGIN <i>R</i> END
SEGÚN <i>C₁, C₂, ..., C_n</i> <i>e₁</i> : <i>S₁</i> <i>e₂</i> : <i>S₂</i> ... <i>e_n</i> : <i>S_n</i> { EN OTRO CASO <i>S_{n+1}</i> } FIN_SEGUN	IF <i>e₁</i> THEN BEGIN <i>S₁</i> END ELSE IF <i>e₂</i> THEN BEGIN <i>S₂</i> END ELSE IF <i>e₃</i> THEN BEGIN <i>S₃</i> END ... ELSE IF <i>e_n</i> THEN BEGIN <i>S_n</i> END { ELSE BEGIN <i>S_{n+1}</i> END }

En la tabla, **cond**, indica una condición, es decir una expresión booleana cuyo resultado es verdadero o falso; **S** y **R** denotan un bloque de instrucciones. La semántica de la instrucción **IF** es igual a la estudiada en clase para la composición **SI**.

NOTA: Cuando sólo hay una sentencia en un caso ésta puede ir sola, sin necesidad de estar encerrada entre un par **BEGIN-END**. Esto solo es necesario cuando hay más de una sentencia que debe ejecutarse en un caso.

Como la mayoría de lenguajes de programación, Pascal dispone de una estructura de selección múltiple llamada **case**. Aunque esta estructura aparentemente es similar al **SEGÚN** de la notación algorítmica, presenta importantes diferencias que se enumeran a continuación:

- Sólo es posible efectuar selecciones en función del valor de una única expresión, a diferencia del **SEGÚN** en el que podíamos poner tantas como fuese preciso.
- La expresión en función de la cual se efectúa la selección ha de ser necesariamente de un tipo ordinal.
- Los valores con los que se compara esta expresión deben ser *estáticos*. Esto quiere decir que podemos comparar con valores literales, constantes o expresiones cuyo valor sea conocido en tiempo de compilación, y no con valores de expresiones cuyo valor dependa, por ejemplo, de variables.

La sintaxis de la estructura case es la siguiente:

```

case expresión_ordinal of
    selector_1 :
        sentencias_1;
    selector_2 :
        sentencias_2;
    ...
    {else
        sentencias_else;}
end;

```

La sentencia **case** evalúa la *expresión_ordinal* que va a continuación de la palabra **case** y después trataría de encajar el valor obtenido con el de alguno de los selectores. Si el valor encaja con el *selector_i* entonces se ejecuta la secuencia de sentencias asociadas a este selector (*sentencias_i*). Si no encaja con ningún selector, se ejecutarían las sentencias asociadas a la cláusula **else**, si esta cláusula existe (*sentencias_else*). Esta última cláusula, no tiene porqué aparecer siempre.

Los selectores pueden estar compuestos por los siguientes elementos:

- Literales o valores constantes del tipo de la *expresión_ordinal*
- Intervalos del tipo de la *expresión_ordinal*, en la forma:
límite_inferior .. *límite_superior*
- Varios elementos de los tipos anteriores separados por comas

En el siguiente cuadro, podemos ver un ejemplo de utilización de esta sentencia

```

Program Ejemplo_case;

var
    c : char;
begin
    write ('Introduce un carácter: ');
    read (c);
    case c of
        'a'..'z':           Writeln ('Es una minúscula');
        'A'..'Z':           Writeln ('Es una mayúscula');
        '0'..'9':           Writeln ('Es un dígito');
        ',', '.', ':', ';': Writeln ('Es un signo de puntuación')
        else                Writeln ('Carácter desconocido');

    end
end.

```

Problema 1

Teniendo en cuenta el párrafo anterior transcriba a Pascal los dos ejemplos siguientes vistos en clase y compruebe la semántica de la composición alternativa con diferentes ejecuciones.

Algoritmo 2.8

Léxico

nota: entero;

Algoritmo

Leer (nota);

SEGÚN nota

nota = 20:

Escribir ("M. De Honor");

(nota = 19) **O** (nota = 18):

Escribir ("Sobresaliente");

(nota = 17) **O** (nota = 16):

Escribir ("Notable");

(nota = 15) **O** (nota = 14):

Escribir ("Aprobado");

EN OTRO CASO:

Escribir ("Suspenso");

FIN_SEGÚN

Fin.

Algoritmo 2.10

Léxico

n1, n2 :entero;

Algoritmo

Leer (n1, n2);

SI n1 > n2 **ENTONCES** Escribir ("El máximo es ", n1)

SI_NO Escribir ("El máximo es", n2)

FIN.

Problema 2:

Un banco realiza préstamos rápidos a sus clientes por un período entre 1 y 24 meses. Para calcular el interés total a pagar, el banco multiplica la cantidad prestada, el número de meses que durará el préstamo y el interés mensual. Si el cliente tiene otras deudas contraídas con el banco, se considera que es un cliente de riesgo, y se le aplica un recargo extra de un 1% sobre la suma de la cantidad prestada y el interés total. El total a devolver, esto es, la suma de la cantidad prestada, los intereses y el recargo, se distribuye en plazos mensuales iguales a lo largo del período que dura el préstamo.

Escriba un algoritmo que dados los valores de entrada: la cantidad prestada en euros, un entero que representa el número de meses, el interés mensual (tanto por ciento) y si es un cliente de riesgo o no, calcule la cantidad mensual que debe abonar el cliente

Problema 3:

Diseñe un algoritmo y transcribalo en forma de programa Pascal que lea de la entrada una fecha en forma de tres números positivos (día, mes, año) y ofrezca como salida la fecha del día siguiente. Supondremos que el año no es bisiesto. A continuación se muestran algunos ejemplos de entradas con sus salidas correctas correspondientes:

19	6	2005	→	20	6	2005
30	4	1990	→	1	5	1990
31	12	2010	→	1	1	2011

Trabajo personal del alumno

Transcriba a Pascal el problema correspondiente al algoritmo 2.9 del libro de texto: Escriba un algoritmo que simule el funcionamiento de una calculadora. Los datos de entrada están constituidos por dos enteros (operandos) y un carácter (símbolo de operación, +, -, *, /). El resultado es el valor de la expresión dada.

La compañía de seguros

Estudie el siguiente problema:

Una compañía de seguros de vida, ante un nuevo cliente emplea las siguientes reglas: Si tiene menos de 30 años, buena salud, y no ha tenido nunca un accidente, ofrece un contrato de tipo A, si tiene mala salud o ha tenido un accidente llama a un experto, si tiene mala salud y ha tenido ya un accidente no le hace el seguro. Si tiene mas de 30 años aplica las mismas reglas pero ofrece un contrato de tipo B, en lugar de un contrato de tipo A. A continuación se muestra un ejemplo de análisis del problema.

Cuestiones a tener en cuenta para resolver el problema

✓ ¿Qué variables de entrada intervienen?

- Mas de treinta o menos de treinta variable *edad* (*mas de treinta verdad*)
- Buena o mala salud variable *salud* (*buena verdad*)
- Accidente o no accidente variable *accidente* (*lo ha tenido verdad*)

✓ ¿Qué queremos obtener?, uno de estos tres mensajes:

- Contrato A
- Contrato B
- Llamar a experto
- Rechazado

Análisis de casos en función de los datos de entrada

edad	salud	accidente	resultado
V	V	V	Llamar experto
V	V	F	Contrato B
V	F	V	Rechazado
V	F	F	Llamar experto
F	V	V	Llamar experto
F	V	F	Contrato A
F	F	V	Rechazado
F	F	F	Llamar expero

✓ ¿Se cubre todos los casos?

✓ ¿Se solapan?

A la hora de escribir el algoritmo agrupamos los casos por tipo de salida y simplificamos las variables que se pueda.

Léxico

salud, accidentes, edad: BOOLEANO;
respuesta: CARÁCTER;

Algoritmo

Escribir ("Tiene el cliente buena salud? S/N ");
Leer (respuesta);
salud ← respuesta = 'S';
Escribir ("Ha sufrido accidentes? S/N ");
Leer (respuesta);
accidentes ← respuesta = 'S';
Escribir ("Tiene más de 30 años? S/N ");
Leer(respuesta);
edad ← respuesta = 'S';

SEGUN salud, accidentes, edad

(salud **Y** accidentes) **O** (**no**(salud) **Y no**(accidentes)) :

Escribir ("Llamar a un experto");

no (salud) **Y** accidentes:

Escribir (" Rechazado");

edad **Y** salud **Y no** (accidentes):

Escribir ("Contrato B");

no (edad) **Y** salud **Y no** (accidentes):

Escribir ("Contrato A");

FIN_SEGUN;

Fin.

✓ Codifique el programa en Pascal

Triángulos

Escriba un programa que lea tres números reales y asigne el valor apropiado **true** o **false**, a las siguientes variables booleanas:

- Triangulo: **true** si los números reales pueden representar longitudes de los lados de un triangulo, y **false** en caso contrario (la suma de dos cualesquiera de las longitudes debe ser mayor que la tercera).
- Equilatero: **true** si Triangulo tiene el valor **true** y además sus tres lados son iguales.
- Isósceles: **true** si Triangulo tiene el valor **true** y además al menos dos de los lados son iguales
- Escaleno: **true** si Triangulo tiene el valor **true** y además no tiene lados iguales

- ✓ Realice el análisis del problema.
- ✓ Escriba el algoritmo en notación SP.
- ✓ Transcriba el algoritmo a programa Pascal.

Codificación de caracteres

Escriba un programa en Pascal que leerá un carácter y escribirá en su lugar un carácter codificado. Si el carácter es letra o dígito, lo reemplazara por el siguiente carácter, excepto en el caso de **Z**, que será reemplazada por la **A**; **z**, que será reemplazada por **a**; y **9** que será reemplazado por **0**. De esta forma, el 1 se convierte en 2, C en D, p en q, etc. Cualquier carácter distinto de letra o dígito se sustituirá por un asterisco.

- ✓ Realice el análisis del problema.
- ✓ Escriba el algoritmo en notación SP.
- ✓ Transcriba el algoritmo a programa Pascal.

Calificaciones

Escriba un algoritmo y transcríballo luego a Pascal que, al igual que el problema dos de la segunda práctica, lea tres notas de un alumnos, ponderadas al 50%, 30% y 20%, respectivamente. A continuación calcule la nota final, pero en esta ocasión lo que debe mostrar es la calificación en formato alfabético:

‘Suspendido’:	si la nota no llega al cinco
‘Aprobado’:	si es mayor o igual que cinco y menor que siete
‘Notable’:	si es mayor o igual que siete y menor que nueve
‘Sobresaliente’:	si es mayor o igual que nueve y menor que diez
‘Matrícula’:	si la nota es diez