




Resolución de Problemas y Algoritmos

Clase 3
Programación en Pascal



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 Pascal:
 - Constantes y Variables
 - Primitiva de Asignación
- Expresiones numéricas
- Expresiones lógicas

¿Qué recuerdan de las clases anteriores?



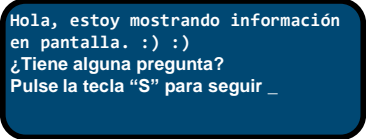
Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

Interacción con el usuario

Es usual que un programa interactue con el usuario:

- mostrando información en pantalla
- esperando por datos ingresados por el usuario

HOY en la práctica se verán detalles sobre las primitivas de Pascal para mostrar por pantalla y obtener valores ingresados por teclado.



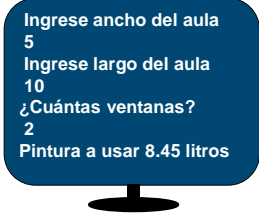
Hola, estoy mostrando información en pantalla. :) :)
¿Tiene alguna pregunta?
Pulse la tecla "S" para seguir _

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

Interacción con el usuario

Por ejemplo, se podría hacer que el programa "pintura_aula":

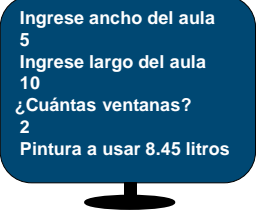
- pida los datos del ancho y largo de un aula y la cantidad de ventanas del aula, y luego
- muestre por pantalla el resultado calculado.



Ingrese ancho del aula
5
Ingrese largo del aula
10
¿Cuántas ventanas?
2
Pintura a usar 8.45 litros

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

```
PROGRAM pintura_aula;
CONST alto = 3; puerta = 3.20;
      ventanas = 2; cubrelitro = 8;
VAR
  ancho, largo, a_no_pintar: REAL;
  cant_litros, a_pintar :REAL;
  cant_ventanas:INTEGER;
BEGIN
  writeln('Ingrese ancho del aula');
  readln(ancho);
  writeln('Ingrese largo del aula'); readln(largo);
  writeln('¿Cuántas ventanas?'); readln(cant_ventanas);
  a_no_pintar:= 2*puerta + 2 * cant_ventanas ;
  a_pintar := 2*(ancho*alto)+2*(largo*alto) - a_no_pintar;
  cant_litros:= a_pintar / cubrelitro;
  writeln('Pintura a usar =', cant_litros,'litros');
END.
```



Ingrese ancho del aula
5
Ingrese largo del aula
10
¿Cuántas ventanas?
2
Pintura a usar 8.45 litros

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

Conceptos: tipos de datos

Tipo de Dato: define el conjunto de valores posibles que puede tomar una variable, y también define las operaciones que puede usarse.

Algunos tipos predefinidos en Pascal Estándar:

INTEGER: es un subconjunto de los números enteros

REAL: es un subconjunto de los números reales.

BOOLEAN: es el conjunto {true, false} (valores lógicos)

CHAR: es el conjunto de los 256 caracteres del código ASCII (letras, dígitos, y símbolos)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
"Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

El código ASCII

American Standard Code for Information Interchange
Está formado por 256 símbolos, aquí se muestran algunos:

		32	33	!	34	"	35	#	36	\$	37	%	38	&	39	'			
40	(41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0	49	1
50	2	51	3	52	4	53	5	54	6	55	7	56	8	57	9	58	:	59	;
60	<	61	=	62	>	63	?	64	@	65	A	66	B	67	C	68	D	69	E
70	F	71	G	72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X	89	Y
90	Z	91	[92	\	93]	94	^	95	_	96	`	97	a	98	b	99	c
100	d	101	e	102	f	103	g	104	h	105	i	106	j	107	k	108	l	109	m
110	n	111	o	112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124		125	}	126	~	127		128	Ç	129	ü

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

Ejemplos

Tipos Predefinidos

- REAL
- BOOLEAN: true, false
- CHAR todos los símbolos del código ASCII
- INTEGER rango de enteros

Indique un tipo de dato apropiado para cada variable:

VAR Precio: REAL;
Es_Nro_Primo: BOOLEAN;
Letra: CHAR;
Día: INTEGER;
Mes: INTEGER;

Tener tipos de datos para las variables permite claridad y abstracción. Dos conceptos fundamentales en el desarrollo, mantenimiento y futuras actualizaciones del software.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

Operaciones del Tipo de Dato Simple INTEGER

Operador	operación	operandos	resultado
+	suma	Integer	integer
-	resta	Integer	integer
*	producto	Integer	integer
div	división entera	Integer	integer
mod	resto	Integer	integer

Función SQR: devuelve el cuadrado (square) de un entero. Ejemplo: SQR(3) = 9.

Constante predefinida MAXINT: Es el máximo valor del tipo INTEGER.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

Realice una traza y luego pase a la máquina

```

PROGRAM Ejemplo1;
    {Algunos ejemplos para el tipo entero}
VAR N1,N2,N3,N4,N5,N6:INTEGER;
BEGIN
    N1 := 1+ 2000 mod 2;
    N2 := SQR(9);
    N3 := SQR(SQR(3));
    N3 := N3+N2;
    N4 := MAXINT;
    N5 := 1+ N4 ; {¿qué valor toma N5?}
    N3 := N6; {error de programación: N6 no tiene valor}
END.
    
```

Recuerde: Usar una variable sin valor es un error de programación.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

Tipo de Dato Simple REAL

Tipo REAL : Corresponde a un subconjunto de los números reales. Pascal permite reales con **punto decimal** (ej : 3.5459), o en **notación científica**:
 Ej: $3.5 \cdot 10^{-3} = 0.0035$ en Pascal es **3.5E-3**
 Ej: $1.28 \cdot 10^8 = 128000000$ en Pascal es **1.28E8**

Operador	Operación	Operandos	Resultado
+	suma	real o integer	real (1)
-	resta	real o integer	real (1)
*	mult.	real o integer	real (1)
/	división real	real o integer	real

(1) El resultado es real cuando al menos un operando es real

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

Algunas funciones predefinidas para REAL

Funciones trigonométricas: SIN, COS y TAN. Dado un valor de un ángulo (en radianes), devuelven su seno, coseno o tangente. Ejemplos: SIN(0) = 0, COS(0) = 1

Función raíz cuadrada (square root) SQRT
Ejemplo: SQRT(4) = 2.0

Función de redondeo ROUND: dado un valor real, devuelve el entero más cercano.
Ejemplos: ROUND(2.9) = 3 ROUND(2.3) = 2

Función truncado TRUNC: dado un valor real, devuelve el entero que resulta de eliminar la parte decimal.
Ejemplos: TRUNC (2.9) = 2 TRUNC(2.3) = 2

(p)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

Realice una traza y luego pase a la máquina

```
PROGRAM Ejemplo2;
{Algunos ejemplos para el tipo real}
VAR N1,N2:INTEGER;
    R1,R2: REAL;
BEGIN
  R1 := 20 mod 2;
  R2 := MAXINT + 1;
  N1 := TRUNC(2.5);
  N2 := ROUND(2.5);
END.
```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

13

Tipo de Dato Simple BOOLEAN

Este tipo de dato tiene únicamente dos valores asociados, **true** y **false** (corresponden a **verdadero** y **falso**).

Los operadores booleanos **y**, **o** y **no** se representan en Pascal con las palabras reservadas **and**, **or** y **not**.

- Puedo utilizar los símbolos = > y <
- El símbolo ≠ se escribe <>
- El símbolo ≥ se escribe >= o =>

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

14

Realice una traza y luego pase a la máquina

```
PROGRAM Ejemplo3;
VAR R1: REAL;
    es_par, positivo, mayor_a_maxint:BOOLEAN;
    condicion:BOOLEAN;
BEGIN
  R1 := 23;
  Es_par := (TRUNC(R1) mod 2) <> 1;
  positivo := R1 >= 0;
  Mayor_a_maxint := R1 > MAXINT;
  Condicion:= es_par and es_positivo
              and not mayor_a_maxint;
END.
```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

15

Tipo CHAR : es el conjunto de los 256 símbolos ASCII.

¿Cómo se diferencia entre una variable cuyo identificador es A y el símbolo ASCII A ?

- Para indicar un valor de tipo CHAR, se utilizan las comillas simples. Ej : 'a', '?', '+', ' ', etc.
- Por lo tanto, en Pascal: 'A' es una letra, y en cambio A es un identificador.

Función CHR: permite obtener un caracter cualquiera a partir de su código ASCII.

Ejemplos: chr(65) = 'A'; chr(33) = '!'. (p)

Función ORD: dado un caracter cualquiera, devuelve su código ASCII.

Ejemplos: ord('A') = 65, ord('!') = 33. (p)

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

16

El lenguaje de programación Pascal

Niklaus Wirth fue el jefe de **diseño** de los **lenguajes de programación Euler, Algol W, Pascal, Modula, Modula-2** y **Oberon**. Su artículo de "Desarrollo de un programa por refinamiento sucesivo" es considerado un texto clásico en la **ingeniería del software**, así como su libro: "Algoritmos + Estructuras de datos = Programas", que recibió un amplio reconocimiento, y que aun hoy en día es útil en la enseñanza de la **programación**.

Recibió el **Premio Turing** por el desarrollo de estos **lenguajes de programación** en 1984.

Se jubiló en 1999.



Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

17

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:



Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

18

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

"Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

(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:

programa → **program** → **identificador** → **:** → **bloque** → **.**

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 19

Diagramas sintácticos

programa → **program** → **identificador** → **:** → **bloque** → **.**

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 20

Diagramas sintácticos para "identificador"

identificador → **letra** → **dígito** → **letra**

dígito → 1 2 3 4 5 6 7 8 9 0

letra → A B ... Y Z a b ... y z

Las letras incluyen solamente el símbolo "_" (underscore), las mayúsculas: ABCDEFGHIJKLMNOPQRSTUVWXYZ y las minúsculas: abcdefghijklmnopqrstuvwxyz. No pueden utilizarse por ejemplo: á, ó, Ú, ñ, Ñ, -.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 21

Parte de los diagramas sintácticos para programa (p)

bloque → **const** → **Identificador** → **=** → **constante** → **:**

bloque → **var** → **Identificador** → **:** → **tipo** → **:**

bloque → **begin** → **proposición** → **:** → **end** → **:**

proposición → **asignación** → **:**

asignación → **Identificador** → **:=** → **expresión**

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 22

Uso de punto y coma

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

Observación: puede haber 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);

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 23

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 24

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 25

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
Sino val_abs es Num * -1
Muestro val_abs en pantalla

Verificación:
ejemplos significativos 3, 0 y -3

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 26

Estructura de control condicional (1)

IF expresión lógica
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 en cambio la evaluación de la expresión lógica da falso, entonces se ejecuta sólo la sentencia que sigue al “ELSE”

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 27

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

Solución: de ‘A’ a la ‘Z’ es una mayúscula, de ‘a’ a la ‘z’ es una minúscula y de ‘0’ a ‘9’ un dígito.

Algoritmo:

- leer el caracter
- Si está entre ‘A’ y ‘Z’ entonces es una mayúscula
- Si está entre ‘a’ y ‘z’ entonces es una minúscula
- Si está entre ‘0’ y ‘9’ entonces es un dígito

Verificación:
ejemplos significativos ‘G’, ‘g’, ‘3’, ‘\$’

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 28

Estructura de control condicional (2)

IF Exp. lógica
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. ');
...
        
```

- Este 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.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 29

Programa del problema anterior

```

program leer_char;
var ch: char;
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. ');
IF (ch >= '0') and (ch <= '9')
  then writeln(ch, 'es un dígito. ');
end.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 30

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

Tarea para el hogar ☺

Problema propuesto:

- Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, un dígito, o un símbolo distinto a los anteriores.

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

31

- En el problema de calcular la cantidad de litros para pintar un aula, ¿cómo calcular la cantidad de latas de 4 litros a comprar?

```
PROGRAM pintura_aula;
CONST alto = 2.60; puerta = 3.20; ventanas = 2;
      cubrelitro = 8; litroslata = 4;
VAR ancho, largo, a_no_pintar, a_pintar, cant_litros, cant_ventanas, cant_latas:REAL;
BEGIN
  ancho := 5;
  largo := 10; cant_ventanas := 2;
  a_no_pintar := 2*puerta + 2 * cant_ventanas ;
  a_pintar := 2*(ancho*alto)+2*(largo*alto)-a_no_pintar;
  cant_litros := a_pintar / cubrelitro;
  cant_latas := cant_litros / litroslata;
END.
```

Solución:
SI cant_litros dividido cubrelitro da justo
ENTONCES cant_latas es la división
SINO tengo que considerar una lata mas

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

32

Dos soluciones posibles

```
...
IF (cant_litros / litroslata) = TRUNC (cant_litros / litroslata)
THEN cant_latas := TRUNC(cant_litros / litroslata)
ELSE cant_latas := TRUNC(cant_litros / litroslata) + 1;
writeln(' Latas a usar: ', cant_latas);
```

```
...
cant_latas := TRUNC(cant_litros / litroslata);
IF cant_latas < (cant_litros / litroslata)
THEN cant_latas := cant_latas + 1;
writeln(' Latas a usar: ', cant_latas);
```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

33

Herramientas a disposición de los alumnos

Herramientas brindadas por la cátedra:

1. clase teórica
2. clase práctica
3. ejercicios en los prácticos
4. horarios de laboratorio
5. evaluación en máquina y parciales (obligatorio)

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

34

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.