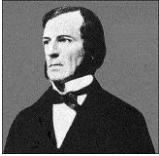



Resolución de Problemas y Algoritmos

Clase 3
Programación en Pascal.
Tipos de datos. Expresiones




George Boole



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.
- Lenguaje de programación Pascal:
 - Palabras reservadas e identificadores
 - Constantes y variables
 - Tipos de datos
 - Primitiva de asignación (:=)
 - Primitivas para interacción con el usuario (read & write)

¿Tienen alguna pregunta?

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

Interacción con el usuario

Es usual que un programa interactúe con los usuarios. Por ejemplo, el programa "pintura_aula" podría:

- pedir los datos del ancho y largo de un aula y la cantidad de ventanas,
- y luego, mostrar en pantalla el resultado obtenido.

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

Observación: en el horario de la práctica se explicarán más detalles sobre estas primitivas (no se lo pierda) ☺

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

```
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:4:2, ' 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 4

Programación de computadoras a bajo nivel






Acme Chimtel Maple Achepe

- Cada CPU de una computadora es capaz de ejecutar un único lenguaje llamado lenguaje máquina.
- Generalmente, cada marca de CPU tiene su propio lenguaje máquina.

¿Tengo que aprender el lenguaje máquina de cada CPU?

Afortunadamente **NO** (como verá a continuación)

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

Programación de computadoras a alto nivel





Acme Chimtel Achepe

Normalmente hay un compilador para cada marca de CPU.

Compilador Pascal para Acme

Compilador Pascal para Chimtel

Compilador Pascal para Achepe

Programa en Pascal

Un compilador permite traducir el código fuente de un programa, a otro lenguaje (típicamente lenguaje de máquina) permitiendo generar **código ejecutable**.

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)1998-2014.

Programación de computadoras a alto nivel

De esta manera, un **mismo programa** puede compilarse y luego puede ejecutarse sobre **distintas computadoras**.

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

Conceptos: compilación

- Un **compilador** es un programa que traduce un programa, que está escrito en un lenguaje de programación, a otro lenguaje de programación, generando un programa equivalente.
- Este proceso de traducción se conoce como **compilación**.
- Un compilador permite traducir el código fuente de un programa, a otro lenguaje (típicamente lenguaje de máquina) permitiendo generar código ejecutable.
- El **código ejecutable** es una secuencia de código que la computadora puede ejecutar directamente al ser invocado, (sin necesidad que el compilador esté presente.) Generalmente de extensión EXE o COM.

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

Edición / Compilación / Ejecución

Existen en la actualidad aplicaciones que brindan un **entorno de programación**, donde se puede editar, compilar y ejecutar programas en Pascal.

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

Elementos predefinidos

- Los elementos predefinidos son generalmente propuestos en la definición del lenguaje y luego provistos por el compilador de dicho lenguaje.
- Así puede haber tipos predefinidos, constantes predefinidas, primitivas predefinidas, operaciones o funciones predefinidas.
- La **definición** de un lenguaje es algo **teórico**. A la definición original se lo llama **Estándar**.
- Instituciones o compañías ofrecen luego un **compilador** que respeta a la definición estándar.
- Muchas veces estos compiladores agregan elementos y **extienden** al estándar.

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

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

Ejemplo: VAR stock_monitores: INTEGER;

Una variable de tipo INTEGER puede tomar como valor cualquier número entero (negativo, positivo o cero) entre un mínimo valor y un máximo definido por el compilador usado.

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

Algunas operaciones del tipo de dato simple INTEGER

Operación	Operador	Tipo de los operandos	Tipo del resultado
suma	+	integer	integer
resta	-	integer	integer
producto	*	integer	integer
división entera	div	integer	integer
resto de la división entera	mod	integer	integer

Constante predefinida **MAXINT**: Es el máximo valor del tipo INTEGER.
 Función predefinida **SQR**: devuelve el cuadrado (square) de un entero. Ejemplo: SQR(3) = 9.

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)1998-2014.

Expresiones, operadores y operandos

- Una expresión está formada por “operadores” y “operandos” (Ej. 3 + 4)
- Los **operadores** están definidos para un **tipo** específico de **operando** y para una cantidad predeterminada de **operandos**.

¿son correctas estas expresiones?

3 + 1

3 +

+ 3 1

- 4

3 + 2 + 1

1 + 2 + 3

3 ++ 4 1

3 + 4 * 2

3 * 4 + 2

3 > 2 + 1

```

PROGRAM ejemplo;
VAR recibido, vendido, stock, sucursales,
    cant_x_suc, sobrante: INTEGER;
BEGIN
    recibido := 10; vendido := 12; sucursales:=4;
    cant_x_suc:= recibido div sucursales;
    sobrante:= recibido mod sucursales;
    stock := recibido - vendido;
END.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

Expresiones numéricas

Los operadores toman los valores de los operandos y obtienen un nuevo valor. Este valor puede ser el resultado final de la expresión, o un resultado parcial utilizado en la expresión por otro operador.

La expresión $3 + 4 * 2 - 4$ ¿en que orden se evalúa?

- El orden en que se evalúa la operación depende de la **precedencia de los operadores**.
 $3 + 4 * 2 - 4 = 7$
- Los **paréntesis** permiten cambiar el orden de evaluación
 $(3+4) * 2 - 4 = 10$
 $(3+4) * (2 - 4) = -14$

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

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 15

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.

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 16

Algunas operaciones del tipo de dato simple REAL

Es un subconjunto de los números reales en dos sentidos: (1) tiene un mínimo y un máximo; (2) tiene una “precisión” máxima. No se cumple que “entre dos números reales existe siempre otro número real”.

Operación	Operador	Tipo de los operandos	Tipo del resultado
suma	+	real	real ⁽¹⁾
resta	-	real	real ⁽¹⁾
producto	*	real	real ⁽¹⁾
división real	/	real	real

(1) El resultado es integer si los dos operandos son integer.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

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 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)1998-2014.

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 19

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

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

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 21

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?

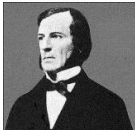
- Para indicar un **valor** de tipo CHAR, se utilizan las comillas simples. Ej: 'a', '?', '+', etc.
- Por lo tanto, en Pascal: 'A' es una letra, y en cambio A es un identificador.

Función preddefinida CHR: permite obtener un caracter cualquiera a partir de su código ASCII.
Ejemplos: chr(65) = 'A'; chr(33) = '!'.
Función preddefinida ORD: dado un caracter cualquiera, devuelve su código ASCII.
Ejemplos: ord('A') = 65, ord('!') = 33.

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

George Boole (1815-1864)

Matemático y filósofo Inglés. Inventor del "álgebra de Boole", la base de la aritmética computacional moderna. Es considerado uno de los fundadores del campo de las Ciencias de la Computación.



En 1854 publicó "An Investigation of the Laws of Thought" donde desarrolla un sistema de reglas que permite expresar, manipular y simplificar, problemas lógicos y filosóficos cuyos argumentos admiten dos estados (verdadero o falso) por procedimientos matemáticos. Se podría decir que es el padre de las operaciones lógicas, y gracias a su álgebra hoy en día podemos manipular operaciones lógicas.

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

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.
- **CHAR:** es el conjunto de los 256 caracteres del código ASCII (letras, dígitos, y símbolos)
- **BOOLEAN:** es el conjunto {true, false} (valores lógicos)

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)1998-2014.

Tipo de Dato Simple BOOLEAN

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

- Los operadores lógicos “y”, “o” y “no” se representan en Pascal con las palabras reservadas **and**, **or** y **not**.
- Puedo utilizar los operadores = > y <
- El símbolo ≠ se escribe <>
- El símbolo ≥ se escribe >= o =>

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

Expresiones numéricas y lógicas

- Al introducir la primitiva de asignación, se mostró que el lado derecho al símbolo “:=” es una **expresión** que da un valor.
- Las expresiones indican (“expresan”) como calcular adecuadamente un valor.
- Saber construir correctamente expresiones es muy importante porque:
 - se utilizan de muchas maneras en un algoritmo (no solo en asignaciones)
 - hay expresiones de muchos tipos de valores (no solo numéricos)

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

Operadores y valores

- Operadores **numéricos**: (ej. + - / *)
Toman números y tienen un número por resultado
- Operadores **relacionales**: (ej. = > < ≠ ≥ ≤)
Relacionan dos datos del mismo tipo y tienen un resultado que es **verdadero** o **falso**.
- Operadores **lógicos**: (ej. y o no)
Toman valores del conjunto { verdadero, falso } y su resultado es un valor verdadero o falso.

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

Expresiones Lógicas

Así como hay infinitos valores numéricos, hay sólo dos valores lógicos: **verdadero** y **falso**. (true – false)

Las **expresiones lógicas** se construyen con los siguientes tres **operadores básicos**:

y (and) conjunción
o (or) disyunción
no (not) negación

A diferencia de los operadores numéricos, **alcanza una tabla** para definir todos los resultados

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

Tablas de verdad

Una tabla de verdad para un operador lógico, muestra explícitamente el **resultado** de **todas** las **operaciones posibles** con ese operador, y por lo tanto define al operador.

Sea A una expresión lógica (esto es, su resultado es verdadero o falso), la tabla de verdad de la negación es la siguiente:

A	no A
verdadero	falso
falso	verdadero

Por ejemplo, A podría ser “es par” o “tengo crédito” o “está vacía” o “es cero” o “estoy en clase”, etc.

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

Tabla de Verdad para el “y” (conjunción)

A	B	A y B
verdadero	verdadero	verdadero
verdadero	falso	falso
falso	verdadero	falso
falso	falso	falso

Ejemplo: “tengo señal” y “tengo saldo”

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)1998-2014.

Tabla de Verdad para el "o" (disyunción)

A	B	A o B
verdadero	verdadero	verdadero
verdadero	falso	verdadero
falso	verdadero	verdadero
falso	falso	falso

Ejemplo: "voy caminando" o "voy en bicicleta"

Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
31

Precedencia en Expresiones Lógicas

El orden de evaluación asociado a los distintos operadores lógicos determina su precedencia.

La **precedencia en expresiones lógicas** tiene analogía con la usada en expresiones aritméticas:

(1) no	(1) menos unitario
(2) y	(2) producto
(3) o	(3) suma

no A o B y C

(no A) o (B y C)

- 3 + 4 x 5

(-3) + (4 x 5)

Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
32

Expresiones Lógicas

- Notemos que el uso de los **paréntesis** ayuda a dar **distintos significados**...

Compare:

(tomo502 o tomo503 o tomo500) y tengo_tarjeta

tomo502 o tomo503 o (tomo500 y tengo_tarjeta)

Calcule el resultado de ambas expresiones con:
 tomo502 = verdadero, tomo503=falso,
 tomo500 =falso, tengo_tarjeta=falso

Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
33

Expresiones Lógicas Equivalentes

- Decimos que dos expresiones lógicas son **equivalentes** si una es verdadera si y solo si la otra también lo es. **Ejemplo:**

(tomo502 o tomo503 o tomo500) y tengo_tarjeta

es equivalente a

(tomo502 y tengo_tarjeta) o (tomo503 y tengo_tarjeta) o (tomo500 y tengo_tarjeta)

¿Por qué es importante poder encontrar expresiones equivalentes?

Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
34

Expresiones numéricas y lógicas

Ahora si, escriba expresiones para:

- Un número N es mayor a 10
- N es mayor a 10 y menor a 100.
- N tiene a lo sumo 4 dígitos.
- N tiene 4 dígitos (exactamente).
- N tiene dos o cuatro dígitos.
- N es un número impar.
- N es divisible por 7 y divisible por 11 y tiene dos dígitos.

Hay más ejercitación vea el práctico ☺

Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
35

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
36

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)1998-2014.