



Introducción al lenguaje de programación Modula-2 y tipos de datos abstractos (TDA)

Bibliografía:

Aho, Hopcroft y Ullman. *Estructuras de datos y algoritmos*. [A]
Jensen & Partners International. *Modula-2 Owners Handbook* [JPI]
[Niklaus Wirth](#). *Programación en Modula-2* [W]

Ejercicio 0:

En la actualidad existen cientos de lenguajes de programación diferentes y día a día se implementan nuevos. Una de las tareas habituales de un Licenciado en Ciencias de la Computación es la de aprender un nuevo lenguaje de programación.

Para la realización de este práctico es conveniente que haga lo siguiente:

- (1) Lea el libro *Programación en Modula-2* [W]
- (2) Identifique las diferencias que existen entre Modula-2 y Pascal. (Ver ejercicio 7)
- (3) Revise el manual del compilador para familiarizarse con él.
- (4) Lea el manual de instalación de Modula-2.
- (5) Instale el compilador en la PC en la cual trabajará.
- (6) Pruebe el compilador con programas simples.
- (7) Analice las ayudas del compilador. ☺

Ejercicio 1:

- a) Defina tipo de datos, estructura de datos (ED) y tipo de datos abstracto (TDA). Compárelos, analice las diferencias entre ellos.
- b) ¿Qué diferencia existe entre definir e implementar un TDA? ¿Cómo se refleja esta diferencia en Modula-2? En Pascal, ¿ocurre lo mismo?
- c) ¿Qué diferencia existe entre una exportación opaca y una transparente en Modula-2? ¿Cómo se logra cada una de ellas?

Ejercicio 2:

- a) Defina una estructura de datos (ED) para manejar los legajos de los afiliados a una obra social. Los datos que deben almacenarse son:
 - ☐ Número de afiliado, nombre, apellido, domicilio, fecha de nacimiento,
 - ☐ Estado civil:
 - si es casado, debe indicar nombre y fecha de nacimiento del cónyuge, y nombre, sexo y edad de los hijos
 - si es soltero, debe indicar nombre y fecha de nacimiento de los padres
 - sino deberá indicar si es viudo, separado u otro estado.
 - ☐ Nacionalidad:
 - si es argentino, tipo y nro. de documento;
 - si es extranjero, país de origen y nro. de pasaporte.
- b)  Escriba en Modula-2 dicha estructura.



TRABAJO PRÁCTICO N° 1

ESTRUCTURAS DE DATOS Y ALGORITMOS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2008
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Ejercicio 3:

Pascal estándar solo posee tipos *integer* y *real* para el manejo de números, Modula-2 en cambio tiene predefinidos más tipos de datos numéricos (ver [JPI] pág. 108). Tener diferentes tipos numéricos resulta útil para reducir la cantidad de memoria usada, cuando hay que construir estructuras de datos grandes y el tipo base es numérico. Escriba un programa que de acuerdo a dos cotas numéricas, le indique cuál es el tipo de datos adecuado para utilizar.

Ejemplo: dadas las cotas 1 y 100 el programa sugeriría *shortcard*, si en cambio las cotas son -10 y 1.000.000, el programa debería sugerir *longint*.

Ejercicio 4:

Analice como se realiza la compatibilidad de tipos en Modula-2 (ver [W] capítulo 15 y [JPI] pág. 114). Indique si el siguiente código es correcto; en caso contrario indique las partes del código que no lo son. Si tiene dudas sobre la correctitud de este código, puede compilarlo y el compilador le dirá la verdad. ☺

```
MODULE Comp;
(* Autor      : Alejandro J. García
   Fecha       : marzo de 1994
   Utilidad    : Verificación de la compatibilidad y coerción de tipos.
   Observación: Si tiene dudas sobre la correctitud de este código,
   puede compilarlo y el compilador le dirá la verdad!
*)

CONST (*.....Constantes.....*)
Maximo = 10;
Materia = 'Estructuras de datos y algoritmos';

TYPE (*.....Tipos.....*)
Tnumero = INTEGER;
Tedad = Tnumero;
Tnatural = CARDINAL;
TCadena = ARRAY [0..20] OF CHAR;
Tarreglo1 = ARRAY [1..Maximo] OF INTEGER;
Tarreglo2 = ARRAY [1..10] OF INTEGER;
Tarreglo3 = Tarreglo1;
Tarreglo4 = ARRAY [11..20] OF INTEGER;

VAR (*.....Variables .....*)
cantidadAnios : REAL;
edad : Tedad;
anioActual, anioNacimiento : Tnumero;
anios_que_tiene : Tnatural;
numero_anios : SHORTCARD;
Frase: TCadena;
Cadena : ARRAY [1..100] OF CHAR;
Tabla1 : Tarreglo1;
Tabla2 : Tarreglo2;
Tabla3 : Tarreglo3;
Tabla4 : Tarreglo4;

(*.....Código de Comp..... *)
BEGIN
cantidadAnios := 0;
anioActual := 1994;
anioNacimiento := 1969;
cantidadAnios := anioActual - anioNacimiento;
edad := cantidadAnios;
edad := anioActual - anioNacimiento;
anios_que_tiene := edad;
```



TRABAJO PRÁCTICO N° 1

ESTRUCTURAS DE DATOS Y ALGORITMOS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2008

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



```
cantidadAnios    := 100000000000;  
numero_anios     := edad;
```

```
Frase := Materia;  
Frase := 'Hola';  
Cadena := Frase;
```

```
Tabla1 := Tabla2;  
Tabla2 := Tabla3;  
Tabla3 := Tabla1;  
Tabla1 := Tabla4;
```

```
Tabla1[1] := Tabla2[2];  
END Comp.
```

Ejercicio 5: El lenguaje de programación Modula-2 (ver [W, JPI]), no posee un tipo de datos para cadenas de caracteres (*string*), pero si tiene las herramientas necesarias para su construcción.

Defina el *tipo de datos abstracto* (TDA) cadena con las siguientes operaciones:

- Concatenar**: concatena dos cadenas, devolviendo otra cadena con el resultado,
- Subcadena**: indica si una cadena es o no subcadena de otra,
- Longitud**: indica la longitud de una cadena,
- ExtraerSubcadena**: extrae la subcadena de longitud L, que comienza en la posición P,
- SubcadenaIzquierda**: extrae la subcadena de longitud L, que comienza en la posición 1,
- SubcadenaDerecha**: extrae la subcadena de longitud L, comenzando del extremo derecho de la cadena,

Luego de definir el TDA, impleméntelo utilizando exportación transparente y exportación opaca. Compruebe, al escribir el programa de prueba, que Modula-2 permite ocultar efectivamente la implementación del TDA.

Ejercicio 6:

Escriba un programa que dados dos archivos de números enteros, verifique si hay elementos en común e indique cuáles son.

Ejercicio 7:

Analice las diferencias que encuentra entre las sentencias de Pascal estándar y Modula-2. Indique qué ventajas y desventajas presenta a su criterio cada uno de ellos.

Ejercicio 8:

- Implemente un programa que dado el nombre de un archivo de texto donde el contenido es una secuencia de números enteros separados por punto y coma, permita cargar dicha secuencia de números en un arreglo de enteros.
- Implemente un programa que dado un arreglo de números enteros, genere un archivo de texto con el contenido de dicho arreglo.
- Implemente un programa que dada una secuencia de números enteros ingresados por teclado, genere un archivo binario llamado c:\arch.dat conteniendo dicha secuencia de enteros.
- Implemente un programa que cargue en un arreglo de enteros la secuencia de números enteros almacenada en el archivo c:\arch.dat generado en el inciso anterior.



TRABAJO PRÁCTICO N° 1

ESTRUCTURAS DE DATOS Y ALGORITMOS

Licenciatura en Ciencias de la Computación - Primer cuatrimestre de 2008
Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Ejercicios adicionales ⇓

I: ?

Agregue al TDA *cadena* del ejercicio 1 las operaciones:

- Mayúscula: Dada una cadena C, devuelve C con todas sus letras en mayúscula.
- Minúscula: Dada una cadena C, devuelve C con todas sus letras en minúscula.
- Posición: Dada una cadena C y una letra L, devuelve la posición de la primera aparición de L en C y 0 en el caso que L no pertenezca a C.
- Cadena: Dado un número real N, devuelve una cadena con ese número.
- Número: Dada una cadena C, devuelve el número que ella representa.

II: ?

Defina e implemente el TDA *número fraccionario* (números de la forma A/B con A y B enteros, y $B \neq 0$) con las operaciones: suma, resta, producto y equivalencia. Debe hacer dos implementaciones del TDA: una utilizando exportación transparente y la otra utilizando exportación opaca. Escriba además un programa que permita verificar el correcto funcionamiento del TDA implementado. Compruebe, al escribir el programa de prueba, que Modula-2 permite ocultar efectivamente la implementación del TDA.

III: ?

Una fecha está compuesta de tres números: un día, un mes y un año. Defina el *tipo de datos abstracto* (TDA) *fecha* con las siguientes operaciones:

- a) **Leer_fecha**: función sin parámetros que lee una fecha y la devuelve como resultado. Observación: la función debe verificar que la fecha sea válida, considerando años bisiestos.
- b) **Mostrar_fecha**: imprime en la pantalla una fecha en el formato: <día> de <nombre_mes> de <año> (Ejemplo: 26 de julio de 1969)
- c) **Mostrar_fecha_abr**: imprime en la pantalla una fecha en el formato abreviado: DD/MM/AAAA (Ejemplo: 26/7/1969)
- d) **Igual**: dadas dos fechas F1 y F2 devuelve true si F1 es igual que F2 o false en caso contrario.
- e) **Anterior**: dadas dos fechas F1 y F2 devuelve true si F1 es anterior que F2 o false en caso contrario.
- f) **Posterior**: dadas dos fechas F1 y F2 devuelve true si F1 es posterior que F2 o false en caso contrario.

Implemente dos versiones del TDA, una con exportación transparente y otra con exportación opaca.

IV: ?

Defina e implemente el TDA *número complejo* con las operaciones: suma, resta, producto y conjugado. Debe hacer dos versiones del TDA: una utilizando exportación transparente y la otra utilizando exportación opaca. Escriba además un programa que permita verificar el correcto funcionamiento del TDA implementado.

IV: ?

Dada la estructura de datos del ejercicio 2, escriba un programa que permita visualizar los datos, y dar de alta o de baja a los afiliados de la obra social. Puede agregar al legajo otros datos que no figuren aquí si lo considera necesario.