

Práctica Nro. 4

Identificadores

Objetivo: Conocer el manejo de identificadores en memoria y como lo definen e implementan los diferentes lenguajes.

Ejercicio 1: a) Tome una de las variables de la línea 3 del siguiente código e indique y defina cuales son sus atributos:

```
1. Procedure Practica4();  
2. var  
3. a,i:integer  
4. p:puntero  
5. Begin  
6. a:=0;  
7. new(p);  
8. p:= ^i  
9. for i:=1 to 9 do  
10. a:=a+i;  
11. end;  
12. ....  
13. p:= ^a;  
14. ....  
15. dispose(p);  
16. end;
```

b) Compare los atributos de la variable del punto a) con los atributos de la variable de la línea 4. Que dato contiene esta variable?, que otra variable hay en este código?

Ejercicio 2:

- a. Indique cuales son las diferentes formas de inicializar una variable en el momento de la declaración de la misma.
- b. Analice en los lenguajes: Java, C, Python y Ruby las diferentes formas de inicialización de variables que poseen. Realice un cuadro comparativo de esta característica.

Ejercicio 3: Explique los siguientes conceptos asociados al atributo l-valor de una:

- a. Variable estática.
- b. Variable automática o semiestática.
- c. Variable dinámica.
- d. Variable semidinámica.

De al menos un ejemplo de cada uno.

Investigue sobre que tipos de variables respecto de su l-valor hay en los lenguajes C y Ada.

Ejercicio 4:

- a. ¿A qué se denomina variable local y a qué se denomina variable global?
- b. ¿Una variable local puede ser estática respecto de su l-valor? En caso afirmativo de un ejemplo
- c. Una variable global ¿siempre es estática? Justifique la respuesta.
- d. Indique que diferencia hay entre una variable estática respecto de su l-valor y una constante

Ejercicio 5:

- a. En Ada hay dos tipos de constantes, las numéricas y las comunes. Indique a que se debe dicha clasificación.

- b. En base a lo respondido en el punto **a)**, determine el momento de ligadura de las constantes del siguiente código:

```
H: constant Float:= 3,5;
I: constant:= 2;
K: constant float:= H*I;
```

Ejercicio 6: Sea el siguiente archivo con funciones de C:

```
Archivo.c
{   int x=1; (1)
    int func1();{
        int i;
        for (i:=0; i < 4; i++) x=x+1;
    }
    int func2();{
        int i, j;
        /*sentencias que contienen declaraciones y
        sentencias que no contienen declaraciones*/
        .....
        for (i:=0; i < 3; i++) j=func1 + 1;
    }
}
```

Analice si llegaría a tener el mismo comportamiento en cuanto a alocaión de memoria, sacar la declaración **(1)** y colocar dentro de **func1()** la declaración **static int x =1;**

Ejercicio 7: Sea el siguiente segmento de código escrito en Java, indique para los identificadores si son globales o locales.

```
Clase Persona {
    public long id
    public string nombreApellido
    public Domicilio domicilio
    private string dni;
    public string fechaNac;
    public static int cantTotalPersonas;

    //Se tienen los getter y setter de cada una de las variables
    //Este método calcula la edad de la persona a partir de la fecha de nacimiento

    public int getEdad(){
        public int edad=0;
        public string fN = this.getFechaNac();
        ...
        ...
        ...
        return edad;
    }
}

Clase Domicilio {
    public long id;
    public static int nro
    public string calle
    public Localidad loc;

    //Se tienen los getter y setter de cada una de las variables
}
```

Ejercicio 8: Sea el siguiente ejercicio escrito en Pascal

```
1- Program Uno;
2- type tpuntero= ^integer;
3- var mipuntero: tpuntero;
4- var i:integer;
5- var h:integer;
6- Begin
7-     i:=3;
8-     mipuntero:=nil;
9-     new(mipuntero);
10-    mipuntero^:=i;
11-    h:= mipuntero^+i;
12-    dispose(mipuntero);
13-    write(h);
14-    i:= h- mipuntero;
15- End.
```

- a) Indique el rango de instrucciones que representa el tiempo de vida de las variables i, h y mipuntero.
- b) Indique el rango de instrucciones que representa el alcance de las variables i, h y mipuntero.
- c) Indique si el programa anterior presenta un error al intentar escribir el valor de h. Justifique
- d) Indique si el programa anterior presenta un error al intentar asignar a i la resta de h con mipuntero. Justifique
- e) Determine si existe otra entidad que necesite ligar los atributos de alcance y tiempo de vida para justificar las respuestas anteriores. En ese caso indique cuál es la entidad y especifique su tiempo de vida y alcance.
- f) Especifique el tipos de variable de acuerdo a la ligadura con el l-valor de las variables que encontró en el ejercicio.

Ejercicio 9: Elija un lenguaje y escriba un ejemplo:

- a. En el cual el tiempo de vida de un identificador sea mayor que su alcance
- b. En el cual el tiempo de vida de un identificador sea menor que su alcance
- c. En el cual el tiempo de vida de un identificador sea igual que su alcance

Ejercicio 10: Si tengo la siguiente declaración al comienzo de un procedimiento:

```
int c; en C
var c:integer; en Pascal
c: integer; en ADA
```

Y ese procedimiento NO contiene definiciones de procedimientos internos. ¿Puedo asegurar que el alcance y el tiempo de vida de la variable "c" es siempre todo el procedimiento en donde se encuentra definida?. Analícelo y justifique la respuesta, para todos los casos.

Ejercicio 11: a) Responda Verdadero o Falso para cada opción. El tipo de dato de una variable es?

- I) Un string de caracteres que se usa para referenciar a la variable y operaciones que se pueden realizar sobre ella.
- II) Conjunto de valores que puede tomar y un rango de instrucciones en el que se conoce el nombre.
- III) Conjunto de valores que puede tomar y lugar de memoria asociado con la variable.
- IV) Conjunto de valores que puede tomar y conjunto de operaciones que se pueden realizar sobre esos valores.

- b) Escriba la definición correcta de tipo de dato de una variable.

Ejercicio 12: Sea el siguiente programa en ADA, completar el cuadro siguiente indicando para cada variable de que tipo es en cuanto al momento de ligadura de su l-valor, su r-valor al momento de alocaación en memoria y para todos los identificadores cuál es su alcance y cual es su el tiempo de vida.

Indicar para cada variable su r-valor al momento de alocaación en memoria

```

1. with text_io; use text_io;
2. Procedure Main is;
3.   type vector is array(integer range <>);
4.   a, n, p:integer;
5.   v1:vector(1..100);
6.   c1: constant integer:=10;
7.   Procedure Uno is;
      1. type puntero is access integer;
      2. v2:vector(0..n);
      3. c1, c2: character;
      4. p,q: puntero;
      5. begin
          7.5.1. n:=4;
          7.5.2. v2(n):= v2(1) + v1(5);
          7.5.3. p:= new puntero;
          7.5.4. q:= p;
          7.5.5. ....
          7.5.6. free p;
          7.5.7. ....
          7.5.8. free q;
          7.5.9. ....
      7.6. end;
8. begin
9.   n:=5;
10.  ....
11.  Uno;
12.  a:= n + 2;
13.  ....
14. end

```

Ident.	l-valor	r-valor	Alcance	T.V.
a (línea 4)	automatica	basura	4-14	1-14

Conceptos y Paradigmas de lenguajes de Programación 2016

Aclaración:

Ident.= Identificador

T.V. = Tiempo de Vida

r-valor debe ser tomado al momento de la alocaión en memoria

El *alcance de los identificadores* debe indicarse desde la línea siguiente a su declaración.

Ejercicio 13: El nombre de una variable puede condicionar:

- a) Su tiempo de vida.
- b) Su alcance.
- c) Su r-valor.
- d) Su tipo.

Justifique la respuesta

Ejercicio 14: Sean los siguientes archivos en C, los cuales se compilan juntos

Indicar para cada variable de que tipo es en cuanto al momento de ligadura de su l-valor.

Indicar para cada identificador cuál es su alcance y cual es su el tiempo de vida.

Indicar para cada variable su r-valor al momento de alocaión en memoria

ARCHIVO1.C

```
1.  int v1;
2.  int *a;
3.  Int fun2 ()
4.  { int v1, y;
5.    for(y=0; y<8; y++)
6.    { extern int v2;
7.      ...}
8.  }
9.  main()
10. {static int var3;
11.  extern int v2;
12.  int v1, y;
13.  for(y=0; y<10; y++)
14.  { char var1='C';
15.    a=&v1;}
16. }
```

ARCHIVO2.C

```
17. static int aux;
18. int v2;
19. static int fun2( )
20. { extern int v1;
21.   aux=aux+1;
22.   ...
23. }
24. int fun3( )
25. { int aux;
26.   aux=aux+1;
27.   ...
28. }
```

Ident.	l-valor	r-valor	Alcance	T.V.

Aclaración:

Ident.= Identificador

T.V. = Tiempo de Vida

r-valor debe ser tomado al momento de la aloca3ión en memoria

El *alcance de los identificadores* debe indicarse desde la línea siguiente a su declaraci3n.