Sintaxis de PSeInt
Tutorial

Introducción a la Programación (T.U.M - T.U.G. - T.U.E. - T.U.T. - Prof)
Introducción a la Computación (T.U.R. - T.U.W.)
Fundamentos de la Informática (Ing. en Minas - Ing. Electr.)

Área de Servicios
Departamento de Informática
Universidad Nacional de San Luis

Índice

INTRODUCCION

PSeInt es un software que interpreta pseudocódigo. Pseudocódigo es un tipo de lenguaje de diseño que permite expresar algoritmos acercándose a los lenguajes de programación con elementos del lenguaje de problema.

El objetivo de este manual es brindar al alumno una guía rápida de diferencias y similitudes entre la sintaxis de Lenguaje de Diseño vista en clase y la sintaxis usada por el software PSeInt.

Entre las características generales del pseudocódigo se encuentra una sintaxis sencilla y un manejo de estructuras básicas de control, entre ellas: secuencial, condicional y repetición.

1. Sintaxis general

1.1. Formalización de Algoritmos

Lenguaje de Diseño	**PSeInt**
ALGORITMO ”Nombre de Algoritmo” COMENZAR Declaraciones de variables Acciones FIN	PROCESO SinTitulo Acción 1; ... Acción n FINPROCESO

1.2. Tipos de datos

Lenguaje de Diseño	**PSeInt**
Numéricos: enteros y reales. **Lógico:** solo puede tomar dos valores Verdadero o Falso. **Caracter** Un carácter es una letra, un número o un signo de puntuación, encerrado entre comillas simples.	**Numéricos:** enteros y reales, los reales se separan con un punto. **Lógico:** solo puede tomar dos valores Verdadero o Falso. **Caracter** Un carácter es una letra, un número o un signo de puntuación, encerrado entre comillas simples o dobles.

1.3. Declaración de variable

Lenguaje de Diseño	**PSeInt**
<variable>: [Real/Entero/Logico/Caracter] *Ejemplo:* Contador : Entero X, Y : Real	DEFINIR <variable >COMO [Real/Entero/Logico/Caracter]; *Ejemplo:* DEFINIR Contador COMO Entero; DEFINIR X, Y COMO Real;

Importante: En PseInt los tipos de datos simples pueden determinarse automáticamente no solo en el momento de creación de la variable sino en el momento en que se referencia dicha variable. A pesar de esta particularidad que ofrece PseInt, los ejercicios deben realizarse definiendo el tipo de dato simple en el momento de creación de la variable.

1.4. Nombre de una variable

Para definir una variable:

1. Debe comenzar con una letra (A a Z, mayúsculas o minúsculas) y no deben contener espacios en blanco ni operadores.
2. Después del primer caracter se permiten: letras, dígitos y el guión bajo (_).
3. La longitud de identificadores puede ser de varios caracteres. Pero es recomendable una longitud promedio de 8 caracteres.
4. El nombre de la variable debe *dar una idea del valor que contiene*.

2. Expresiones

2.1. Expresiones relacionales

Expresión	**Lenguaje de Diseño**	**PSeInt**
Mayor	$>$	$>$
Menor	$<$	$<$
Igual	=	=
Menor o Igual	$<=$	$<=$
Mayor o Igual	$>=$	$>=$
Distinto	$><$	$><$

2.2. Expresiones lógicas

Expresión	**Lenguaje de Diseño**	**PSeInt**
Conjunción (y)	$\wedge$	& ó Y
Disyunción (o)	$\vee$	| ó O
Negación (no)	$\neg$	$\sim$ ó NO

2.3. Expresiones aritméticas

Expresión	**Lenguaje de Diseño**	**PSeInt**
Suma	+	+
Resta	-	-
Multiplicación	*	*
División	/	/
Potenciación	$\uparrow$	^
Módulo (Resto de la división entera)	//	% ó MOD

3. Entrada Salida

3.1. Entrada - Lectura

Lenguaje de Diseño	**PSeInt**
LEER $<$variable $>$	LEER $<$variable $>$;
LEER $<$vble1 $>, \ldots, <$vbleN $>$	LEER $<$vble1 $>, \ldots, <$vbleN $>$;

3.2. Salida - Escritura

Lenguaje de Diseño	PSeInt
ESCRIBIR ”Mensaje”	ESCRIBIR ”Mensaje”;
ESCRIBIR <vble1 >, . . . , <vbleN >	ESCRIBIR <vble1 >, . . . , <vbleN >;
ESCRIBIR”El valor de la variable es:”,<variable>	ESCRIBIR”El valor de la variable es:”<variable>;

4. Estructuras de Control

4.1. Secuencial

En PseInt, igual que en lenguaje de diseño las acciones se escriben una debajo de otra, y pueden ir separadas o no por un punto y coma.

4.1.1. Asignación

Lenguaje de Diseño	PSeInt
<variable><- <expresión >	<variable><- <expresión >;
Ejemplo:	*Ejemplo:*
A <- 156;	A <- 156;
B<- ’X’;	B<- ’X’;
C<- 157- A;	C<- 157- A;

4.2. Condicional

Lenguaje de Diseño	PSeInt
SI <condición >ENTONCES	SI <condición >ENTONCES
<instrucciones >	<instrucciones >
SINO	SINO
<instrucciones >	<instrucciones >
FINSI	FINSI
Ejemplo:	*Ejemplo:*
SI X >Y ENTONCES	SI X >Y ENTONCES
Escribir X	Escribir X;
SINO	SINO
Escribir Z	Escribir Z;
FINSI	FINSI

4.3. Repeticion

4.3.1. Mientras

Lenguaje de Diseño	PSeInt
MIENTRAS <condición >HACER <instrucciones > FINMIENTRAS *Ejemplo:* MIENTRAS X >1 HACER X <- X-1; ESCRIBIR X; FINMIENTRAS	MIENTRAS <condición >HACER <instrucciones > FINMIENTRAS *Ejemplo:* MIENTRAS X >1 HACER X <- X-1; ESCRIBIR X; FINMIENTRAS

4.3.2. Para

Lenguaje de Diseño	PSeInt
PARA <variable >DESDE <inicial >HASTA <final >CON PASO <paso >HACER <instrucciones > FINPARA *Ejemplo:* PARA i DESDE 1 HASTA 5 CON PASO 1 HACER ESCRIBIR i; FINPARA	PARA <variable ><- <inicial >HASTA <final >CON PASO <paso >HACER <instrucciones > FINPARA *Ejemplo:* PARA i<- 1 HASTA 5 CON PASO 1 HACER ESCRIBIR i; FINPARA

5. Arreglos

5.0.1. Definición de Arreglo

Lenguaje de Diseño	PSeInt
<identificador >: arreglo [1 ... <máximo >] de <tipo >	Definir <identificador >Como <tipo; >Dimension <identificador >[<max1 >, ... , <maxN >];
Ejemplo: A: arreglo [1...10] de Real	*Ejemplo:* Definir A Como Real; Dimension A [10];

Se pueden declarar mas de un arreglo en una misma instrucción separándolos con una coma.

Para declarar el tipo de elemento del arreglo, se puede definir una variable de tipo específico y luego utilizarla para asignale valores. O se puede asignarle directamente por medio de la instrucción Leer, y el tipo del arreglo va a depender de lo ingresado.

Ejemplo:
Definir A como Entero;
Definir B como Caracter;
Dimension A[5];
Dimension B[5];
Definir x como Caracter;

```
Leer A[1]; // Suponer que se ingresa un 8
Escribir A[1];// Muestra por pantalla 8
x <- 'p';
B[1] <- x; //B[1] tiene el caracter 'p'
B[2] <- 58; // ERROR DE TIPO porque tenía asignado un valor de tipo caracter y 58 es un valor entero
```

Para acceder a un elemento del arreglo se utiliza el siguiente comando:

```
<identificador >[posición];
Ejemplo:
Escribir A[3]; //Muestra por pantalla el contenido del arreglo A en la posición 3
i <- 3;
Escribir A[i]; //Muestra por pantalla el contenido del arreglo A en la posición 3
```

6. Subalgoritmos

6.1. Definición de Subalgoritmo

Lenguaje de Diseño	PSeInt
SUBALGORITMO "Nombre"(<clase parámetro> <Nombre parámetro>:<Tipo>,...) COMENZAR <Secuencia de acciones> FIN	SUBPROCESO <variable tipo Out><- <Nombre función>(<Nombre parámetro 1>, <Nombre parámetro 2>,...) acción 1; ... acción N; FINSUBPROCESO

En PseInt un Subalgoritmo comienza con la palabra clave Subproceso, seguida de la *Variable de tipo Out*, el signo de asignación, el *Nombre del Subproceso*, y finalmente, la *Lista de argumentos* (o Parámetros).
Existen variantes para esta estructura, si el Subproceso no devuelve ningún valor, pueden omitirse el identificador *<variable tipo Out>* y el signo de asignación, es decir, colocar directamente el nombre y los parámetros a continuación de la palabra clave Subproceso. Si el Subproceso no recibe ningún valor pueden colocarse los parétesis vacíos u omitirse, finalmente la primer línea con el nombre del Subproceso.
Además, opcionalmente pueden agregarse las palabras claves Por Valor o Por Referencia para indicar el tipo de pasaje en cada argumento, en donde un argumento Por Valor es equivalente a un argumento de tipo In en lenguaje de diseño y un argumento Por Referencia es equivalente a un argumento de tipo In Out en lenguaje de diseño. Si no se indica, los arreglos se pasan Por Referencia (In Out),y el resto de los argumentos Por Valor (In).

6.2. Invocación de Subalgoritmo

Igual que en lenguaje de diseño, para invocar a un Subproceso se debe utilizar su Nombre (Parámetros). Debe respetarse la cantidad de Parámetros formales, que debe coincidir con la cantidad de parámetros actuales. Además, deben coincidir los tipos utilizados en la invocación con los tipos declarados en la definición de Subproceso.
Cada vez que un Subproceso es invocado desde un Algortimo o Subalgoritmo (Proceso o Subproceso) se establece, automáticamente una correspondencia entre los parámetros formales y los actuales. Esta correspondencia está definida por la posición que los parámetros ocupan dentro de la lista de parámetros.
Para los parámetros formales que fueron definidos como:
In/ Por Valor los parámetros actuales pueden ser constantes, variables (definidas en el ambiente del Algoritmo invocante), expresiones o valores de funciones.
Out o In Out/ Por Referencia, los parámetros actuales deben ser variables definidas en el ambiente del algoritmo invocante, pues allí el subalgoritmo devuelve sus resultados.

Ejemplo: El siguiente programa calcula el promedio de una lista de N datos utilizando un SubProceso o Subalgoritmo.

6.2.1. Lenguaje de Diseño:

```
SUBALGORTIMO ”Ingresar”( IN OUT arr : entero, IN tam: entero, OUT cant: entero)
COMENZAR
i ,n, a: entero
Escribir ”Ingrese la cantidad de datos:”
LEER n
MIENTRAS (n < 0 v n > tam) HACER
ESCRIBIR ”Ingrese nuevamente la cantidad de datos”
LEER n
FINMIENTRAS
PARA i DESDE 1 HASTA n CON PASO 1 HACER
ESCRIBIR ”Ingrese el dato ”
LEER a
arr[i] <- a
FINPARA
cant<-n
FIN
SUBALGORITMO ”Promedio” ( IN arreglo: entero , IN cantidad: entero, OUT prom: real )
suma: entero
suma <- 0
PARA i DESDE 1 HASTA cantidad CON PASO 1 HACER
suma <- suma + arreglo[i]
FINPARA
prom <- suma/cantidad
FIN
ALGORITMO ”Principal”
datos: arreglo [1..100] de entero
num :entero
prom: real
Ingresar(datos,100, num)
ESCRIBIR ” La cantidad de datos ingresados es”: num
ESCRIBIR ”El promedio es: ”, Promedio(datos ,num, prom)
FIN
```

6.2.2. PSeInt

```
SubProceso cant <-Ingresar( arr ,tam)
Definir i, n, a como entero
Escribir ”Ingrese la cantidad de datos:”
Leer n
Mientras (n < 0|n > tam)
Escribir ”Ingrese nuevamente la cantidad de datos”
Leer n
FinMientras
Para i<-1 Hasta n con paso 1 Hacer
Escribir ”Ingrese el dato ”
Leer a
arr[i]<- a
Finpara
cant <- n
FinSubProceso
SubProceso prom <- Promedio ( arreglo , cantidad )
Definir suma como entero
suma <- 0
Para i<-1 Hasta cantidad con paso 1 Hacer
suma <- suma + arreglo[i]
FinPara
prom <- suma/cantidad
FinSubProceso
Proceso Principal
Definir datos como entero
Dimension datos[100]
Definir num como entero
num <-Ingresar(datos,100)
Escribir ”La cantidad de datos ingresados es ”: num
Escribir ”El promedio es: ”, Promedio(datos,num)
FinProceso
```

6.3. Funciones primitivas o predefinidas

Lenguaje de Diseño	**PSeInt**
ABS(X) Valor Absoluto	ABS(X) Valor Absoluto
RC(X) Raíz Cuadrada de X	RC(X) o RAIZ(X) Raíz Cuadrada de X

7. Resumen

7.1. Tabla Comparativa

Operación	**Lenguaje de Diseño**	**PSeInt**
Declaración	<variable>: [Real/Entero/Logico/Caracter]	DEFINIR<variable>COMO [Real/Entero/Logico/Caracter] ;
Asignación	<variable><- <expresión >	<variable><- <expresión >;
Expresión Relacional		
Mayor	>	>
Menor	<	<
Igual	=	=
Menor o Igual	<=	<=
Mayor o Igual	>=	>=
Distinto	><	><
Expresión Lógica		
Conjunción (y)	$\wedge$	& ó Y
Disyunción (o)	$\vee$	| ó O
Negación (no)	$\neg$	$\sim$ ó NO
Expresión aritmética		
Suma	+	+
Resta	-	-
Multiplicación	*	*
División	/	/
Potenciación	$\uparrow$	^
Módulo (Resto de la división entera)	//	% ó MOD
Entrada-Lectura	LEER <variable > LEER <vble1 >, . . . , <vbleN >	LEER <variable >; LEER <vble1 >, . . . , <vbleN >;
Salida-Escritura	ESCRIBIR ”Mensaje” ESCRIBIR <vble1 >, . . . , <vbleN > ESCRIBIR”El valor de la variable es:”,<variable>	ESCRIBIR ”Mensaje”; ESCRIBIR <vble1 >, . . . , <vbleN >; ESCRIBIR”El valor de la variable es:”<variable>;
Arreglo	<identificador >: arreglo [1 ... <máximo >] de <tipo >	Definir <identificador >Como <tipo; >Dimension <identificador >[<max1>, ... ,<maxN >];