

PSeint

PSeint es principalmente un intérprete de pseudocódigo.

El proyecto nació como trabajo final para la cátedra de *Programación I* de la carrera *Ingeniería en Informática* de la *Universidad Nacional del Litoral*, razón por la cual el tipo de pseudocódigo que interpreta está basado en el pseudocódigo presentado en la cátedra de *Fundamentos de Programación* de dicha carrera.

PSeint

Actualmente incluye otras funcionalidades como editor y ayuda integrada, generación de diagramas de flujo o exportación a código C++ (en etapa experimental).

El proyecto se distribuye como software libre bajo licencia GPL.

Para descargarlo o conseguir actualizaciones visite **<http://pseint.sourceforge.net>**

Forma general de un algoritmo en Pseudocódigo

Todo algoritmo en pseudocódigo de Pseint tiene la siguiente estructura general:

Proceso Titulo

 accion 1;

 accion 2;

 .

 .

 .

 accion n;

FinProceso

Forma general de un algoritmo en Pseudocódigo

Comienza con la palabra clave **Proceso** seguida del nombre del programa, luego le sigue una secuencia de instrucciones y finaliza con la palabra **FinProceso**.

Una secuencia de instrucciones es una lista de una o más instrucciones, cada una terminada en punto y coma.

Forma general de un algoritmo en Pseudocódigo

Las acciones incluyen operaciones de entrada y salida, asignaciones de variables, condicionales si-entonces o de selección múltiple y/o lazos mientras, repetir o para.

Tipos de datos

- Tipos Simples: Numérico, Lógico, Carácter.
- Estructuras de Datos: Arreglos.

Los identificadores, o nombres de variables, deben constar sólo de letras, números y/o guión_bajo (_), comenzando siempre con una letra.

Tipos de datos Simples

Existen tres tipos de datos básicos:

- *Numérico*: números, tanto enteros como decimales. Para separar decimales se utiliza el punto.

Ejemplos: 12 23 0 -2.3 3.14

- *Lógico*: solo puede tomar dos valores: VERDADERO o FALSO.

- *Carácter*: caracteres o cadenas de caracteres encerrados entre comillas (pueden ser dobles o simples). Ejemplos 'hola' "hola mundo" '123' 'FALSO' 'etc'

Tipos de datos Simples

Los tipos de datos simples se determinan automáticamente cuando se crean las variables. Las dos acciones que pueden crear una variable son la lectura(LEER) y la asignación(<-).

Por ejemplo, la asignación "A<-0;" está indicando implícitamente que la variable A será una variable numérica.

Una vez determinado el tipo de dato, deberá permanecer constante durante toda la ejecución del proceso; en caso contrario el proceso será interrumpido.

Operadores

Operador Relacionales

>	Mayor que
<	Menor que
=	Igual que
<=	Menor o igual que
>=	Mayor o igual que
<>	Distinto que

Ejemplo

3>2
'ABC'<'abc'
4=3
'a'<='b'
4>=5
var1<>var2

Lógicos

& ó Y	Conjunción (y).
ó O	Disyunción (o).
~ ó NO	Negación (no).

(7>4) & (2=1) //falso
(1=1 | 2=1) //verdadero
~(2<5) //falso

Algebraicos

+	Suma
-	Resta
*	Multiplicación
/	División
^	Potenciación
% ó MOD	Módulo (resto de la división entera)

total <- cant1 + cant2
stock <- disp – venta
area<- base * altura
porc<- 100 * parte / total
sup<- 3.41 * radio ^ 2
resto <- num MOD di

Funciones Matemáticas

Función

RC(X)
ABS(X)
LN(X)
EXP(X)
SEN(X)
COS(X)
TAN(X)
TRUNC(X)
REDON(X)
AZAR(X)

Significado

Raíz Cuadrada de X
Valor Absoluto de X
Logaritmo Natural de X
Función Exponencial de X
Seno de X
Coseno de X
Tangente de X
Parte entera de X
Entero más cercano a X
Entero aleatorio entre 0 y x-1

La función raíz cuadrada no debe recibir un argumento negativo.

La función exponencial no debe recibir un argumento menor o igual a cero.

Primitivas Secuenciales

(Comandos de Entrada, Proceso y Salida)

- Escritura (Salida).
- Lectura (Entrada).
- Asignación (Proceso).

Escritura o salida

La instrucción Escribir permite mostrar valores al ambiente.

Escribir <expr1> ,<expr2> , ... , <exprN> ;

Esta instrucción imprime al entorno (en este caso en la pantalla) los valores obtenidos de evaluar N expresiones. Dado que puede incluir una o más expresiones, mostrará uno o más valores.

Asignación o proceso

La instrucción de asignación permite almacenar un valor en una variable.

`<variable><- <expresión> ;`

Al ejecutarse la asignación, primero se evalúa la expresión de la derecha y luego se asigna el resultado a la variable de la izquierda.

El tipo de la variable y el de la expresión deben coincidir.

Lectura o entrada

La instrucción Leer permite ingresar información desde el entorno.

Leer <variable1> ,<variable2> , ... , <variableN> ;

Esta instrucción lee N valores desde el entorno (en este caso el teclado) y los asigna a las N variables mencionadas. Pueden incluirse una o más variables, por lo tanto el comando leerá uno o más valores.

Condicionales

Si-Entonces (If-Then)

La secuencia de instrucciones ejecutadas por la instrucción Si-Entonces-Sino depende del valor de una condición lógica.

Si <condición> Entonces

 <instrucciones>

Sino

 <instrucciones>

FinSi

Selección Múltiple (SelectIf)

La secuencia de instrucciones ejecutada por una instrucción *Según* depende del valor de una variable numérica.

```
Segun<variable> Hacer  
    <número1>: <instrucciones>  
    <número2>,<número3>: <instrucciones>  
    <...>  
    De Otro Modo: <instrucciones>  
FinSegun
```


Repetitivas

Mientras Hacer (while)

La instrucción *Mientras* ejecuta una secuencia de instrucciones mientras una condición sea verdadera.

```
Mientras <condición> Hacer  
    <instrucciones>  
FinMientras
```

Repetitivas

Repetir Hasta Que (do-while)

La instrucción *Repetir-Hasta Que* ejecuta una secuencia de instrucciones hasta que la condición sea verdadera.

Repetir

<instrucciones>

Hasta Que <condición>

Repetitivas

Para (for)

La instrucción *Para* ejecuta una secuencia de instrucciones un número determinado de veces.

```
Para <variable> <- <inicial> Hasta <final>( Con Paso <paso> ) Hacer  
    <instrucciones>  
FinPara
```

Repetitivas

Para (for)

La instrucción *Para* ejecuta una secuencia de instrucciones un número determinado de veces.

```
Para <variable> <- <inicial> Hasta <final>( Con Paso <paso> ) Hacer  
    <instrucciones>  
FinPara
```

Observaciones

- Se pueden introducir comentarios luego de una instrucción, o en líneas separadas, mediante el uso de la doble barra (//). Todo lo que precede a //, hasta el fin de la línea, no será tomado en cuenta al interpretar el algoritmo. No es válido introducir comentario con /* y */.
- No puede haber instrucciones fuera del proceso (antes de PROCESO, o después de FINPROCESO), aunque si comentarios.

Observaciones

- Las estructuras no secuenciales pueden anidarse. Es decir, pueden contener otras adentro, pero la estructura contenida debe comenzar y finalizar dentro de la contenedora.
- Los identificadores, o nombres de variables, deben constar sólo de letras, números y/o guión bajo (_), comenzando siempre con una letra.

Observaciones

- Los tipos de datos de las variables no se declaran explícitamente, sino que se infieren a partir de su utilización.
- Las constantes de tipo carácter se escriben entre comillas (").
- En las constantes numéricas, el punto (.) es el separador decimal.

Observaciones

- Las constantes lógicas son Verdadero y Falso.
- Actualmente este pseudolenguaje no contempla la creación de nuevas funciones o subprocesos.