Pseudocódigo

Introducción

La importancia de sistematizar procesos y crear programas de cómputo radica esencialmente en que estos se puedan utilizar como resolución de problemas similares en muchos casos, dicho de otra forma: la resolución del problema por medio de un programa informático debe funcionar para el uso de distintas variables y en diferentes lenguajes de cómputo. Esa es principalmente la función de un pseudocódigo.

La resolución de problemas es una tarea únicamente humana comprobable en todos los casos con los mismos resultados.

Definición

El pseudocódigo (o falso lenguaje) es comúnmente utilizado por los programadores para omitir secciones de código o para dar una explicación del paradigma que tomó el mismo programador para hacer sus códigos, esto quiere decir que el pseudocódigo no es programable sino que facilita la programación.

El principal objetivo del pseudocódigo es el de representar la solución a un algoritmo de la forma más detallada posible, y a su vez lo más parecida posible al lenguaje que posteriormente se utilizará para la codificación del mismo.

El pseudocódigo utiliza para representar las acciones, sucesivas palabras reservadas en inglés (similares a sus homónimos en los lenguajes de programación), tales como start, begin, end, stop, if-then-else, while, repeat-until, etc.

Es un lenguaje de especificación de algoritmos. El uso de tal lenguaje hace el paso de codificación final (esto es, la traducción a un lenguaje de programación) relativamente fácil.

El pseudocódigo nació como un lenguaje similar al inglés y era un medio para representar básicamente las estructuras de control de programación estructurada.

Se considera un primer borrador, dado que el pseudocódigo tiene que traducirse posteriormente a un lenguaje de programación. Cabe señalar que el pseudocódigo no puede ser ejecutado por una computadora.

Dicho de otra manera, considerado como un lenguaje falso, el pseudocódigo es un lenguaje intermedio entre nuestro lenguaje y el de programación, debido a que quien lo utiliza se guía por una serie de normas, pero sin llegar a usar una estructura tan rígida como la del lenguaje de programación.

El objetivo al que apunta es que quien lo pone en práctica se centre más en la solución del algoritmo o el diseño de un software que en el programa que utiliza para crearlo. Y esto es posible porque es más fácil de manipular ya que no tiene que tener en mente el lenguaje en sí y además, más fácil de codificar.

Por ejemplo, si alguien tiene que hacer un software con un fin determinado, utiliza un pseudocódigo propio en donde confluyen frases del lenguaje coloquial y algunas palabras de programación, y una vez que se logra concretar el software, se puede pasar al escalón siguiente que es el de la transformación al lenguaje de programación formal que se vaya a utilizar.

De esta manera, al ser un lenguaje intermedio, no tiene una composición estandarizada por lo que no todos los programadores utilizan la misma sintaxis con exactitud. Pero a la vez, como es una herramienta que está un paso previo al lenguaje formal de programación, es fácil de transformar al que será ejecutado en la computadora.

Reglas de construcción

Características y Estructuras

Las principales características de este “lenguaje” son:

- Se puede ejecutar en un ordenador una vez traducido a un lenguaje de programación
- Es una forma de representación sencilla de utilizar y de manipular.
- Facilita el paso del programa al lenguaje de programación.
- Es independiente del lenguaje de programación que se vaya a utilizar.
- Es un método que facilita la programación y solución al algoritmo del programa.

En cuanto a las estructuras, contamos con las siguientes:

- Cabecera.
- Programa.
- Modulo.
- Tipos de datos.
- Constantes.
- Variables.
- Cuerpo.
- Inicio.
- Instrucciones.
- Fin.

Tipos de Variables

Cuando programemos, necesitamos un lugar donde guardar los datos con los que operamos, ya sean números, texto, etc.

Estos datos pueden variar o no durante el proceso de ejecución del algoritmo, según esa variación, pueden ser:

- **Variables:** es un objeto en el que su contenido puede variar durante el proceso de ejecución del algoritmo, se identifican por un nombre y su tipo de valores que puede contener para realizar un uso correcto. Por ejemplo: acumulación de una suma, realizar otras operaciones matemáticas, etc.

- **Constantes:** es un objeto que permanece sin cambios durante todo el desarrollo del algoritmo. Por ejemplo: el número π (PI), el IVA, etc. Para distinguirlos de las variables, podemos ponerle el nombre en mayúsculas, esto es simplemente un consejo.

Vamos a ver un simple ejemplo donde manipularemos tres variables:

```
1    Inicio
2        A = 5
3        B = 3
4        C = A+B
5    Fin
```

Si mostráramos el resultado de **C** seria 8, ya que sumamos el valor de A que vale 5 y el valor de B que vale 3.

Una **variable** también puede guardar, por ejemplo, ***cadenas de texto*** (que es una secuencia de caracteres y se indican encerrando el texto entre comillas), ***valores booleanos*** (es un tipo de dato que solo puede almacenar dos valores, **TRUE** -verdadero - o **FALSE** - falso -**)** o un valor que el usuario teclee por teclado.

Algo muy recomendable a la hora de programar es llamar a las **variables** por un nombre significativo sobre lo que contenga para mejorar su legibilidad. Por ejemplo, si queremos almacenar un día del mes, esa variable se puede llamar **dia** o si queremos almacenar el total de un producto, podemos usar **precio_final.**

Es recomendable no dejar espacios en blanco e inicializar las variables que no tiene que ser introducidas por el usuario (en general con 0 si son numéricas, o en blanco, si son de texto, por ejemplo)

Listado de palabras reservadas y su significado

Instrucción	Significado
algoritmo nombre	Marca el comienzo de un algoritmo y le adjudica un nombre
Inicio	Marca el comienzo de un bloque de instrucciones
fin	Marca el final de un bloque de instrucciones
variables	
nombre_var es tipo_de_datos	Declaración de variables. Indica el identificador y el tipo de las variables que se van a usar en el algoritmo
constantes	
nombre_const = expresión	Declaración de constantes. La expresión se evalúa y su resultado se asigna a la constante. Este valor no puede modificarse a lo largo del programa.
leer (variable)	Entrada de datos. El programa lee un dato desde un dispositivo de entrada (si no se indica otra cosa, es por el teclado), asignando ese dato a la variable
escribir (variable)	Salida de datos. Sirve para que el programa escriba un dato en un dispositivo de salida (si no se indica otra cosa, es por pantalla).
variable = expresión	Asignación. La expresión se evalúa y su resultado es asignado a la variable
si (condición) entonces inicio acciones-1 fin si_no inicio acciones-2 fin	Instrucción condicional doble. El ordenador evaluará la condición, que debe ser una expresión lógica. Si es verdadera, realiza las acciones-1, y, si es falsa, las acciones-2.Instrucción condicional simple. Es igual pero carece de la rama “si_no”, de modo que, si la expresión de falsa, no se realiza ninguna acción y la ejecución continúa por la siguiente instrucción
según (expresión) hacer inicio valor1: acciones-1 valor2: acciones-2 … valor3: acciones-N si_no: acciones-si_no fin	Instrucción condicional múltiple. Se utiliza cuando hay más de dos condiciones posibles (verdadero o falso) . Se evalúa la expresión, que suele ser de tipo entero, y se busca un valor en la lista valor1, valor2,… valorN que coincida con ella, realizándose las acciones asociadas al valor coincidente. Si ningún valor de la lista coincide con la expresión del “según”, se realizan las acciones de la parte “si_no”.
mientras (condición) hacer inicio acciones fin	Bucle mientras. Las acciones se repiten en tanto la condición, que debe ser una expresión lógica, sea verdadera. La condición se evalúa antes de entrar al bloque de acciones, de modo que pueden no ejecutarse ninguna vez.

Instrucción	Significado
repetir inicio acciones fin mientras que (condición)	Bucle repetir. Las acciones se repiten en tanto que la condición, que debe ser una expresión lógica, sea verdadera. Se parece mucho al anterior, pero la condición se evalúa al final del bucle, por lo que éste se ejecuta, como mínimo, una vez
para variable desde expr-ini hasta expr-fin hacer inicio acciones fin	Bucle para. Se evalúa la expresión expr-ini, que debe ser de tipo entero, y se asigna ese valor a la variable. Dicha variable se incrementa en una unidad en cada repetición de las acciones. Las acciones se repiten hasta que la variable alcanza el valor expr-fin.

Ejemplo

Se requiere preguntar dos valores, y a continuación ofrecer un menú con las operaciones básicas (+, -, *, /). Después de presentar el resultado se ofrecerá la posibilidad de una nueva operación.

Declaración de variables:
Real : X, Y, RESPUESTA
Entero: OPCION
Carácter: OP

Inicio
Repetir
escribir('Primer valor : ')
leer(X)
escribir('Segundo valor : ')
leer(Y)
escribir('1) Suma ')
escribir('2) Resta ')
escribir('3) Multiplicación ')
escribir('4) División ')
escribir('Qué operación deseas realizar ? : ')
leer(OPCION)
casos OPCION de
1 : RESULTADO X+Y
2 : RESULTADO X-Y
3 : RESULTADO X*Y
4 : si Y=0 entonces escribir(' Error ') RESULTADO 0
en caso contrario RESULTADO X/Y escribir ('Resultado : ',RESULTADO)
escribir('Deseas otro cálculo : [S/N] ')
leer(OP) Hasta que RES = 'N'
Fin

Ejercicios

En los siguientes casos, escribir el pseudocódigo que resuelva los siguientes problemas:

1. Que imprima “Hola Mundo”.
2. Que imprima los primeros 10 números enteros.
3. Que pida un nombre y que después lo escriba.
4. Que pida nombre, edad, fecha de nacimiento, y estado civil y que después lo escriba.
5. Que imprima los números pares menores a 20.
6. Hacer un pseudocódigo que admita un número y decida si es par o impar.
7. Hacer un pseudocódigo para calcular el promedio de tres números.
8. Hacer un pseudocódigo para admitir dos números, decidir cual es el mayor y escribirlo por pantalla.
9. Hacer un pseudocódigo que al ingresar un número del 1 al 12 indique a que mes del calendario corresponde.
10. Hacer un pseudocódigo que reciba la fecha de nacimiento y te de su signo del Zodiaco.