



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura:

Fundamentos de programación.

Grupo: 3

No de Práctica(s): 5

Integrante(s): Ortiz Garcia Cesar Alan.

*No. de Equipo de cómputo
empleado:* 24

No. de Lista o Brigada: 9070

Semestre: 2020-1

Fecha de entrega: Lunes 16 de Septiembre del 2019

Observaciones: Tu primera actividad está bien, pero a la segunda no le encontré ningún sentido lógico. Tanto el día de la práctica como el martes 17 expliqué esa actividad, si tienes dudas no dudes en preguntarlas

Calificación **6**

Pseudocódigo:

En ciencias de la computación, y análisis numérico, el **pseudocódigo** (o lenguaje de descripción algorítmico) es una descripción de alto nivel compacta e informal¹ del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de lenguaje de programación real,² pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación. Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas subrutinas. El lenguaje de programación se complementa, donde sea conveniente, con descripciones detalladas en lenguaje natural, o con notación matemática compacta. Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo.

Aplicaciones

Generalmente se utiliza pseudocódigo en los libros de texto y publicaciones científicas relacionadas con la informática y la computación numérica, para la descripción de algoritmos, de manera que todos los programadores puedan entenderlo, aunque no todos conozcan el mismo lenguaje de programación. Generalmente, en los libros de texto se adjunta una explicación que acompaña a la introducción y que explica las convenciones particulares en uso. El nivel de detalle del pseudocódigo puede, en algunos casos, acercarse a la de formalizar los idiomas de propósito general.

Un programador que tiene que aplicar un algoritmo específico, sobre todo uno desfasado, generalmente comienza con una descripción en pseudocódigo, y luego "traduce" esa descripción en el lenguaje de programación meta y lo modifica para que interactúe correctamente con el resto del programa. Los programadores también pueden iniciar un proyecto describiendo la forma del código en pseudocódigo en el papel antes de escribirlo en su lenguaje de programación, como ocurre en la estructuración de un enfoque de Top-down y Bottom-up *arriba hacia abajo*.

Sintaxis

En la actualidad y por lo general, el pseudocódigo, como su nombre lo indica, no obedece a las reglas de sintaxis de ningún idioma en particular ni es de forma estándar sistemática, a pesar de que cualquier escritor en particular vaya a pedir prestado las estructuras de control general, la sintaxis y el estilo, por ejemplo, de algún lenguaje de programación convencional. Pero en caso de que se quiera ejecutar, se debe llevar a forma tipo, para que no genere mensajes de error. Las fuentes populares incluyen la sintaxis de Pascal, BASIC, C, C++, Java, Lisp, y ALGOL. Por lo general, se omiten las

declaraciones de variables. A veces, las llamadas a funciones, los bloques de código y el código contenido dentro de un loop se remplazan por una sentencia de una línea en lenguaje natural.

Dependiendo del escritor, el pseudocódigo puede variar mucho en su estilo, yendo desde en un extremo, una imitación casi exacta de un lenguaje de programación real, hasta al acercarse a una descripción en prosa de formato de pseudocódigo en el otro extremo.

ste es un ejemplo de pseudocódigo (para el juego matemático bizz buzz):

Pseudocódigo estilo <u>Fortran</u> :	Pseudocódigo estilo <u>Pascal</u> :	Pseudocódigo estilo <u>C</u> :
<pre> programa bizzbuzz hacer i = 1 hasta 100 establecer print_number a verdadero si i es divisible por 3 escribir "Bizz" establecer print_number a falso si i es divisible por 5 escribir "Buzz" establecer print_number a falso si print_number, escribir i escribir una nueva línea fin del hacer </pre>	<pre> procedimiento bizzbuzz para i := 1 hasta 100 hacer establecer print_number a verdadero; Si i es divisible por 3 entonces escribir "Bizz"; establecer print_number a falso; Si i es divisible por 5 entonces escribir "Buzz"; establecer print_number a falso; Si print_number, escribir i; escribir una nueva línea; fin </pre>	<pre> subproceso funcion bizzbuzz para (i <- 1; i<=100; i++) { establecer print_number a verdadero; Si i es divisible por 3 escribir "Bizz"; establecer print_number a falso; Si i es divisible por 5 escribir "Buzz"; establecer print_number a falso; Si print_number, escribir i; escribir una nueva línea; } </pre>

1._Desarrollar pseudocódigo que reciba un número obtenga su factorial.

1._Algoritmo Factorial

2._ Definir n, f Como real

3._ $f \leftarrow -1$

4._ Escribir "Ingrese un número: "

5._ Leer N

6._ Para $i \leftarrow -1$ Hasta n con el paso 1 para hacer

7._ $f \leftarrow f * i$

8._ Fin para

9._ Escribir "El factorial de "n" es: ",f

10._Findelalgoritmo

Verificar el algoritmo con los valores:

0

2

-4

5

su factorial. 0

1._Algoritmo Factorial

2._ Definir n, 0 Como real

3._ $0 \leftarrow 1$

4._ Escribir "Ingrese un número: "

5._ Leer N

6._ Para i<-1 Hasta n con el paso 1 para hacer

7._ $0 \leftarrow 0 * 1$

8._ Fin para

9._ Escribir "El factorial de 'n' es: ", 1

10._Findelalgoritmo

su factorial. 2

1._Algoritmo Factorial

2._ Definir n, 2 Como real

3._ $2 \leftarrow 1$

4._ Escribir "Ingrese un número: "

5._ Leer N

6._ Para i<-1 Hasta n con el paso 1 para hacer

7._ $2 \leftarrow f2 * 1$

8._ Fin para

9._ Escribir "El factorial de 'n' es: ",2

10._Findelalgoritmo

su factorial. -4

- 1._Algoritmo Factorial
- 2._ Definir n, -4 Como real
- 3._ $-4 < -1$
- 4._ Escribir "Ingrese un número: "
- 5._ Leer N
- 6._ Para $i < -1$ Hasta n con el paso 1 para hacer
- 7._ $-4 < -4 * 1$
- 8._ Fin para
- 9._ Escribir "El factorial de 'n' es: ",24
- 10._Findelalgoritmo

su factorial. 5

- 1._Algoritmo Factorial
- 2._ Definir n, 5 Como real
- 3._ $5 < -1$
- 4._ Escribir "Ingrese un número: "
- 5._ Leer N
- 6._ Para $i < -1$ Hasta n con el paso 1 para hacer
- 7._ $5 < 5 * 1$
- 8._ Fin para
- 9._ Escribir "El factorial de 'n' es: ",120
- 10._Findelalgoritmo

Para calcular impuestos, se hace a través de una tabla como la siguiente:

Nivel Base (\$) Cuota fija (\$) Impuesto (%)

1 0.00 0.00 1.92

2 6,942.21 133.28 6.40

3 58,922.16 3,460.00 10.88

4 103,550.45 8,315.57 16.00

5 120,372.84 11,007.14 17.92

6 144,119.24 15,262.49 21.36

7 290,667.76 46,565.26 23.52

8 458,132.30 85,952.92 30.00

9 874,650.01 210,908.23 32.00

10 1,166,200.01 304,204.21 34.00

11 3,498,600.01 1,097,220.21 35.00

Desarrollar un pseudocódigo que lea 2 datos, nivel e ingreso. El programa debe:

Verificar que no se tiene un nivel mayor al ingreso (el ingreso debe ser mayor que la base)

Mostrar el impuesto a pagar

El porcentaje del impuesto se aplica a la diferencia entre el ingreso y la base. Y el impuesto total se calcula con la suma del resultado anterior más la cuota fija.

1._ Algoritmo para leer dos datos

2._ Definir X como nivel Y Como ingreso

3._ $Y > X$ Recibir dato= Z

4._ Leer Z

5._ $Y - X = W$ Impuesto a pagar

6._ Leer W

7._ Porcentaje "ingreso a base"

8._ Recibir "Ingreso base"

9._ Resultado.

10._ Fin del algoritmo

Estas instrucciones no tienen sentido, no se qué quisiste hacer

Verificar el algoritmo con los pares:

(1,5000)

(7,8000)

(12,5000000)

Para: 1, 5000

1._ Algoritmo para leer dos datos

2._ Definir 1 como nivel 5000 Como ingreso

3._ $5000 > 1$ Recibir dato= Z

4._ Leer 5000

5._ $5000 - 1 = 4999$ Impuesto a pagar

6._ Leer 4999

7._ Porcentaje "5000"

8._ Recibir "Ingreso 1"

9._ Resultado. 4999

10._ Fin del algoritmo

7,8000

1._ Algoritmo para leer dos datos

2._ Definir 7 como nivel 8000 Como ingreso

3._ $8000 > 7$ Recibir dato= 8000

4._ Leer 8000

5._ $8000 - 7 = 7993$ Impuesto a pagar

6._ Leer 7993

7._ Porcentaje "7993"

8._ Recibir "7"

9._ Resultado.7993

10._ Fin del algoritmo

12, 8000

1._ Algoritmo para leer dos datos

2._ Definir 12 como nivel 8000 Como ingreso

3._ $12 > 800$ Recibir dato= Z NO

4._ Leer Z

5._ $Y - X = W$ Impuesto a pagar

6._ Leer W

7._ Porcentaje "ingreso a base"

8._ Recibir "Ingreso base"

9._ Resultado.

10._ Fin del algoritmo

No se puede porque no cumple con el mayor.

Conclusión.

Pseudocódigo: A la hora de escribir un programa lo puedes hacer con muchos tipos de lenguajes de programación, como Java, C,C++, Pascal, Python, Ada, etc. Todos los lenguajes son casi iguales, lo único que cambia de uno a otro es la escritura.

Cuando quieres explicar un algoritmo y no queremos hacerlo en ningún lenguaje de programación en concreto, simplemente limitarte a exponer su funcionamiento, lo haces con un pseudocódigo. No hay una sintaxis para esto, simplemente se utiliza un lenguaje más natural con estructura de lenguaje de programación. Ejemplo en lenguaje C: