



Algoritmo



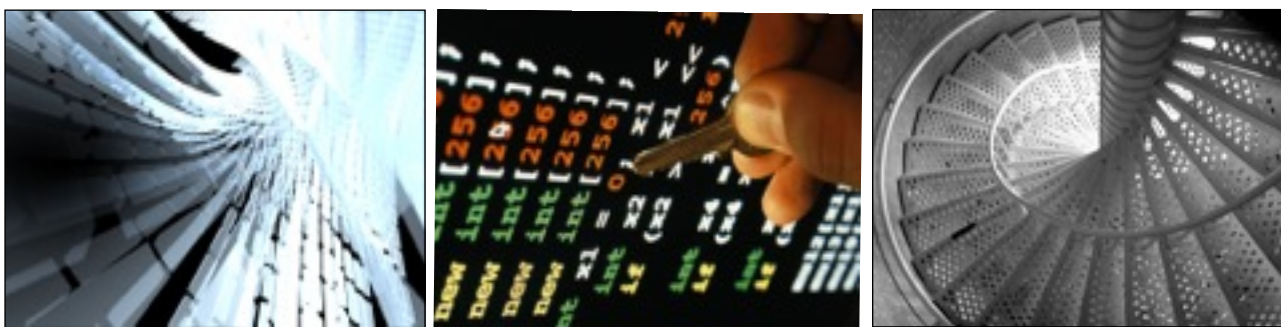
Programación Estructurada

Unidad 1 - Bases para el método de "programación estructurada"



Concepto

Un algoritmo es un método para resolver un problema. Aunque la popularización del término ha llegado con el advenimiento de la era informática, algoritmo proviene de Mohammed al-Khowarizmí, matemático persa que vivió durante el siglo IX y alcanzó gran reputación por el enunciado de las reglas paso a paso para sumar, restar, multiplicar y dividir números decimales; la traducción al latín del apellido en la palabra algorismus derivó posteriormente en algoritmo. Euclides, el gran matemático griego (del siglo IV antes de Cristo), que inventó un método para encontrar el máximo común divisor de dos números, se considera con Al-Khowarizmí el otro gran padre de la algoritmia (ciencia que trata de los algoritmos). El profesor Niklaus Wirth –inventor de Pascal, Modula-2 y Oberon– tituló uno de sus más famosos libros, Algoritmos + Estructuras de datos = Programas, significándonos que sólo se puede llegar a realizar un buen programa con el diseño de un algoritmo y una correcta estructura de datos. Esta ecuación será una de las hipótesis fundamentales consideradas en esta obra.



Un algoritmo es una serie de pasos ordenados y finitos que permiten resolver un problema o una tarea específica.

Características de un algoritmo

En los algoritmos siempre debe estar definido un inicio y un final y cada paso debe estar redactado con precisión. Las principales características fundamentales que debe cumplir todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

La definición de un algoritmo debe describir tres partes: Datos de entrada, proceso y datos de salida.

Datos de entrada: Son los valores que se necesitan para iniciar con el proceso. En la programación estos datos la computadora los recibirá desde algún dispositivo de entrada.

Proceso: Son las operaciones o procedimientos que se realizarán con los datos de entrada definidos.

Datos de Salida: Son los valores que representan el resultado del proceso ya sean operaciones o procedimientos. En la programación estos datos los regresará la computadora desde un dispositivo de salida.

Resolución de problemas computables

Lo primero que se tiene que asegurar es que la solución del problema sea computable, es decir que sea un problema que se pueda definir de manera finita y que sea contado o calculado. Si la solución del problema se vuelve exponencial entonces el problema deja de ser computable. También se pueden definir problemas no computables cuando intervienen emociones humanas o que la solución es subjetiva

Para poder resolver problemas computables es necesario definir el planteamiento del problema, esto quiere decir que las necesidades a resolver deben estar redactadas de forma clara, precisa y completa. En este planteamiento es necesario reconocer los datos de entrada, los procesos y los datos de salida, esto hará más sencilla la resolución.

La resolución de problemas en computadora consiste en las siguientes fases:

- Análisis del problema.
- Diseño del algoritmo.
- Implementación del algoritmo en la computadora
- Pruebas y depuración.
- Mantenimiento

fases para la resolución de problemas computables

Empezaremos por explicar cada una de las fases para la resolución del problema a través del método de la programación estructurada.

Análisis del problema. Cuando se está haciendo el análisis del problema debe quedar bien definidos cuales serán los datos de entrada, el o los procesos y los datos de salida ya que serán la base para diseñar el algoritmo.

Ejemplo A: El planteamiento del problema a solucionar es: *ver la película "WALL•E "* en el cine. Entonces lo primero que ubicamos en el planteamiento son los datos de entrada que en este problema son tener un boleto para la película. Los procesos serían entrar al cine así como ver la película. Por último los datos de salida serán salir del cine.

A continuación se establecen los pasos que definiera al algoritmo (creación del algoritmo) para ejemplo A:

1. inicio
2. comprar boleto para la película WALL-E
3. entrar a la sala indicada del cine
4. ver la película
5. salir del cine
6. final

Ejemplo B: El planteamiento del problema a solucionar es: *indicar el área de un triángulo*. Lo primero que tengo que saber entonces es cómo soluciono el problema fuera de la computadora (de forma manual). Una vez que sé cómo se soluciona el área de un triángulo entonces ubico en el planteamiento los datos de entrada que para este caso son cuánto mide la base del triángulo y cuánto mide la altura del triángulo. El proceso es la operación de multiplicar base x altura y dividirlo entre dos. Los datos de salida serán en resultado del área.

A continuación se establecen los pasos que definiera al algoritmo (creación del algoritmo) para el ejemplo B:

-
1. inicio
 2. guardar la base
 3. guardar la altura
 4. realizar la operación $\text{base} \times \text{altura} / 2$ y guardarlo en area
 5. imprimir el resultado que se encuentra en area
 6. Final

Diseño del algoritmo. Entonces podemos concluir que los algoritmos son la representación de la solución del problema, esta representación puede ser gráfica a lo que se le conoce como diagrama de flujo o bien puede ser escrita a lo que se le conoce como pseudocódigo. Por lo que, la siguiente fase es diseñar la solución primero gráficamente a través de su diagrama de flujo y luego acercarnos a la implementación en la computadora a través de un pseudocódigo que son los pasos en instrucciones definidas en nuestro propio idioma todavía. Cabe mencionar que esta fase se estudiará en temas y actividades posteriores.

Implementación del algoritmo en la computadora. La implementación no es más que seguir las instrucciones que están en nuestro idioma establecidas en el pseudocódigo y convertirlas a código, para ello se necesita elegir un lenguaje de programación, escribir las instrucciones en el lenguaje de la computadora elegido (código), comprobar que este bien escrito a través de la compilación y ejecutar el programa para comprobar que se tiene el resultado deseado o planteado en el diseño. Al resultado de esta fase se le conoce como creación de programas de computación. De la misma manera que en la fase del diseño, esta fase se estudiará en temas posteriores.

Pruebas y depuración. Las pruebas y depuración sirve para asegurar que el programa no vaya a fallar, son muy importantes porque éstas nos permiten hacer programas de calidad. Es importante mencionar que aunque no se indiquen explícitamente en las actividades es necesario hacer pruebas y depuraciones con los códigos de los programas que se vayan creando antes de entregarlos en la plataforma.

Mantenimiento. Al pasar el tiempo, las necesidades que llevaron a la creación de un programa pueden cambiar y entonces será necesario aplicar un mantenimiento para modificar y actualizar dicho programa, es por eso que podemos encontrar varias versiones de diversos programas de computación que nosotros mismo utilizamos cotidianamente.

Creación de algoritmos

En este momento nos enfocaremos a la creación de algoritmos de la fase de análisis para dar inicio al aprendizaje del método de la programación estructurada. Recuerda los pasos para la creación de algoritmos son: primero ubicar datos de entrada, procesos y datos de salida en el planteamiento del problema y segundo establecer pasos finitos, claros y precisos con instrucciones que establezcan la solución de problema.

A continuación se presentan varios ejemplos de la creación de algoritmos siguiendo las indicaciones expresadas con anterioridad.

Ejemplo 1

Planteamiento: Cálculo del área de un hexágono regular

Datos de entrada: obtener la longitud.

Proceso: aplicar la fórmula de cálculo de área de un hexágono regular

Datos de Salida: dar a conocer el resultado del cálculo

Algoritmo del ejemplo 1:

1. Inicio
2. guardar la longitud de un lado obtenida.
3. guardar la longitud del apotema obtenida.
4. Multiplicar la longitud del lado por el apotema, esto es el resultado parcial
5. Dividir el resultado parcial 1 entre dos, esto es el resultado parcial 2.
6. Multiplicar el resultado parcial 2 por 6, esto es el resultado final.
7. Mostrar el resultado final.
8. Final

Ejemplo 2

Planteamiento: Detectar de tres números, cuál es el mayor

Datos de entrada: tres números.

Proceso: ordenar los números de mayor a menor

Datos de Salida: mostrar el primer número

Algoritmo del ejemplo 1:

1. Inicio
2. guardar el primer número.
3. guardar el segundo número.
4. guardar el tercer número.

-
5. Si el primer número es mayor que el segundo número, entonces el número mayor es el primero, de lo contrario, el número mayor es el segundo. Esto es el resultado parcial.
 6. Si el tercer número es mayor al resultado parcial, entonces el número mayor es el tercer número, de lo contrario, el número mayor es el resultado parcial. Esto es el resultado final.
 7. Mostrar el resultado final.
 8. Final

Ejemplo 3

Planteamiento: Hacer café de grano

Datos de entrada: obtener el grano de café, obtener una cafetera

Proceso: procedimiento adecuado para utilizar la cafetera con el grano de café

Datos de Salida: café

Algoritmo del ejemplo 3:

1. Inicio
2. obtener el grano de café molido.
3. obtener una cafetera que funcione
4. introducir el agua según el número de tazas que se desean preparar en el lugar disponible en la cafetera
5. introducir el grano de café molido en el lugar disponible en la cafetera.
6. encender la cafetera.
7. tener el producto que genere la cafetera: café preparado.
8. Final

Referencias

Vancells-Floats, Joan. Algoritmos y programas. Editorial UOC.

Obtenida de <http://books.google.com/books?id=O55qXjsDjnYC&lpg=PA7&dq=algoritmo&hl=es&pg=PA1#v=onepage&q=algoritmo&f=false>

10 de junio de 2010.