PROGRAMACIÓN ORIENTADA A OBJETOS

Taller No. 1

Presentado por:

Iván Daniel Valencia Carvajal

Presentado a:

Carlos Andrés Londoño

Corporación de estudios tecnológicos del norte del valle

COTECNOVA

Cartago Valle

2016-2017

1. **Definir los conceptos fundamentales involucrados en el proceso de resolución de un problema a través de la computadora.**

Para resolver un problema a través de la computadora es necesario tener en cuenta una serie de pasos a seguir:

* Definición del problema
* Seleccionar métodos de solución
* Creación del algoritmo
* Poner a prueba el algoritmo
* Depuración y documentación del programa
* Validación de la solución
* Instalación, producción y mantenimiento del programa

**Definición del problema:**

En este primer paso se debe definir la situación que se intenta resolver, la definición debe ser clara y concisa para que otras personas lo puedan entender fácilmente, se deben diseñar objetivos específicos que se esperan lograr

**Seleccionar métodos de solución:**

El método de solución se emplea aplicando tres conceptos:

* Métodos geométricos
* Métodos analíticos
* Métodos interactivos
* **Métodos geométricos:**

Este método emplea gráficas para resolver problemas.

* **Métodos analíticos:**

Este método se usa haciendo planteamientos matemáticos y analíticos.

* **Métodos interactivos:**

Este concepto es el más usado para la solución de problemas en la computación, emplea procesos interactivos y recurrentes a la solución de problemas.

**Creación del algoritmo:**

Cuando se establece un método de solución este se debe pasar por escrito, bien definido y detalladamente claro, todos los más mínimos detalles deben estar tan completos como sea posible, nada puede quedar al azar.

**Poner a prueba el algoritmo:**

Como ya se ha hablado en el paso anterior el algoritmo debe estar bien definido, para así poder ponerlo a prueba en la máquina, estas instrucciones algorítmicas se deben definir en un lenguaje de programación, empleando programas tales como editores de texto y compiladores, para poder pasarlos al lenguaje de máquina, y verificar si su funcionamiento es el deseado.

**Depuración y documentación del programa:**

En este punto se toma en cuenta todos los errores que podemos cometer al momento de estar programando, como la falta de un punto y coma, las comillas, entre otros; esto hace que no compile nuestro código, o en otros casos compila pero no da los resultados deseados, como por ejemplo, el valor no está redondeado, se cruzan los resultados o cifras incoherentes al resultado que deseábamos esperar, todos estos errores deben ser eliminados de nuestra línea de código, haciendo un análisis exhaustivo. Además se debe comentar las líneas de código más importantes para poder guiarse más fácil.

**Validación de la solución:**

Dependiendo de la eficiencia con la que se realizaron los aspectos anteriores, y nuestro programa da los resultados acertados después de muchos intentos, además de que se halla analizado rústicamente en todos sus aspectos, se puede definir que nuestro programa funciona correctamente y puede ser entregado a su propietario o a su empresa.

**Instalación, producción y mantenimiento del programa:**

Este último paso quiere decir que nuestro programa ya está listo para realizar su instalación en los equipos de cómputo que se requiere, ya aquí se logra una mejor rapidez y flexibilidad frente a los cálculos y demás funcionalidades que se requieran, pero esto no quiere decir que no se necesitara más la ayuda del programador, el algunos casos el software que desarrollo va a presentar problemas, como bugs, o debilidades del mismo, en este caso, siempre se pone un periodo de prueba para el funcionamiento del programa, en este tiempo se verifica en que lo que está fallando y si es necesario el programa es revisado de nuevo y corregido.

1. **Definir lenguaje, programa, compilador e intérprete.**

**Lenguaje:**

Son un conjunto de instrucciones consecutivas y ordenadas (algoritmos), que nos lleva a una tarea específica, estas instrucciones son un código fuente, existen muchos lenguajes de programación, los más usados son php, java, C++.

**Programa:**

Cuando se habla de programa en informática, se está haciendo referencia a software, tan bien lo llaman aplicaciones, esto nos permite realizar determinadas tareas con una interfaz gráfica la cual el usuario pueda entender con facilidad haciendo el uso de todas sus herramientas, no solo se habla de software de equipos de cómputo, también se habla de Smart pones, smartv entre otros equipos tecnológicos.

**Compilador:**

Es un software que analiza un código fuente y lo traduce a código de máquina, ósea en código binario, el proceso de compilar tarda dependiendo de lo extenso que sea el código fuente.

**Intérprete:**

Se refiere a que solo analiza el código fuente y lo ejecuta directamente sin necesidad de espera, no genera ningún código máquina, como PHP.

1. **Identificar los antecedentes de la programación orientada a objetos.**

Simula (1967) es aceptado como el primer lenguaje que posee las características principales de un lenguaje orientado a objetos. Fue creado para hacer programas de simulación, en donde los "objetos" son la representación de la información más importante.

Smalltalk (1972 a 1980) es posiblemente el ejemplo canónico, y con el que gran parte de la teoría de la programación orientada a objetos se ha desarrollado.

1. **Definir los conceptos involucrados en la programación orientada a objetos.**

* Clase
* Herencia
* Objeto
* Método
* Evento
* Atributos
* Mensaje
* Propiedad o atributo
* Estado interno
* Componentes de un objeto
* Identificación de un objeto

1. **Identificar las etapas para la resolución de un problema bajo un enfoque orientado a objetos.**

* Definición del problema o dominio del problema
* Análisis orientado a objetos y diseño orientado a objetos
* Programación
* Documentación

1. **Identificar los tipos de datos que se emplean en el desarrollo de un programa.**

* Entero
* Real
* Carácter
* Cadena
* Booleano o lógico

1. **Definir los conceptos de variables, expresiones y constantes.**

**Variable:**

Es un ligar de almacenamiento temporal, donde se guarda un carácter o número, el cual puede ser modificado durante su función.

**Expresiones:**

Una expresión es una combinación de constantes, variables o funciones, que es interpretada de acuerdo a las normas particulares de precedencia y asociación para un lenguaje de programación en particular.

**Constante:**

Una constante es un valor asignado que no cambia durante su ejecución.

1. **Describir el procedimiento para declarar los formatos de las sentencias de asignación y de entrada y salida.**

**Asignación:**

La instrucción de asignación se usa para asignar valores a variables.

Ejemplo:

Variable = Valor asignado

**Entrada:**

Permiten leer valores y asignarlos a determinadas variables.

Ejemplo:

Leer (var1, var2, var 3) donde var# es el nombre de una variable

**Salida:**

Permite imprimir los resultados de los valores que se han guardado en la memoria.

Ejemplo:

Cout (“lista de variables”)

Bibliografía:

<http://www.monografias.com/trabajos105/uso-computadoras-solucion-problemas/uso-computadoras-solucion-problemas.shtml>

<https://colombiadigital.net/actualidad/articulos-informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html>

<http://definicion.de/programa-en-informatica/>

<https://funprogramacion.wikispaces.com/Compiladores+e+Int%C3%A9rpretes>

<https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos>

<http://es.slideshare.net/vladinike/programacion-orientada-a-objetos-introduccion>