INVESTIGACION:

PROGRAMACION ORIENTADA A OBJETOS.

Presentado por:

Jairo Andrés Salazar Ruiz

Presentado a:

Carlos Alberto Londoño

CORPORACION DE ESTUDIOS TECNOLOGICOS DEL NORTE DEL VALLE

TECNICO PROFESIONAL EN PROGRAMACIÓN DE APLICACIONES INFORMÁTICAS

Cartago 20 de Agosto del 2016

TALLER 1 PROGRAMACION ORIENTADA A OBJETOS

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

El proceso de solución de un problema con un computadora conduce a la escritura de un programa y a su ejecución en la misma. Aunque el proceso de diseñar programas es esencialmente un proceso creativo, se puede considerar una serie de pasos comunes.

Los pasos para la resolución de un problema son:

* Análisis del problema
* Diseño del algoritmo.
* Codificación.
* Compilación y ejecución.
* Verificación y depuración.

Los dos primeros pasos conducen a un diseño detallado escrito en forma de algoritmo. Durante el tercer paso (codificación) se implementa el algoritmo en un código escrito en un lenguaje de programación.

La fase de compilación y ejecución traduce y ejecuta el programa. En la fase de verificación y depuración el programador busca errores de las etapas anteriores y los elimina.

1. Definir lenguaje, programa, compilador e intérprete.

LENGUAJE

Es un sistema de comunicación estructurado para el que existe un contexto de uso y ciertos principios combinatorios formales. Existen contextos tanto naturales como artificiales.

PROGRAMA

Programa es una secuencia de instrucciones detalladas y codificadas a fin de que una computadora realice las operaciones para resolver un determinado problema. La programación de un computador consiste en crear programas o software, que luego será ejecutado por otro programa o el hardware de la computadora, asimismo, es un proceso de escribir, analizar, probar, mantener el código fuente de un software. En este punto, se añade al software los sistemas operativos como windows o los programas ofimáticos formado por los procesadores de texto como Microsoft Word, Microsoft Excel, Microsoft Outlook, etcétera.

COMPILADOR E INTÉRPRETE

Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje diferente. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio, o simplemente texto.

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

La Programación Orientación a Objetos (P.O.O.) surge en Noruega en 1967 con un lenguaje llamado Simula 67, desarrollado por Krinsten Nygaard y Ole-Johan Dahl, en el centro de cálculo noruego.

Simula 67 introdujo por primera vez los conceptos de clases, corrutinas y subclases (conceptos muy similares a los lenguajes Orientados a Objetos de hoy en día).

En los 70’s científicos del centro de investigación en Palo Alto Xerox (Xerox park) inventaron el lenguaje Small talk que dio respuesta al problema anterior (investigar no planificar).

Small talk fue el primer lenguaje Orientado a Objetos puro de los lenguajes Orientados a Objetos, es decir, únicamente utiliza clases y objetos (Java usa tipos de datos primitivos, o bien los Wrappers que son clases que encapsulan tipos de datos primitivos).

En los años 80’s Bjarne Stroustrup de AT&T Labs., amplió el lenguaje C para crear C++ que soporta la programación Orientada a Objetos.

En esta misma década se desarrollaron otros lenguajes Orientados a Objetos como Objective C, Common Lisp Object System (CIOS), object Pascal, Ada y otros.

En el inicio de los 90’s se consolida la Orientación a Objetos como una de las mejores maneras para resolver problemas. Aumenta la necesidad de generar prototipos más rápidamente (concepto RAD Rapid Aplication Developments). Sin esperar a que los requerimientos iniciales estén totalmente precisos.

En 1996 surge un desarrollo llamado JAVA (extensión de C++). Su filosofía es aprovechar el software existente. Facilitar la adaptación del mismo a otros usos diferentes a los originales sin necesidad de modificar el código ya existente.

En 1997-98 se desarrollan herramientas ‘CASE’ orientadas a objetos (como el diseño asistido por computadora).

Del 98 a la fecha se desarrolla la arquitectura de objetos distribuidos RMI, Corba, COM, DCOM.

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

La programación orientada a Objetos básicamente define una serie de conceptos y técnicas de programación para representar acciones o cosas de la vida real basada en objetos, a diferencia de otras formas de programación como por ejemplo la estructurada, con la POO trabajamos de manera distinta vinculando diferentes conceptos tales como clases, objetos, métodos, propiedades, estados, herencia, encapsulación entre otros, generando cada vez interrelaciones en nuestro desarrollo en pro del funcionamiento del sistema principal, definiendo el programa como un conjunto de estos objetos relacionados entre si

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

* Definición del problema o dominio del problema

Descripción en forma narrativa o esquemática, de modo claro y concreto el problema a resolver.

* 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.

Los tipos de datos hacen referencia al tipo de información que se trabaja, donde la unidad mínima de almacenamiento es el dato, también se puede considerar como el rango de valores que puede tomar una variable durante la ejecución del programa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Declaración** | **Memoria requerida** | **Rango** | **Descripción** |
| Booleano | boolean | - | true - false | Define una bandera que puede tomar dos posibles valores: true o false. |
| Byte | byte | 1 byte (8 Bits) | [-128 .. 127] | Representación del número de menor rango con signo. |
| Entero pequeño | short | 2 byte (16 Bits) | [-32,768 .. 32,767] | Representación de un entero cuyo rango es pequeño. |
| Entero | int | 4 byte (32 Bits) | [-231 .. 231-1] | Representación de un entero estándar. Este tipo puede representarse sin signo usando su clase *Integer* a partir de la Java SE 8. |
| Entero largo | long | 8 byte (64 Bits) | [-263 .. 263-1] | Representación de un entero de rango ampliado. Este tipo puede representarse sin signo usando su clase *Long* a partir de la Java SE 8. |
| Real | float | 4 byte (32 Bits) | [±3,4·10-38 .. ±3,4·1038] | Representación de un real estándar. Recordar que al ser real, la precisión del dato contenido varía en función del tamaño del número: la precisión se amplia con números más próximos a 0 y disminuye cuanto más se aleja del mismo. |
| Real largo | double | 8 byte (64 Bits) | [±1,7·10-308 .. ±1,7·10308] | Representación de un real de mayor precisión. Double tiene el mismo efecto con la precisión que float. |
| Carácter | char | 2 byte (16 Bits) | ['\u0000' .. '\uffff'] o [0 .. 65.535] | Carácter o símbolo. Para componer una cadena es preciso usar la clase *String*, no se puede hacer como tipo primitivo. |

1. Definir los conceptos de variable, expresiones y constantes.

VARIABLE

En programación, una variable está formada por un espacio en el sistema de almacenaje (memoria principal de un ordenador) y un nombre simbólico (un identificador) que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un valor. El nombre de la variable es la forma usual de referirse al valor almacenado: esta separación entre nombre y contenido permite que el nombre sea usado independientemente de la información exacta que representa. El identificador, en el código fuente de la computadora puede estar ligado a un valor durante el tiempo de ejecución y el valor de la variable puede por lo tanto cambiar durante el curso de la ejecución del programa

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.

CONSTANTES

Una constante es un valor que no puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído.

Corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde el programa almacena valores fijos.

1. Describir el procedimiento para declarar los formatos de la sentencia de la asignación y de entrada y salida.

Llamamos instrucción, orden o sentencia a cualquier expresión que el ordenador reconoce y le induce a realizar un proceso.

Inicio = equivalente a “Comienza a procesar información”

Fin = equivalente a “Termina de procesar información”

Variable1 = número, equivalente a “Asigna el número a la variable 1”

Variable1 = texto, equivalente a “Asigna el texto a la variable 1”

Variable1 = Variable2, equivalente a “Asigna el contenido de la variable 2 a la variable 1”

BIBLIOGRAFIA

<http://es.slideshare.net/serlob/resolucin-de-problemas-con-computadora>

<https://es.wikipedia.org/wiki/Lenguaje>

<http://www.significados.com/programa/>

<https://es.wikipedia.org/wiki/Compilador>

<http://sis324loo.blogspot.com.co/2008/09/historia-de-los-lenguajes-de_29.html>

<http://codejavu.blogspot.com.co/2013/05/conceptos-de-programacion-orientada.html>

<https://prezi.com/37q-cj1fvqdp/solucion-de-problemas-con-enfoque-orientado-a-objetos/>

<https://es.wikipedia.org/wiki/Tipo_de_dato#Tipos_de_datos_primitivos>

<https://es.wikipedia.org/wiki/Variable_(programaci%C3%B3n)>

<https://es.wikipedia.org/wiki/Expresi%C3%B3n_(inform%C3%A1tica)>

<https://es.wikipedia.org/wiki/Constante_(inform%C3%A1tica)>

<http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=291:sentencias-basicas-programacion-en-pseudocodigo-asignar-contenido-a-variables-pedir-ejemplos-cu00136a&catid=28:curso-bases-programacion-nivel-i&Itemid=59>