



INSTITUTO TECNOLÓGICO DE CHETUMAL

CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA: FUNDAMENTOS DE PROGRAMACIÓN

PARADIGMAS DE PROGRAMACIÓN

AUTORA:

MARTIN MARTIN YAMILI GUADALUPE



CHETUMAL, QUINTANA ROO A 2 DE SEPTIEMBRE DEL 2010.

ÍNDICE

INTRODUCCIÓN.....	3
-------------------	---

CONTENIDO

PROGRAMACIÓN MULTIPARADIGMA.....	4
----------------------------------	---

TIPOS DE PARADIGMAS.....	5
--------------------------	---

LOS MÁS COMUNES:

- programación imperativa.....5
- programación funcional.....6
- programación lógica.....6
- programación orientada a objetos6
- programación estructurada.....7
- programación guiada por eventos7

OTROS:

- computación en nube.....8
- programación declarativa.....8
- programación orientada a aspectos.....8
- programación con restricciones.....9

CONCLUSIONES.....	10
-------------------	----

REFERENCIAS.....	11
------------------	----

INTRODUCCIÓN

Los paradigmas de programación son la forma, que determinan los métodos y las herramientas que un programador usara en la construcción de un software.

Mayormente los lenguajes de programación están basados en uno o más paradigmas, ha estos se les puede llamar multiparadigmas.

También menciona los diferentes tipos de paradigmas que se conocen, pero solamente se hace referencia a los mas importante ya que suelen haber muchos más que no se mencionaran en esta investigación.

PROGRAMACIÓN MULTIPARADIGMA

Un lenguaje de programación puede soportar distintos paradigmas de programación con el objetivo de que un programador utilice el más conveniente a la hora de resolver un problema.

Ningún paradigma es capaz de resolver todos los problemas de forma sencilla y eficiente, por lo tanto es útil poder elegir entre distintos “estilos” de programación dependiendo del tipo de problema. También hay lenguajes que permiten mezclar los paradigmas que, en principio, parecerían irreconciliables. Se debe aclarar que hay subparadigmas que se incluyen en paradigmas más generales, pero hay otros que utilizan métodos de programación totalmente distintos entre sí e igualmente hay lenguajes que los combinan. Por ejemplo, el lenguaje Oz emplea programación lógica, funcional, orientada a objeto y otras.

Lenguajes como Delphi, C++ y Visual Basic combinan el paradigma imperativo, el procedural y el orientado a objetos.

Incluso lenguajes más puros en sus paradigmas como Prolog (paradigma lógico) o Scheme (paradigma funcional) poseen estructuras iterativas típicas de los lenguajes de paradigma imperativo.

TIPOS DE PARADIGMAS

Los más comunes:

Paradigma imperativo

Generalmente el código máquina, describe una programación como un flujo de instrucciones o comandos que van formando la estructura del software.

Es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.

La implementación de hardware de la mayoría de computadores es imperativa; prácticamente todo el hardware de los computadores está diseñado para ejecutar código de máquina, que es nativo al computador, escrito en una forma imperativa. Esto se debe a que el hardware de los computadores implementa el paradigma de las Máquinas de Turing. Desde esta perspectiva de bajo nivel, el estilo del programa está definido por los contenidos de la memoria, y las sentencias son instrucciones en el lenguaje de máquina nativo del computador (por ejemplo el lenguaje ensamblador).

Los lenguajes imperativos de alto nivel usan variables y sentencias más complejas, pero aún siguen el mismo paradigma. Las recetas y las listas de revisión de procesos, a pesar de no ser programas de computadora, son también conceptos familiares similares en estilo a la programación imperativa; cada paso es una instrucción, y el mundo físico guarda el estado (Zoom).

Paradigma funcional

Este tipo de paradigma demuestra un tipo de programación en forma de funciones matemáticas. Los lenguajes puramente funcionales evitan efectos secundarios que pueden ser comunes en otros tipos de paradigmas.

El objetivo es conseguir lenguajes expresivos y *matemáticamente elegantes*, en los que no sea necesario bajar al nivel de la máquina para describir el proceso llevado a cabo por el programa, y evitando el concepto de *estado* del cómputo. La secuencia de computaciones llevadas a cabo por el programa se regiría única y exclusivamente por la *reescritura* de definiciones más amplias a otras cada vez más concretas y definidas, usando lo que se denominan "definiciones dirigidas".

Paradigma lógico

Se definen reglas lógicas que luego a traves de un motor de inferencias lógicas, resolver problemas planteados por el sistema. Algunos programas que utilizan el paradigma lógico: prolog.

Hay muchos más paradigmas de programación y cada uno de ellos posee sus ventajas y desventajas, sin embargo sea el paradigma que elijamos podremos resolver todos los problemas por caminos diferentes a los demás.

Paradigma orientado a Objetos

Este tipo de programación posee como idea principal encapsular estado y operaciones en objetos. Estos objetos pueden comunicarse entre si, poseen clases y herencia. Se puede destacar como ventaja la facilidad de reutilización de código. Algún ejemplo de algunos lenguajes orientados a objetos pueden ser: Simula, Smalltalk, C++, Java, Visual Basic .NET, etc.

Paradigma estructurado

Básicamente el código se divide en bloques, estructuras, que pueden o no comunicarse entre sí. Este software se controla con secuencia, selección e interacción. Otorga una mejor comprensión del código y permite su reutilización de una forma más fácil.

Paradigma de programación guiada por eventos

Es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema o que ellos mismos provoquen.

Mientras que en la programación secuencial es el programador el que define cuál va a ser el flujo del programa, en la programación dirigida por eventos será el propio usuario, o lo que sea que esté accionando el programa, el que dirija el flujo del programa.

Al comenzar la ejecución del programa se llevarán a cabo las inicializaciones correspondiente y a continuación el programa quedará bloqueado hasta que se produzca algún evento.

Cuando alguno de estos eventos tenga lugar, el programa pasará a ejecutar el código del correspondiente manejador de evento.

Otros:

Computación en nube

Por computación en nube se entiende a aquella tecnología que permite ofrecer servicios de computación a través de Internet, de manera que todo aquello que puede ofrecer un sistema informático es ofrecido como servicio, al cual pueden a su vez acceder los usuarios sin que tengan conocimientos en la gestión de los diversos recursos que utilizan.

Se puede concebir el cloud computing, especialmente desde un punto de vista algo más técnico, como un modelo de aprovisionamiento rápido de recursos IT que potencia la prestación de servicios IT y servicios de negocio, facilitando a fin de cuentas la operativa del usuario final y del prestador del servicio.

Paradigma declarativo

No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe (declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado para encontrar esa solución. Es más complicado de implementar que el paradigma imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados problemas.

Programación Orientada a Aspectos

La Programación Orientada a Aspectos (POA) es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Gracias a la POA se pueden encapsular los diferentes conceptos que componen una aplicación en entidades bien definidas, eliminando las dependencias entre cada uno de los módulos. De esta forma se consigue razonar mejor sobre los conceptos, se elimina la dispersión del código y las implementaciones resultan más comprensibles, adaptables y reusables. Varias tecnologías con nombres diferentes se encaminan a la consecución de los mismos objetivos y así, el término POA es usado para referirse a varias tecnologías relacionadas como los métodos adaptivos, los filtros de composición, la programación orientada a sujetos o la separación multidimensional de competencias.

Programación con restricciones

Es un paradigma de la programación en informática, donde las relaciones entre las variables son expresadas en términos de restricciones (ecuaciones). Actualmente es usada como una tecnología de software para la descripción y resolución de problemas combinatorios particularmente difíciles, especialmente en las áreas de planificación y programación de tareas (calendarización).

Este paradigma representa uno de los desarrollos más fascinantes en los lenguajes de programación desde 1990 y no es sorprendente que recientemente haya sido identificada por la ACM (Asociación de Maquinaria Computacional) como una dirección estratégica en la investigación en computación.

Se trata de un paradigma de programación basado en la especificación de un conjunto de restricciones, las cuales deben ser satisfechas por cualquier solución del problema planteado, en lugar de especificar los pasos para obtener dicha solución.

La programación con restricciones se relaciona mucho con la programación lógica y con la investigación operativa. De hecho cualquier programa lógico puede ser traducido en un programa con restricciones y viceversa. Muchas veces los programas lógicos son traducidos a programas con restricciones debido a que la solución es más eficiente que su contraparte.

La diferencia entre ambos radica principalmente en sus estilos y enfoques en el modelado del mundo. Para ciertos problemas es más natural (y por ende más simple) escribirlos como programas lógicos, mientras que en otros es más natural escribirlos como programas con restricciones.

El enfoque de la programación con restricciones se basa principalmente en buscar un estado en el cual una gran cantidad de restricciones sean satisfechas simultáneamente. Un problema se define típicamente como un estado de la realidad en el cual existe un número de variables con valor desconocido. Un programa basado en restricciones busca dichos valores para todas las variables.

CONCLUSIONES

Los paradigmas juegan un papel importante en el mundo de la programación ya que a través de esta nos podemos hacer planteamientos para desarrollar un software.

Nos estructuran nuestra programación y nos dicen cómo resolver el problemas si a través de sentencias lógicas o matemáticas.

También ahora se pueden comprender mejor las diferentes definiciones de cada tipo de paradigmas. Los lenguajes de programación a veces pueden tener influencias de otros tipos de paradigmas.

REFERENCIAS BIBLIOGRÁFICAS

<http://www.alegsaonline.com/art/14.php>

http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_multiparadigma

<http://www.eviltec.com/Software/Programaci%C3%B3n/paradigmas-de-programacion.html>

http://es.wikipedia.org/wiki/Programaci%C3%B3n_Orientada_a_Aspectos

http://es.wikipedia.org/wiki/Programaci%C3%B3n_con_restricciones