

Paradigmas de programación

Paradigmas de programación.

Autor 1: Juan camilo Sánchez Fernández

Ingeniería de sistemas y computación, universidad tecnológica de Pereira

Correo-e: j.sanchez5@utp.edu.co

I. INTRODUCCIÓN

Resumen— Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con un conjunto de normas específicas, tales como: estructura modular, fuerte cohesión, alta rentabilidad, etc.

Los paradigmas pueden ser considerados como patrones de pensamiento para la resolución de problemas. Desde luego siempre teniendo en cuenta los lenguajes de programación, según nuestro interés de estudio.

No es mejor uno que otro, sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro.

Hay multitud de ellos atendiendo a alguna particularidad metodológica o funcional. Cuando un lenguaje refleja bien un paradigma particular, se dice que soporta el paradigma, y en la práctica un lenguaje que soporta correctamente un paradigma, es difícil distinguirlo del propio paradigma, por lo que se identifica con él.

Palabras clave— paradigma, programación, programa, tipos

Abstract— A programming paradigm is a basic model of program design and development, which allows to produce programs with a set of specific standards, such as: modular structure, strong cohesion, high profitability, etc.

Paradigms can be considered as thought patterns for problem solving. Of course, always taking into account the programming languages, according to our study interest.

One is not better than another but each has advantages and disadvantages. There are also situations where one paradigm is more appropriate than another.

There are many of them attending to some methodological or functional particularity. When a language reflects a particular paradigm well, it is said that it supports the paradigm, and in practice a language that correctly supports a paradigm, it is difficult to distinguish it from the paradigm itself, so it identifies with him.

Key Word — paradigm, programming, program, types

Hay diferentes tipos de paradigmas de programación aquí haremos mención de los principales de programación los cuales son: imperativo, declarativo, funcional, lógico y orientado a objetos



II. CONTENIDO

Programación imperativa: Son aquellos que facilitan los cálculos por medio de cambios de estado, entendiendo como estado la condición de una memoria de almacenamiento. Los lenguajes estructurados en bloques, se refieren a los ámbitos anidados, es decir los bloques pueden estar anidados dentro de otros bloques y contener sus propias variables. Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones, entendiendo éstas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas, en las que se verifican ciertas propiedades como la "transparencia referencial" (el significado de una expresión depende únicamente del significado de sus subexpresiones), y por tanto, la carencia total de "efectos laterales".

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.

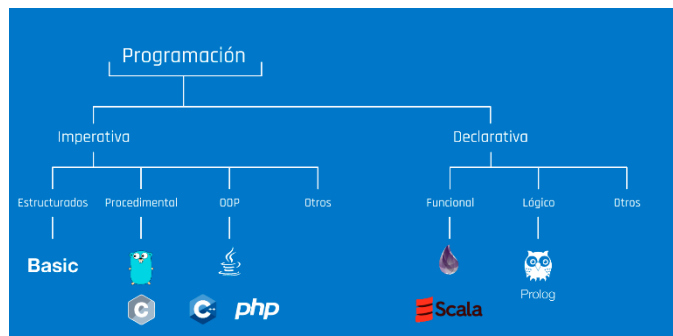
La secuencia de computaciones llevadas a cabo por el programa se regiría única y exclusivamente por la "reescritura" de

definiciones más amplias a otras cada vez más concretas y definidas, usando lo que se denominan "definiciones dirigidas".

A este tipo de paradigma de programación se le suele llamar algorítmico

Otras características propias de estos lenguajes son la no existencia de asignaciones de variables y la falta de construcciones estructuradas como la secuencia o la iteración (lo que obliga en la práctica a que todas las repeticiones de instrucciones se lleven a cabo por medio de funciones recursivas)

Paradigma declarativo: en contraposición a la programación imperativa, es un paradigma de programación que está basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. La solución es obtenida mediante mecanismos internos de control, sin especificar exactamente cómo encontrarla (tan solo se le indica a la computadora qué es lo que se desea obtener o qué es lo que se está buscando). No existen asignaciones destructivas, y las variables son utilizadas con transparencia referencial.



La Programación Lógica: es un Paradigma de Programación basado en la Lógica.

Los programas construidos en un lenguaje lógico están contruidos únicamente por expresiones lógicas, es decir, que son ciertas o falsas, en oposición a una expresión interrogativa (una pregunta) o expresiones imperativas (una orden). Un ejemplo de lenguaje lógico es Prolog (Programación lógica).

Prolog, proveniente del inglés Programming in Logic, es un lenguaje lógico bastante popular en el medio de investigación en Inteligencia Artificial. Prolog es un lenguaje muy diferente, tanto de los imperativos como Fortran, Pascal, C etc., como de los funcionales como Lisp. En todos los mencionados, las

instrucciones se ejecutan normalmente en orden secuencial, es decir, una a continuación de otra, en el mismo orden en que están escritas, que sólo varía cuando se alcanza una instrucción de control (un bucle, una instrucción condicional o una transferencia).

En Prolog, las cosas son distintas: el orden de ejecución de las instrucciones no tiene nada que ver con el orden en que fueron escritas. Tampoco hay instrucciones de control propiamente dichas. Para trabajar con este lenguaje, un programador debe acostumbrarse a pensar de una manera muy diferente a la que se utiliza en los lenguajes clásicos.

Paradigma orientado a objetos: La programación orientada a objetos (OOP, por las siglas inglesas de Object-Oriented Programming) es una forma de programar que proliferó a partir de los años ochenta.

La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto).

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

De esta forma, un objeto contiene toda la información, (los denominados atributos) que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases (e incluso entre objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos). A su vez, dispone de mecanismos de interacción (los llamados métodos) que favorecen la comunicación entre objetos (de una misma clase o de distintas), y en consecuencia, el cambio de estado en los propios objetos.

Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separan (ni deben separarse) información (datos) y procesamiento (métodos).

Dada esta propiedad de conjunto de una clase de objetos, que al contar con una serie de atributos definitorios, requiere de unos métodos para poder tratarlos (lo que hace que ambos conceptos están íntimamente entrelazados), el programador debe pensar indistintamente en ambos términos, ya que no debe nunca separar o dar mayor importancia a los atributos en favor de los métodos, ni viceversa.

Las principales diferencias entre la programación imperativa y la orientada a objetos son:

- La programación orientada a objetos es más moderna, es una evolución de la programación imperativa que plasma en el diseño de una familia de lenguajes conceptos que existían previamente con algunos nuevos.

- La programación orientada a objetos se basa en lenguajes que soportan sintáctica y semánticamente la unión entre los tipos abstractos de datos y sus operaciones (a esta unión se la suele llamar clase).

La programación orientada a objetos incorpora en su entorno de ejecución mecanismos tales como el polimorfismo y el envío de mensajes entre objetos.

REFERENCIAS

<https://www.monografias.com/trabajos107/paradigmas-programacion/paradigmas-programacion.shtml>

<https://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>