

Paradigmas de programación

Programming Paradigms

Juan Manuel Tamayo Garcia
 Oscar David Galindez Zuñiga
universidad tecnológica de Pereira, Risaralda, Colombia.
 Correo-e: juanmanuel.tamayo@utp.edu.co
d.galindez@utp.edu.co

Resumen— En este trabajo se presentará los diferentes paradigmas que se tiene de la programación. Se trata de entender primero que todo, que es un paradigma y cual es su definición en este campo científico. También nos enfocamos mucho en qué tipo de paradigmas de programación existen en la actualidad, y los más utilizados. Su significado, influencia y su por qué, añadir que daremos a conocer sus ventajas y desventajas por medio de ejemplos claros y concisos.

Palabras clave— programación, imperativo, paradigmas, lógica, multiparadigma.

Abstract— in this work the different paradigms of programming will be presented. It is about understanding first of all, what is a paradigms and what is its definition in this scientific field. We also focus a lot on what kind of programming paradigms exist today, and the most used. Its meaning, influence and why, add that we will make known its advantages and disadvantages through clear and concise examples.

Key Word — programming, imperative, paradigms, logic, multiparadigm.

I. INTRODUCCIÓN

¿QUÉ ES UN PARADIGMA DE PROGRAMACIÓN?

Paradigma de programación es una propuesta tecnológica que es adoptada por una Comunidad de Programadores cuyo núcleo central es incuestionable en cuanto a que unívocamente trata de resolver uno o varios problemas claramente delimitados. La resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de Software. Tiene una estrecha relación con la formalización de determinados lenguajes en su momento de definición. Un paradigma de programación está delimitado en el tiempo en cuanto a

aceptación y uso ya que nuevos paradigmas aportan nuevas o mejores soluciones que la sustituyen parcial o totalmente. La introducción puede contener:



- Programación Imperativa
- Programación funcional
- Programación lógica
- Declarativo
- POO Orientado a Objetos
- Por procedimientos

Habitualmente se mezclan todos los tipos de paradigmas a la hora de hacer la programación. De esa manera se origina la programación multiparadigma, pero el que actualmente es más usado de todos esos paradigmas es el de la programación orientada a objetos.

II. CONTENIDO

En la evolución de la programación han surgido diversas técnicas de programación que se han ido adaptando a las

necesidades tecnológicas e informáticas del momento. Aunque la forma de enfocar la elaboración de los programas es diferente en cada una de ellas, el objetivo es el mismo: facilitar la creación y el mantenimiento de programas informáticos. Estas técnicas se han traducido en diferentes filosofías de creación de programas que son los denominados paradigmas de programación.

Aunque hay muchos paradigmas de programación, los más comunes son:

Paradigma imperativo

Los programas imperativos contienen instrucciones que dicen al ordenador cómo realizar una tarea. Los primeros lenguajes imperativos fueron los códigos máquina de los ordenadores, que utilizaban instrucciones sencillas y permitían implementar el hardware fácilmente, pero no servían para desarrollar programas complejos.

El primer lenguaje imperativo que posibilitó la creación de programas con un nivel de complejidad elevado fue FORTRAN. Hoy en día está representado por los lenguajes de programación BASIC, C ó PASCAL, entre otros.

Paradigma funcional

Los programas funcionales se basan en el uso de una o más funciones dentro de las cuales se pueden utilizar funciones creadas anteriormente. Su objetivo es dividir el programa en módulos de forma que cada uno de éstos realice una única función.

El primer lenguaje de programación funcional fue LISP. Existen dos tipos de lenguajes funcionales: los puros (como HASKELL) y los híbridos (SAP, ML, Scheme).

Características

Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones

La no existencia de asignaciones de variables y la falta de construcciones estructuradas como la secuencia o la iteración

Existen dos grandes categorías de lenguajes funcionales: los funcionales puros y los híbridos

En contraste, los lenguajes funcionales puros tienen una mayor potencia expresiva, conservando a la vez su transparencia referencial.

Ventajas

Ausencia de efectos colaterales

Proceso de depuración menos problemático

Pruebas de unidades más confiables

Mayor facilidad para la ejecución concurrente

Desventajas:

Falta de estandarización

Bajo rendimiento de los programas

Paradigma de programación .. Universidad Tecnológica de Pereira.

Paradigma lógico

La programación lógica comprende la programación declarativa y la funcional. El proceso de elaboración de programas está basado en la lógica de primer orden y, a diferencia de los demás paradigmas, especifica qué debe hacer el programa y no cómo hacerlo.

Se emplea en aplicaciones de inteligencia artificial. El lenguaje de programación lógica por excelencia es PROLOG.

Paradigma orientado a objetos (POO)

La programación orientada a objetos expresa un programa como un conjunto de objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener, reutilizar y volver a utilizar. Su uso se popularizó a principios de los 90 y actualmente son muchos los lenguajes de programación asociados a este paradigma.

Las características del paradigma orientado a objetos son: encapsulamiento, abstracción, polimorfismo y herencia.

Muchos lenguajes utilizados en la actualidad están orientados a objetos, como Java, C++, Python o Delphi. Un lenguaje completamente orientado a objetos es Smalltalk.

PROGRAMACIÓN ESTRUCTURADA

Características

La estructura de los programas es clara, puesto que las instrucciones están más ligadas o relacionadas entre sí.

Los programas son más fáciles de entender, pueden ser leídos de forma secuencial.

Un programa escrito de acuerdo a los principios de programación estructurada no solamente tendrá una mejor estructura sino también una excelente presentación.

Ventajas

Reducción de los costos de mantenimiento

Reducción del esfuerzo en las pruebas y depuración

Los bloques de código son casi auto-explicativos, lo que reduce y facilita la documentación.

Se incrementa el rendimiento de los programadores

Desventajas

El principal inconveniente de este método de programación es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático el manejo de su código fuente

Se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar difícil su manejo

PROGRAMACIÓN MODULAR

Características

Este paradigma también se conoce como principio de ocultación de procedimientos y datos

Consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.

Se presenta históricamente como una evolución de la programación estructurada para solucionar problemas de programación más grandes y complejos de lo que ésta puede resolver.

Ventajas:

Al aplicar la programación modular, un problema complejo debe ser dividido en varios subproblemas más simples, y estos a su vez en otros subproblemas más simples.

En caso de que un módulo necesite de otro, puede comunicarse con éste mediante una interfaz de comunicación que también debe estar bien definida.

Es fácil de mantener y modificar

Es más fácil de escribir y depurar

Facilidad de controlar es decir descompone un problema en estructuras jerárquicas, de modo que se puede considerar cada estructura desde dos puntos de vista

Desventajas:

No se dispone de algoritmos formales de modularidad, por lo que a veces los programadores no tienen claras las ideas de los módulos

La programación modular requiere más memoria y tiempo de ejecución

ABSTRACCIÓN DE DATOS

Características

Proporciona un conjunto completo de operaciones válidas y útiles para cada tipo de dato

La abstracción de datos se logra mediante los tipos de datos que suministra el lenguaje y los subprogramas que van a implementar las operaciones permitidas

Se encapsula la representación interna de un dato junto con las implementaciones de todas las operaciones que se pueden realizar con ese dato.

Ventajas

Produce código reutilizable

Favorece la ausencia de errores, al reutilizar código ya probado y forzar a utilizar la estructura de datos correctamente

Aumenta la facilidad de uso y la legibilidad del código

Favorece la extensibilidad del código

Facilita y hace mas rapido el desarrollo de aplicaciones

Desventajas:

Las operaciones requieren mayor tiempo de procesamiento

En la mayoría no existen herramientas predefinidas, por lo tanto es el programador quien implementa las estructuras

Al agregar un nuevo elemento en algún lugar que no sea el último, se tiene que desplazar los elementos hacia abajo que están después de la posición de inserción deseada logrando

que se genere un espacio para poder agregar un nuevo elemento

PROGRAMACIÓN ORIENTADA A OBJETOS

Características

Abstracción: denota las características esenciales de un objeto, donde se capturan sus comportamientos

Encapsulamiento: significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción

Modularidad: propiedad que permite subdividir una aplicación en partes más pequeñas

Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando.

Ventajas:

Permite crear sistemas más complejos

Agiliza el desarrollo de software

Proporciona conceptos y herramientas con las cuales se modela y representa el mundo real tan fielmente como sea posible.

Fomenta la reutilización y extensión del código.

Desventajas:

Complejidad

para adaptarse

Mayor cantidad

de código



III. CONCLUSIONES

Podemos concluir que cada paradigma es importante en la programación y su uso en esta área científica, dejar claro que cada una tiene una función en específico por lo cual todos y cada uno de los paradigmas tiene ventajas y desventajas, hay que indagar un poco para reconocer el paradigma deseado y el cual se va a usar para mejor comodidad.

RECOMENDACIONES

leer atentamente cada paradigma tratar de entenderlo y saber cómo implementarlo. La clave es la buena comprensión de lectura e investigar más a profundidad este tema.

REFERENCIAS

http://cienciasfera.com/materiales/informatica/tecnologiainformacion/tema21/12_paradigmas_de_la_programacin.html

https://www.ecured.cu/Paradigmas_de_programaci%C3%B3n

<http://desarrollo--software.blogspot.com/2013/01/caracteristicas-de-los-paradigmas-de.html>

https://sandramarramirez.fandom.com/es/wiki/Paradigmas_de_Programaci%C3%B3n

<https://es.slideshare.net/AnelVeronicaUchihaLP/paradigmas-de-programacion-70287175>

https://techlandia.com/ventajas-desventajas-programacion-orientada-objetos-info_184901/

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