

Paradigmas de Programación



Centro: Universidad de los Altos / Universidad de Guadalajara.

Integrantes: Angel Alexis Serrano Hernandez.

Profesor: Carlos Javier Cruz Franco.

Actividad: Investigación.

¿Cuáles son los diversos paradigmas de la programación?

Los paradigmas de programación mas comunes que incluyen la imperativa, la programación declarativa, la programación orientada a objetos, la programación funcional, y la programación lógica, que lo vamos a describir o a detallar más afondo estos conceptos.

Programación imperativa: se describe como una serie de secuencias de instrucciones que cambian el estado de un programa. Es el paradigma más antiguo y se utiliza en muchos lenguajes de programación, como C y Pascal.

Programación Declarativa: se centra en el resultado deseado y describe cómo se alcanza ese resultado paso a paso. Aunque puede ser más difícil de implementar que la programación imperativa, tiene ciertas ventajas para determinar problemas.

Programación Orientada a Objetos: se basa en la idea de encerrar estado y operaciones en objetos, que se comunican mediante mensajes. Los lenguajes que soportan este paradigma incluyen Java, Python y C++.

Paradigma Funcional: trata la computación como la evaluación de funciones matemáticas, evitando la declaración y el cambio de datos. Los lenguajes funcionales evitan los efectos secundarios comunes en otros tipos de programación.

Programación Lógica: se basa en la definición lógicas para responder preguntas planteadas al sistema y resolver problemas. Un ejemplo de este paradigma es el Prolog.

Ventajas y Desventajas

En si cada paradigma en concreto tiene sus pros y sus contras, por ejemplo, la programación imperativa es simple y directa, pero a veces puede ser difícil de manejar para problemas complejos, ósea básicamente le cuesta poder resolver problemas difíciles.

La programación declarativa puede ser más fácil de entender desde fuera y más fácil todavía de manejar para ciertos problemas, pero tiene su desventaja, que es más difícil de implementar y a la vez puede requerir más tiempo de aprender.

La programación orientada a objetos promueve de cierta forma la reutilización de código y a la vez facilita en soluciones a determinado problemas. Sin embargo, puede ser muy complejo para ciertos problemas.

La programación funcional ofrece buenos niveles de abstracción y a la vez permite realizar ciertas optimizaciones en particular, pero puede ser muy complicado de entender para los programadores que no están acostumbrado a este paradigma.

Y finalmente y no la menos importante, sería la programación lógica, que puede ser muy completa para ciertos problemas, pero puede ser el doble de difícil de entender, aprender y utilizarla para los programadores que no estén familiarizado con la lógica.

¿Cuál es el mejor paradigma?

La programación orientada a objetos creo yo que por mi parte es una de las más importantes y la más popular, ya que se utiliza en muchos lenguajes de programación modernos, incluyendo Java y Python. Esta programación en si permite a los programadores que tener una organización de su código en torno a objetos, que son instancias de clases que contienen datos como métodos de manipular esos datos en sí.

En resumidas cuentas, esta organización puede hacer el código que sea más fácil de entender y mantener, especialmente creo yo para proyectos grandes y bastantes complejos.

Pero, a fin de cuentas, todo depende de ti a que quieras orientarte, básicamente en tema general no hay un paradigma en concreto que puedas decir que es el más top.



Referencias:

- Ba, S. (2016). Introducción Qué es un paradigma de programación. *www.academia.edu*.
https://www.academia.edu/25443431/Introducci%C3%B3n_Qu%C3%A9_es_un_paradigma_de_programaci%C3%B3n
- Equipo editorial de IONOS. (2021, 21 mayo). *Programación imperativa: resumen del paradigma de programación más antiguo*. IONOS Digital Guide.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/>
- *Programación funcional*. (s. f.). http://ferestrepoca.github.io/paradigmas-de-programacion/progfun/funcional_teor%C3%ADa/index.html