



CONCEPTOS GENERALES SOBRE PARADIGMAS DE PROGRAMACIÓN

PARADIGMAS DE PROGRAMACIÓN I

Carlos Rojas Sánchez

Licenciatura en Informática

Universidad del Mar

1. Introducción
2. Paradigmas de programación

Introducción

Un paradigma es un conjunto de principios y directrices que define un enfoque particular para diseñar, estructurar y escribir código. Cada paradigma impone una forma única de pensar sobre cómo debería desarrollarse el software y cómo interactúan sus componentes.

Los paradigmas de la programación son estilos que se siguen a la hora de programar un software.

Sistema de símbolos y reglas que permite la construcción de programas con los que la computadora puede operar así como resolver problemas de manera eficaz.

“Un hablante que se ha expuesto a un conjunto limitado de expresiones en su lengua, puede, a partir de esta experiencia finita, producir un número infinito de nuevos enunciados que los otros miembros de la comunidad de hablantes aceptará inmediatamente”

<https://www.levenez.com/lang/>

El índice TIOBE o índice de la comunidad de programación TIOBE (en inglés: TIOBE programming community index) mide la popularidad de los lenguajes de programación. El índice fue creado por la compañía TIOBE, con sede en Eindhoven, Países Bajos. El índice se calcula a partir del número de resultados del motor de búsquedas para consultas que contienen el nombre de un lenguaje de programación. El índice cubre búsquedas en Google, Google Blogs, MSN, Yahoo!, Baidu, Wikipedia y YouTube.

<https://www.tiobe.com/tiobe-index/>

Paradigmas de programación

Paradigmas de programación

La programación engloba diferentes enfoques y técnicas que guían el desarrollo de software. Estos enfoques, conocidos como paradigmas de programación, ofrecen distintas filosofías y metodologías para abordar problemas computacionales.

Paradigma imperativo

Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas. El desarrollador describe en el código paso por paso todo lo que hará su programa. Algunos lenguajes: Pascal, COBOL, FORTRAN, C, C++, etc.

Paradigma imperativo

Otros enfoques subordinados

- Programación estructurada: La programación estructurada es un tipo de programación imperativa donde el flujo de control se define mediante bucles anidados, condicionales y subrutinas, en lugar de a través de GOTO.
- Programación procedimental: Este paradigma de programación consiste en basarse en un número muy bajo de expresiones repetidas, englobarlas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse.
- Programación modular: consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más manejable y legible. Se trata de una evolución de la programación estructurada para resolver problemas de programación más complejos.

Paradigma declarativo

Este paradigma no necesita definir algoritmos puesto que describe el problema en lugar de encontrar una solución al mismo. Este paradigma utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas.

Este paradigma a su vez se divide en dos:

- Programación Lógica: Prolog
- Programación funcional: Lisp, Scala, Java, Kotlin

En este paradigma, el énfasis está en la evaluación de funciones matemáticas puras. Se evita el cambio de estado y la mutabilidad de los datos, lo que facilita la escritura de código más conciso y fácil de razonar. Lenguajes como Haskell, Lisp y Scala son ejemplos de lenguajes funcionales.

En este enfoque, los programas están basados en reglas lógicas. Se define qué se quiere obtener y el sistema de programación lógica se encarga de encontrar la solución. Prolog es uno de los lenguajes más conocidos que sigue este paradigma.

Paradigma orientado a objetos

En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

Paradigma orientado a objetos

Podemos definir un objeto como una estructura abstracta que, de manera más fiable, describe un posible objeto del mundo real y su relación con el resto del mundo que lo rodea a través de interfaces. Ejemplos de lenguajes de programación orientados a objetos serían Java, Python o C-Sharp.

Paradigma orientado a objetos

- Abstracción de datos
- Encapsulación
- Eventos
- Modularidad
- Herencia
- Polimorfismo

Paradigma orientado a eventos

Este paradigma se centra en responder a eventos que ocurren en el sistema, como clics de ratón o pulsaciones de teclas. Los programas están estructurados en torno a la definición de manejadores de eventos que se activan cuando ocurre un evento específico. JavaScript en el contexto de desarrollo web es un ejemplo común de este paradigma.

Paradigma orientado a eventos

Algunos ejemplos de plataformas y herramientas que utilizan la programación orientada a eventos incluyen el modelo de eventos del navegador web (como JavaScript en el navegador), frameworks de desarrollo de aplicaciones web como Node.js, y bibliotecas como Qt para desarrollo de interfaces gráficas.

Este paradigma se basa en escuchar lo que emite un evento o cambios en el flujo de datos, en donde los objetos reaccionan a los valores que reciben de dicho cambio. Las librerías más conocidas son Project Reactor, y RxJava.

La programación reactiva es un paradigma que se centra en la propagación automática de los cambios. Se basa en la idea de que los sistemas deberían ser capaces de reaccionar de manera automática y eficiente a los cambios en los datos o en el estado del sistema. Este enfoque es especialmente útil en aplicaciones donde la información fluye constantemente y los eventos pueden ocurrir en cualquier momento.

Otros paradigmas - Programación reactiva

Algunos conceptos clave en la programación reactiva incluyen:

- **Observables:** Representan secuencias de eventos o cambios en los datos. Los observables pueden emitir valores, errores o indicar que han terminado.
- **Observadores:** Son los componentes que están suscritos a los observables y reaccionan ante los eventos que emiten. Los observadores pueden ejecutar código en respuesta a los cambios en los flujos de datos.
- **Operadores:** Permiten transformar, combinar y manipular flujos de datos de manera declarativa. Estos operadores facilitan la composición y la creación de flujos más complejos a partir de flujos más simples.