

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.

Otros enfoques subordinados al paradigma de programación imperativa son:

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

PROGRAMACIÓN ORIENTADA 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 promueve 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.

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#.

La programación orientada a objetos se sirve de diferentes conceptos como:

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

PROGRAMACIÓN REACTIVA

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. React/Angular usan RxJs para hacer uso de la programación reactiva.

PROGRAMACIÓN FUNCIONAL

Una subcategoría de la estrategia declarativa es la llamada programación funcional, que se utiliza en particular en el desarrollo de los siguientes programas o códigos:

- Aplicaciones técnicas y matemáticas
- Inteligencia Artificial (IA)
- Compiladores y analizadores
- Algoritmos

Se centra en las funciones. En un programa funcional, todos los elementos pueden entenderse como funciones y el código puede ejecutarse mediante llamadas de función secuenciales. Por el contrario, no se asignan valores de forma independiente.

PARADIGMA LÓGICO

Estudia el uso de la lógica para el planteamiento de problemas y el control sobre las reglas de inferencia para alcanzar la solución automática.

La programación lógica, junto con la funcional, forma parte de lo que se conoce como Programación Declarativa, es decir la programación consiste en indicar como resolver un problema mediante sentencias, en la Programación Lógica, se trabaja en una forma descriptiva, estableciendo relaciones entre entidades, indicando no como, sino que hacer.

PROGRAMACIÓN ORIENTADA A 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, definidos por el usuario o que ellos mismos provoquen. al comenzar la ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento. es la base de lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

Referencias

Obtenido de la web <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/programacion-funcional/>

Universidad Nacional de Colombia.
https://ferestrepoca.github.io/paradigmas-de-programacion/proglogica/logica_teoría/proglogica.html