**Ensayo: Comparación entre el Uso de Dynamic y los Tipos Estáticos en Programación**

Tanto los tipos dinámicos como los estáticos son metodologías utilizadas para gestionar datos dentro de los lenguajes de programación, y cada uno tiene su propio conjunto de fortalezas y

En la codificación actual, la gente suele debatir los beneficios de utilizar tipos estáticos y dinámicos. Al diseñar un proyecto de la vida real, es vital reconocer los beneficios y desventajas asociados con cada enfoque.

**Tipos estáticos: seguridad y rendimiento.**

"Los tipos estáticos permiten al compilador determinar el tipo de datos de una variable antes de ejecutar el programa". Esto ofrece varias ventajas. ¿Podría simplificarme esta oración? Primero, le permite al programador tener más seguridad ya que el compilador puede detectar errores antes de ejecutar el programa. Este método ayuda a que el código sea más fácil de leer, porque otros programadores (o incluso la misma persona más adelante) pueden ver fácilmente qué tipo de datos se están utilizando. en cualquier momento. Esto se debe a que sabe exactamente qué tipo de datos contendrá cada variable.

El esquema debe consistir en encontrar errores al codificar y hacer que el código sea fácil de entender para que la creación de aplicaciones comerciales importantes o relacionadas con el dinero sea sólida y confiable.

**Dinámica: Flexibilidad y Agilidad.**

Por otro lado, el uso de una característica en lenguajes de tipo estático, como C#, le permite saber qué tipo de datos puede contener una variable cuando el programa se está ejecutando. Frase: Esto ofrece mucha adaptabilidad, particularmente cuando el tipo de los datos no están claros al principio o cuándo el código debe ser amplio y no específico. Cuando se trabaja con diferentes bibliotecas como JavaScript o API COM, o se maneja información en varios formatos como JSON o XML, el uso dinámico puede facilitar la creación de programas.

La programación dinámica es beneficiosa porque le permite escribir código que es más adaptable y menos repetitivo, lo que puede reducir la cantidad de escritura necesaria. Sin embargo, esta flexibilidad también conlleva riesgos. Sin verificación en tiempo de compilación, solo se encuentran errores de escritura cuando el programa se ejecuta, pudiendo provocar fallos.Refraseando el texto: Introducir, en un sistema, un elemento que genera ambigüedad y aumenta la complejidad del proceso de identificación y rectificación de errores.

**Cuándo utilizar Dynamic en un proyecto real.**

El principal factor de decisión para elegir dinámico en lugar de estático es la naturaleza del proyecto y la situación específica en la que se crea. En determinadas situaciones, la utilización de tipos estáticos puede resultar en un código demasiado complicado y largo.

.