

Ejercicio 2: Ensayo sobre la utilidad de los tipos de datos en la programación.

Utilidad de los tipos de datos en programación

Introducción

En la programación es frecuente el uso de variables que siempre están determinadas por los tipos de datos. Dependiendo el punto de vista, limitar las características y tipos de operaciones de un espacio de memoria puede ser o no una ventaja, sin embargo, todos los lenguajes utilizan los tipos de datos, con la única diferencia en el nivel de restricción para convertir dinámicamente un espacio de memoria de un tipo de dato a otro.

La implementación de los tipos de datos en un lenguaje de programación específico, está estrechamente relacionado a las características de ese lenguaje, por ejemplo, si es interpretado o compilado y si es fuerte o débilmente tipificado. Aunque todos los lenguajes de programación utilizan tipos de datos, no todos definen los mismos, ni las mismas características y operaciones.

Tipos de datos

Según (Pratt & Zelkowitz, 2000), un tipo de datos es una clase de objetos ligados a un conjunto de operaciones para crearlos y manipularlos. Es decir que un tipo de dato es un espacio de memoria con restricciones de uso.

Clasificación

Los tipos de datos pueden ser *primitivos* o *compuestos*. Ambos se componen de atributos que los distinguen, valores que permiten y operaciones definidas para la manipulación de ese tipo de datos.

Lenguajes fuerte y débilmente tipificados

Los tipos de datos se implementan en todos los lenguajes, la diferencia radica en si estos se usan de manera estática (lenguaje fuertemente tipado) o dinámica (lenguaje débilmente tipado). Cada uno tiene sus ventajas y desventajas, siendo las demandas del problema a solventar lo que determinen cuál es mejor utilizar. También algunos lenguajes permiten crear tipos de datos especiales, llamados los tipos de datos creados por el usuario, que basan su estructura en los tipos de datos primitivos brindados por el sistema, permitiendo que un objeto de este tipo tenga sus propios atributos y funcionalidades.

Relación entre los tipos de datos y lenguajes compilados e interpretados

Los lenguajes compilados son aquellos que se traducen al lenguaje máquina, en cambio, los lenguajes interpretados producen una forma intermedia del programa cuya ejecución es más fácil que la del programa original pero distinta del lenguaje máquina. (Pratt & Zelkowitz, 2000)

Generalmente los lenguajes compilados son fuertemente tipificados y hacen las validaciones de tipos de datos en la fase de compilación. Los lenguajes débilmente tipificados generalmente son interpretados y la validación de tipos es una responsabilidad directa del programador.

Conclusión

La utilidad de implementar tipos de datos en la programación radica en que estos permiten dotar de características específicas a un objeto y de esta manera determinar la forma en que serán manipulados evitando ambigüedades en los resultados obtenidos al momento de operarlos.

Un tipo de datos es la propiedad de un valor que indica los valores que puede tomar, qué operaciones se pueden realizar con él y la representación que tendrá a nivel interno de la arquitectura.

Referencias

Pratt, T., & Zelkowitz, M. (2000). *Lenguajes de programación diseño e implementación*. Prentice Hall.