C#

Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana entre ellos se encuentra el lenguaje C#

CARACTERÍSTICAS DE LA OOP

Algunas de las principales caracteristicas y razones por las que se usa la programación orientada a objetos o POO/ OOP son las siguientes:

Abstracción
Encapsulamiento
Modularidad
Principio de ocultación
Polimorfismo
Heréncia recoleccion de basura

SINTAXIS

La sintaxis de C# es muy expresiva, pero también sencilla y fácil de aprender. Cualquier persona familiarizada con C, C++ o Java, reconocerá al instante la sintaxis de llaves de C#. Los desarrolladores que conocen cualquiera de estos lenguajes puede empezar normalmente a trabajar en C# de forma productiva en un espacio muy corto de tiempo. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces, como tipos de valor que aceptan valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria. C# admite métodos y tipo genéricos, que proporcionan una mayor seguridad de tipos y rendimiento, e iteradores, que permiten a los implementadores de clases de colecciones definir comportamientos de iteración personalizados que son fáciles de usar por el código de cliente. Las expresiones de Language Integrated Query (LINQ) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

AREAS DE APLICACION

El lenguaje c# tiene un sin fin de aplicaciones y usos pero principalmente es utilizado para Unity el cual es un motor para la creación de videojuegos y aplicaciones

Es un entorno de desarrollo de software para sistemas operativos Windows. Este conjunto de herramientas se utiliza para crear sitios y aplicaciones web, así como generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio aplicaciones móviles.

TIPOS DE VARIABLE

En c# existen 2 tipos de variables, las de valor y las de referencia estos son algunos ejemplos:

Simples:
Signed integral: sbyte, short, int, long
Unsigned integral: byte, ushort, uint, ulong
Unicode characters: char
IEEE binary floating-point: float, double
High-precision decimal floating-point: decimal
Boolean: bool
Enum types
User-defined types of the form enum E {...}
Struct types
User-defined types of the form struct S {...}
Nullable value types
Extensions of all other value types with a null value

De Referencia:
Class types
Ultimate base class of all other types: object
Unicode strings: string
User-defined types of the form class C {...}
Interface types
User-defined types of the form interface I {...}
Array types
Single- and multi-dimensional, for example, int[] and int[,]
Delegate types
User-defined types of the form delegate int D(...)

ESTRUCTURA

Los programas de C# pueden constar de uno o más archivos. Cada archivo puede contener cero o más espacios de nombres. Un espacio de nombres puede contener tipos como clases, structs, interfaces, enumeraciones y delegados, además de otros espacios de nombres. El siguiente es el esqueleto de un programa de C# que contiene todos estos elementos.

TIPOS DE CLASE

Class. Public: Son muy comunes, accesibles desde cualquier otra clase en la misma librería (de otro modo hay que importarlas).
Class. Abstract: Aquellas que tienen por lo menos un método abstracto. No implementan sus métodos, sino que dan las bases para que sean implementados en la herencia.
Class. Final: Son las que terminan la cadena de herencia. Útiles por motivos de seguridad y eficiencia de un programa, ya que no permiten crear más sub-divisiones por debajo de esta clase.
bEspecifica que sus métodos son sincronizados, evitando problemas con los thread (hilo de ejecución), de forma que estos no pueden empezar a correr un método si no ha acabado el otro.