Cuestionario Teórico: C#

1. ¿Qué es el CLR?

2. Describa el proceso de compilación de C#.

3. ¿Qué es el CTS?

4. ¿Qué es un tipo por referencia (Reference Type)? ¿En qué se diferencia de un tipo por valor (Value Type)?

5. Explique las diferencias entre variables escalares y no escalares.

6. ¿Cuál es el punto de entrada (entry point) para los programas en C#?

7. ¿Cuál es la diferencia entre una conversión implícita y una explícita?

8. ¿Para qué sirven los modificadores out y ref? ¿En qué se diferencian?

9. Explique cómo funciona el método TryParse. ¿Qué retorna? ¿Qué recibe? ¿Cuál es la diferencia con el método Parse?

1->CLR es el entorno de ejecución de .NET Framework , maneja y administra la alocación de memoria.

Es también una máquina virtual que ejecuta aplicaciones y compila **lenguaje intermedio** a lenguaje máquina durante la ejecución de la aplicación (just-in-time / JIT).

•Lenguaje intermedio(IL): Los lenguajes de alto nivel de .NET (como C#) se compilan a un código agnóstico del hardware.

•Se lo conoce como CIL: Common Intermediate Language . (Similar al BYTECODE de JAVA)

2->Al compilar el código , este pasa a guardarse en un lenguaje intermedio , en un .exe o .dll. Al ejecutar el programa (runtime) en JIL pasa ese lenguaje intermedio a lenguaje maquina (native code) que se ejecuta automáticamente por el procesador.

3->CTS – Common Type System , es un conjunto común de tipos de datos y reglas que cualquier lenguaje de .NET Framework debe implementar a la hora de representar datos en la memoria.

Provee una biblioteca que contiene los tipos primitivos básicos.

Establece un marco que permite la ejecución inter-lenguajes.

4-> Reference Types:

•Estos tipos son representados por una referencia a la ubicación en memoria del valor actual del objeto, similar a un puntero en C.

•Si asignan un Reference Type a una variable y luego la pasan a una función, cualquier cambio en el objeto se verá reflejado. NO se genera una copia. (Se pasa el lugar en memoria donde está el dato)

•Value Types:

•Estos tipos son representados por sus valores.

•Si asignan un ValueType a una variable se copia el valor.

5->•Variables Escalares:

•Las variables escalares son constantes o variables que contienen un dato atómico y unidimensional. (un entero, un double , un char)

•Variables No Escalares:

•Las variables no escalares son array (vector), lista y objeto, que pueden tener almacenado en su estructura más de un valor.

6-> El punto de entrada en los programas de c# es la función “Main”.

7->Las conversiones Implicitas no necesitan intervención del programador,porque no hay riesgo de perdida de datos.

Las explicitas , necesitan intervención ya que puede haber perdida de datos.

8 -> Ref indica que el objeto ya esta inicializado antes de entrar a una función, la función puede leerlo , y modificarlo.

Out indica que el objeto aun no esta inicializado ( no tiene valor ) y que no puede ser leído por la función hasta que se le asigne un valor. Se debe asignar un valor antes de retornar.

Ref es de entrada y salida , out solo de salida .

Explique cómo funciona el método TryParse. ¿Qué retorna? ¿Qué recibe? ¿Cuál es la diferencia con el método Parse?

Tryparse recibe dos variables , la primera es un string , que deberá contener algo que queramos pasar a otra variable en un formato especifico, el segundo parámetro es una variable donde se va a guardar el valor que estaba en el string, pero en el formato que indicamos. El método tryparse retornara un booleano indicando si fue exitoso.

Bool Int32.TryParse(valor, out dondeguardo);

En cambio , El método Parse devuelve una excepción si no puede parsear el valor.