Parte Teorica Primer TP Programacion Jesus Sivira

Dentro del contexto del IDE de Visual Studio:

1. ¿Para qué se utiliza la ventana denominada Solution Explorer?

Solution Explorer es un panel muy útil, ya que visualiza los diversos archivos que forman parte de nuestro proyecto. Asi como las carpetas que contienen a los mismos, organiza las clases e interfaces del mismo.

2. ¿Para qué se utiliza la ventana denominada Properties?

Indica las propiedades de tiempo de diseño para objetos seleccionados y su configuración actual. Puede cambiar estas propiedades en tiempo de diseño. Si selecciona varios controles, la ventana Propiedades contiene una lista de las propiedades que comunes para todos los controles seleccionados.

3. ¿Para qué se utiliza la ventana denominada Toolbox?

La ventana Cuadro de herramienta muestra los controles que puede agregar a proyectos de Visual Studio, como la ventana solution explorer antes vista.

Dentro del contexto del lenguaje C#

4. ¿Cómo se declara un vector de tipo int de 10 elementos?

Private int[] vector

Vector = new int[10]

5. ¿Qué ámbitos existen para asignarle a los atributos y a los métodos?

Ámbito de las variables y metodos

* **A nivel de bloque:** Sólo el código del bloque tendrá la posibilidad de trabajar con la variable (por ejemplo un bucle ***for next***)
* **A nivel de función**: Sólo el código de la función donde se declara la variable podrá modificar su contenido, se llama normalmente variable local.
* **A nivel de clase**: Una variable declarada en el interior de una clase es accesible al código de esta clase sin restricción y eventualmente a partir de otras porciones de código en función del nivel de acceso de la variable.

**Nivel de acceso de las variables y metodos**

* **public**: Los elementos declarados serán accesibles desde cualquier porción de código del proyecto y desde cualquier otro proyecto que haga referencia a aquel donde están declarados. No se pueden utilizar dentro en el interior de las funciones.
* **protected**: Se puede utilizar en el interior de una clase. Permite restringir el acceso a la variable únicamente al código de la clase y todas las clases que hereden de ella.
* **internal**: Serán accesibles desde el ensamblado en el cual están declarados y tampoco se pueden utilizar en el interior de una función.
* **protected internal**: Es el nivel de acceso de protected e internal.
* **private**: restringe el acceso a la variable al módulo, a la clase o a la estructura en la cual está declarada. No se puede utilizar en el interior de un procedimiento o función.

6. ¿Qué significa que el parámetro de un método se pasa por valor o referencia?

En un caso el método va a recibir una copia exacta de la variable, que al modificarla e imprimir ambas variables dentro y fuera del método, la de adentro se va a ver modificada mientras que la de fuera del método no, este es pasar por valor, mientras que por referencia se va a recibir un puntero para dicha variable y al imprimir las dos van a estar modificadas.

7. ¿Para qué se utiliza null?

Normalmente, los tipos de valor que admiten valores NULL se usan cuando es necesario representar el valor indefinido de un tipo de valor subyacente. Por ejemplo, una variable booleana, o bool, solo puede ser true o false. Sin embargo, en algunas aplicaciones, un valor de variable puede estar sin definir o faltar. Por ejemplo, un campo de base de datos puede contener true o false, o puede no contener ningún valor, es decir, NULL. Puede usar el tipo bool? en ese escenario.

8. Enumere las estructuras de decisión. Ejemplifique cada una de ellas y explique en qué se diferencian.

Estructura IF, que verifica una variable en una condición, de ser cierta entra en ese bloque de código y de ser falsa no entra a hacer nada, a menos que haya una instrucción else, la cual se activa al ser falsa la condición.

Estructura Switch, es la que recibe un valor como opción y pasa a evaluarla en las distintas opciones pre establecidas en la instrucción, si el valor recibido es igual a una de esas opciones entra en ese bloque de código, de no ser igual a ninguna se dirige a la instrucción default.

La diferencia entre una y otra es que el if solo puede evaluar una condición verdadera o falsa, teniendo asi un sistema binario de decisión mientras que en switch podemos tener muchas mas instrucciones dependiendo de cada caso.

9. Enumere las estructuras de repetición. Ejemplifique cada una de ellas y explique en qué se diferencian.

Estructura for

for (int i = 0; i < 3; i++)

{

Console.Write(i);

}

Estructura foreach

var fibNumbers = new List<int> { 0, 1, 1, 2, 3, 5, 8, 13 };

foreach (int element in fibNumbers)

{

Console.Write($"{element} ");

}

Do while

int n = 0;

do

{

Console.Write(n);

n++;

} while (n < 5);

While

int n = 0;

while (n < 5)

{

Console.Write(n);

n++;

}

La diferencia entre ellas es que, el for ejecuta un bloque de código tantas veces diga el ciclo mientras una variable booleana especifica sea true, la diferencia con el foreach es que este ejecuta una acción para cada instancia de un objeto recorrido, es decir se detiene cuando recorre todo el objeto y no depende de una variable booleana. El while ejecuta un bloque de código siempre y cuando una variable booleana especifica sea true, el do while hace lo mismo solo que el while verifica la variable al inicio eso hace que si la variable es false nunca ejecutara el ciclo, mientras que en el do while, el ciclo se ejecuta al menos una vez, aun siendo la variable false desde el principio.

10. ¿Cómo diseñaría una lista enlazada simple de una clase llamada Profesor con los atributos Nombre (String) y Legajo (int)? Grafique y explique.