

Programación 1

Evaluación parcial 1

Profesor: Esteban Menéndez

Unidad/es:

Unidad 1: Introducción a la Programación Visual

Unidad 2: Estructuras Dinámicas I

Alumno:

Nicolás Esteban Olmedo



Dentro del contexto del IDE de Visual Studio:

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

La ventana Solution Explorer sirve para ver los diferentes archivos que incluye la solución. Si se le da click derecho, también puede usarse para agregar referencias (por ejemplo, para usar estructuras de Basic), agregar archivos (como más archivos de C# con clases). Renombrar archivos, etc.

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

La ventana Properties se usa para ver/modificar las propiedades de los elementos añadidos al formulario desde la vista de diseñador. Algunos de estas propiedades son Font, FontSize, Text, Name, etc.

Desde la sección "Events" se pueden definir los distintos eventos que se pueden generar para un determinado elemento añadido como button_Click, DragOver, etc

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

La ventana Toolbox se usa para agregar los controles de usuario necesarios para el programa. Por ejemplo button, textBox, dataGridView, timer, etc

Dentro del contexto del lenguaje C#:

4. Si tengo una variable de tipo Entero (Int) y tengo que guardarlo en una variable de tipo String.

¿Cómo realizo la conversión de tipo?

Para almacenar un int en un string se pueden ejecutar las siguientes operaciones:

1. string variableTipoString = Convert.ToString(variableTipoInt);
2. string variableTipoString = variableTipoInt.ToString();

5. ¿Cómo se declara un método? Ejemplo.

Los métodos se declaran bajo el formato:

ModificadorDeAcceso Return Type NombreDelMétodo(tipo parm, ...) {...cuerpo...}

dónde:

Modificador de acceso: Define desde dónde se puede acceder al método

Return Type: Define el tipo de dato que va a devolver el método

NombreDelMétodo: Define el nombre con el cual se va a llamar al método

tipo: tipo de dato del parámetro que va a usar el método como input

parm: nombre de la variable local con la que se va a trabajar dentro del método

Ejemplo:

public void NombreDelMétodo(string nombre, int edad) {...cuerpo...}





6. ¿Cuál es el uso ideal de una variable de tipo bool?

El uso ideal de una variable tipo bool es realizar comprobaciones lógicas. Una vez hecha la comprobación, se puede definir si una porción de código ejecuta o no.

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

7.1. if...else: Ejecuta el bloque de código dentro del if{} si la expresión se evalúa en true. Caso contrario, se ejecuta el bloque de código dentro del else{} y luego se ejecuta la siguiente sentencia del programa. En el caso en que no haya else{}, se ejecuta la siguiente sentencia del programa.

```
if((primerNodo = ultimoNodo) && (ultimoNodo == null)) {...código...}
else {...código...}
```

7.2. switch: Ejecuta el bloque de código dentro del case que evalúe en true. Se usa para evitar los ifs anidados porque facilita la lectura del código.

```
switch(temperatura)
{
    case < 0: {...código... break;}
    case >=0: {...código... break;}
    default: {...código... break;}
}
```

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

8.1. for: Se usa cuando se conoce la cantidad de iteraciones que se van a realizar

```
for(int i=0; i<valorFinal; i++) {...código...}
```

8.2. foreach: Se usa para iterar sobre una determinada cantidad de elementos en un conjunto de elementos

```
foreach(Button button in GroupBox) {...código...}
```

8.3 while: Se usa para iterar mientras la expresión booleana declarada entre paréntesis devuelva true. Se usa cuando se desconoce la cantidad exacta de iteraciones a realizar. Primero hace la comprobación de la sentencia y después ejecuta, por lo que, si la condición se cumple de antemano, puede no ejecutar nunca su bloque de código.

```
while(nodoActual.Siguiente != nodoSeleccionado) {...código...}
```



8.4. do...while: Se usa para iterar mientras la expresión booleana declarada entre paréntesis devuelva true. Primero ejecuta su bloque de código y después hace la comprobación de la sentencia, por lo que va a ejecutar su bloque de código al menos una vez

```
do{...código...}while(nodoActual.Siguiente != nodoSeleccionado)
```

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

Para definir una lista enlazada simple hay que:

1. Crear una clase Profesor que funcione de como nodo. Esta debe incluir:
 - . Variables de instancia privadas para almacenar el nombre y el legajo y una referencia al siguiente objeto de la lista.
 - . Propiedades públicas para manipular el nombre y el legajo y la referencia al siguiente objeto de la lista. Por convención, es el nombre de la variable de instancia con su primer letra en mayúscula.
 - . Constructores para instanciar un objeto de la clase Profesor.

Importante, como es una lista simple, cada nodo solo tiene una referencia al nodo siguiente.

2. Crear una clase Lista para relacionar las diferentes instancias de la clase Profesor. Esta debe incluir:
 - . Variables de instancia privadas para almacenar una referencia al primer y último nodo de la lista.
 - . Constructores para instanciar un objeto de la clase Lista.
 - . Los métodos que sean necesarios para trabajar con la lista. Estos pueden incluir métodos para agregar nodos al principio, a una posición determinada o al final de la lista; eliminar del principio, de una posición determinada o del final de la lista; buscar un nodo en particular, evaluar si la lista está vacía o no, etc

