

1erParcialProg - 06/10/2020

Tuesday, October 6, 2020
M.Luana Jaimez

Formularios:

- a. Heredan, directa o indirectamente de System.Windows.Forms.
- b. Son objetos que exponen propiedades, métodos y eventos.
- c. Utilizan el concepto de partial object.
- d. Su ciclo de vida consta de estas etapas: New, Load, Paint, FormClosing y FormClosed.
- e. Todas las anteriores.
- f. a, b y d.**
- g. a, b y c.
- h. Ninguna de las anteriores.

Se intenta sobrecargar un operador de decremento (--):

- a. Es un operador binario.
- b. No es un operador sobrecargable.
- c. Es un operador binario y ambos parámetros deben ser del mismo tipo.
- d. El valor de retorno debe ser del tipo que lo contiene.**
- e. El primer parámetro puede ser de un tipo distinto al de la clase que lo contiene.
- f. Ninguna de las anteriores.

¿Cómo debo mostrar a un formulario para 'esperar' un resultado?

- a. frmAlgo.Show().
- b. frmAlgo.Show(this).
- c. frmAlgo.ShowDialog().**
- d. frmAlgo.DialogResult.
- e. Ninguna de las anteriores.

Si existe una clase A y una clase B, siendo B una clase abstracta que hereda de A, y A posee un método virtual llamado Metodo(), entonces:

- a. A debe implementar Metodo().**
- b. A y B deben implementar Metodo().
- c. B no puede heredar de A.
- d. A podría no implementar Metodo().
- e. B debe implementar Metodo().
- f. Ninguna de las anteriores es correcta.

Para agregar ítems a un control ListBox, se utiliza:

- a. lstLista.Add("un valor").
- b. lstLista.Push("un valor")
- c. lstLista.Items.Add("un valor").**
- d. lstLista.Items.Push("un valor").
- e. Ninguna de las anteriores.

Objetos en C#:

- a. Son creados en tiempo de ejecución y su valor está alojado en el Stack.
- b. Sólo pueden acceder a miembros no estáticos.
- c. El tiempo de vida está vinculado al ámbito en el que se declara.
- d. Tienen una creación y destrucción no determinista.
- e. Tienen una creación y destrucción determinista.
- f. Todas las anteriores.
- g. Ninguna de las anteriores.**

Clases en C#:

- a. Deben contener atributos, propiedades, constructores y métodos.
- b. Si un método es público, se puede acceder desde cualquier lado accediéndolo como Clase.Metodo.
- c. Pueden contener atributos, propiedades, constructores y métodos, todos estáticos.
- d. Si una clase es estática, solo puede derivar clases estáticas.
- e. Su creación se da en tiempo de ejecución.
- f. Todas las anteriores.
- g. Ninguna de las anteriores.**

Cuáles de estas afirmaciones sobre los Namespaces NO son ciertas:

- a. Son agrupaciones lógicas.
- b. Son agrupaciones físicas.
- c. Deben contener obligatoriamente al menos una clase.
- d. No pueden contener más de una clase, siendo esto un error de compilación.
- e. No deben contener más de una clase, siendo esto un error en tiempo de ejecución.
- f. a y c.
- g. b y c.
- h. a, c, e y d.
- i. b, c, d y e.**

Formularios:

- a. Heredan, directamente o indirectamente de System.Windows.Forms.
- b. Son objetos que exponen propiedades, métodos y eventos.
- c. Utilizan el concepto de partial class.
- d. Su ciclo de vida consta de 7 etapas: New, Load, Paint, Activated, FormClosing, FormClosed y Dispose.
- e. Todas las anteriores.
- f. Ninguna de las anteriores.
- g. a, b y d.
- h. a, b y c.**

El CTS (Common Type System) define tipos de valor y referencia. Se tiene un array de enteros.

- a. Pertenece al grupo de tipos de valor (y sus elementos también).
- b. Pertenece al grupo de tipos de referencia (y sus elementos también).
- c. Por defecto es inicializado en null.**
- d. Por defecto es inicializado con "".
- e. a y c.
- f. b y c.
- g. c y d.
- h. Ninguna de las anteriores.

Operadores lógicos. Las evaluaciones se realizan por "cortocircuito", esto implica que:

- a. No se evaluará la segunda condición.
- b. Siempre se evaluará la primera condición.
- c. Puede evaluarse la segunda condición, si la primera es verdadera.
- d. Puede evaluarse la segunda condición, si la primera es falsa.
- e. b y c.
- f. b y d.
- g. b, c y d.**

Son pilares de la P.O.O.:

- a. El manejo de errores.
- b. La abstracción.
- c. El encapsulamiento.
- d. Los objetos.
- e. La herencia.
- f. El polimorfismo.
- g. Todas las anteriores.
- h. a, b, d, e y f.
- i. b, c, d, e y f.
- j. b, c, e y f.

Las conversiones implícitas se utilizarán cuando:

- a. Por medio de un casteo se convierte de un tipo a otro.
- b. La conversión implica pérdida de información.
- c. La conversión se realiza por intermedio de un método (por ejemplo, Parse).
- d. Todas las anteriores.
- e. Ninguna de las anteriores.

El CTS (Common Type System) define tipos de valor y referencia. Se tiene un array de strings.

- a. Pertenece al grupo de tipos de valor (y sus elementos también).
- b. Pertenece al grupo de tipos de referencia (y sus elementos también).
- c. Por defecto es inicializado en null.
- d. Por defecto es inicializado con "".
- e. a y c.
- f. b y c.
- g. c y d.
- h. Ninguna de las anteriores.

Colecciones:

- a. Son buenas para trabajar con cantidades dinámicas de elementos.
- b. Hay de distintos tipos, optimizadas para distintos comportamientos de sus elementos.
- c. Siempre son genéricas.
- d. Una colección genérica cumple la seguridad de tipos para que ningún otro tipo de datos se pueda agregar a ella.
- e. Todas las anteriores.
- f. a y b.
- g. b y c.
- h. a, b y d.
- i. Ninguna de las anteriores.

Una propiedad definida como abstracta en una clase:

- a. Debe tener obligatoriamente una implementación en las clases derivadas.
- b. Debe tener una implementación en la clase en la que se definió y también en las clases derivadas, caso contrario arrojaría una excepción.
- c. Debe tener una implementación en la clase en la que se declaró.
- d. No puede definirse una propiedad como abstracta.
- e. Ninguna de las anteriores.

Si se desea sobrescribir un método:

- a. Se debe cambiar el número y/o el tipo y/o el orden de los parámetros.
- b. El compilador de C# distingue métodos comparando las listas de parámetros.
- c. Se debe conservar el mismo nombre, parámetros y valor retorno.
- d. Todas las anteriores.
- d. Ninguna de las anteriores.

Operadores lógicos. Las evaluaciones se realizan por “cortocircuito”, esto implica que:

- a. No se evaluará la segunda condición.
- b. Siempre se evaluará la primera condición.
- c. Puede evaluarse la segunda condición, si la primera es verdadera.
- d. Puede evaluarse la segunda condición, si la primera es falsa.
- e. b y c.
- f. b y d.
- g. b, c y d.

Dada la siguiente porción de código, ¿Qué se mostraría en la consola?

```

Queue<string> a = new Queue<string>();

a.Enqueue("a");
a.Enqueue("b");
a.Enqueue("c");

while (a.Count > 0)
{
    Console.WriteLine(a.Dequeue());
}

```

- a. "a"
- b. "abc"
- c. "cba"
- d. "c"
- e. Sintácticamente está mal.
- f. Arroja una excepción.
- g. a y f.
- h. c y f.

¿Qué segmentos de memoria administra .Net?

- a. Estática y dinámica.
- b. Garbage collector.
- c. Diseño y ejecución.
- d. Stack.
- e. Heap, el segmento Stack es administrado por el programador.
- f. Heap y Stack.
- g. Ninguna de las anteriores

Dado el siguiente código y asumiendo que en la clase Persona se definió un atributo público llamado nombre (de tipo string) y un único constructor que recibe un parámetro de tipo string. ¿Qué haría la ejecución del código?

```

List<Persona> lista = new List<Persona>();
lista.Add(new Persona("Tito"));
lista.Add(new Persona("Pepé"));
lista.Add(new Persona("Roberto"));
lista.Add(new Persona("Julia"));
lista.Add(new Persona("Susana"));

foreach(Persona item in lista)
{
    if (item.nombre == "Tito")
    {
        lista.Remove(item);
    }
}

```

- a. Elimina la persona con nombre "Tito" de la colección.
- b. Arroja un error en tiempo de ejecución.
- c. La sintaxis es incorrecta.
- d. Debería sobrecargarse el operador == en la clase Persona para poder realizar la comparación.
- e. b y c son correctas.
- f. a y b son correctas.
- g. Ninguna de las anteriores.

Sobrecargas:

- a. La sobrecarga de métodos consiste en cambiar el número de parámetros, tipo de parámetros y el tipo de retorno.
- b. Una sobrecarga válida es cuando se cambian los nombres de los parámetros.
- c. El compilador distingue los métodos que están sobrecargados comparando la lista de parámetros.
- d. Las sobrecargas de métodos deben tener el mismo modificador de visibilidad.
- e. Todas las anteriores.
- f. Los constructores de instancia se deben sobrecargar.
- g. Los constructores de clase se pueden sobrecargar.

Constructores:

- a. Los constructores estáticos no reciben parámetros, si son públicos.
- b. Los constructores de instancia pueden recibir parámetros, a menos que sean protected.
- c. Si es de instancia, se puede utilizar this, pero no se puede acceder a miembros estáticos.
- d. Por convención, dentro de un constructor estático no se puede utilizar this.
- e. Los valores de retorno de los constructores de instancia dependen de la clase donde están definidos.
- f. Todas las anteriores.
- g. Ninguna de las anteriores.