

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Programación II

Apellido:		Fecha:	25/07/2017
Nombre:		Docente ⁽²⁾ :	
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP SP RSP FIN </div>		

(2) Campos a ser completados por el docente.

Recuperatorio 1er Parcial: 1 a la 10, Recuperatorio 2do Parcial: 11 a la 20, Final: Todas.

Para aprobar el examen se debe contar con al menos un 60% de respuestas correctas.
Todos los puntos valdrán 0.5.

1. Si tengo una clase A con solamente 3 constructores: un constructor de clase, otro privado y otro público:
 - a. El constructor de clase inicializará elementos de instancia.
 - b. El primer constructor por el que pasará será **siempre** el público.
 - c. El constructor estático será el primero en ejecutarse al instanciar cualquier objeto de ese tipo.
 - d. Todos los constructores deberán contener código.
 - e. El constructor privado sólo podrá alterar elementos que sean privados o protegidos.
 - f. El constructor de clase llamará al constructor privado.
 - g. Se llamará al constructor público desde el de clase.
 - h. Todas las anteriores.
 - i. Ninguna de las anteriores.

2. Cuáles de estas afirmaciones sobre los Namespace son ciertas:
 - a. Deben contener obligatoriamente al menos una clase.
 - b. Son agrupaciones lógicas.
 - c. Son agrupaciones físicas.
 - d. No pueden contener más de una clase, siendo esto un error de compilación.
 - e. Todas las anteriores.

3. Al sobrecargar el método Equals:
 - a. Se deberá sobrecargar el operador de igual y distinto.
 - b. Tendremos una advertencia acerca de otros comparadores de igualdad que no estamos sobrecargando.
 - c. Los parámetros recibidos podrán ser de cualquier tipo.
 - d. Estamos obligados a retornar algo del tipo Object.
 - e. Todas las anteriores.
 - f. Ninguna de las anteriores.

4. La relación de herencia entre clases implica que:
 - a. Se agregan a la clase hijo sólo los comportamientos públicos de la clase base.
 - b. En caso de haber métodos virtuales, estos se podrán sobrecribir.
 - c. Los constructores no forman parte de dicha relación, anulándose los de la clase base.
 - d. Sólo se incorporan comportamientos de la herencia directa, perdiendo las herencias anteriores.
 - e. Sea posible heredar de varias clases a la vez.
 - f. Se tiene la obligación de reescribir los métodos de la clase base, respetando sólo la firma de los mismos.
 - g. Todos los atributos, métodos y propiedades quedan accesibles por sus clases derivadas.
 - h. Todas las anteriores.
 - i. Ninguna de las anteriores.

5. Si necesito que un método retorne dos valores, puedo utilizar:
- var
 - out
 - public
 - struct
 - Todas las anteriores.
 - Ninguna de las anteriores.
6. Un indexador:
- Permite que un objeto sea indexado tal como un array
 - Sólo se utiliza para hacer referencia a un array dentro de un objeto.
 - Sirve para declarar los atributos con visibilidad de protegidos y acceder desde las clases derivadas.
 - Es una propiedad de un objeto que deberá retornar una lista.
 - Todas las anteriores.
 - Ninguna de las anteriores.
7. Si quiero que un método pueda ser sobrescrito en las clases derivadas, y visto sólo por las clases derivadas, debo declararla como:
- Abstracto y protegido.
 - Público.
 - Privado.
 - Virtual y público.
 - Público y abstracto.
 - Protegido y virtual.
 - Cualquiera de las anteriores.
 - Ninguna de las anteriores.
8. Si quiero comparar si dos objetos son del mismo tipo, puedo utilizar:
- `int1 == int2`
 - `int1.GetType() == int2.GetType()`
 - `int1.Equals(int2)`
 - `int1 == typeof(int2)`
 - `typeof(int1) == typeof(int2)`
 - Todas las anteriores.
 - Ninguna de las anteriores.
9. Según el siguiente código:

```
int numero = 0;
for (; ; )
{
    numero++;
    Console.Write(numero);
    if (numero > 2)
        break;
    else
        Console.Write(" - ");
}
```

- Imprime por pantalla: 0 - 1 - 2
- Imprime por pantalla: 1 - 2 - 3
- Imprime por pantalla: 0 - 2 - 4
- Imprime por pantalla: 3 - 2 - 1
- Tiene errores de escritura.

10. Según el siguiente código, y sabiendo que el programa imprimirá por pantalla el valor pasado al constructor de ClaseB y que la función Main es correcta:

a. Marque con un **CÍRCULO** la línea incorrecta (sólo 1 línea).

b. Cómo sería esta línea correcta (sólo 1 línea): _____

```
class ClaseA
{
    int entero;
    public ClaseA(int i)
    {
        this.entero = i;
    }
}
class ClaseB : ClaseA
{
    public ClaseB(int e)
        : base(e)
    {

    }
    public int MostrarNumero
    {
        get
        {
            return base.entero;
        }
    }
}

static void Main(string[] args)
{
    ClaseB b = new ClaseB(10);
    Console.WriteLine(b.MostrarNumero);
}
```

11. Al generar una excepción propia:

- a. Debo generar al menos un constructor propio.
- b. Heredo de la clase InnerException.
- c. Tendré que generar un atributo para contener el mensaje de la excepción.
- d. Generaré una propiedad para retornar el mensaje y cumplir con el encapsulamiento pedido.
- e. La lanzaré dentro de mi código cuando lo desee por medio de la palabra reservada Throw.
- f. Todas las anteriores.
- g. Ninguna de las anteriores.

12. Test unitario:

- a. Se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo.
- b. Nuestra clase de test heredará de UnitTest.
- c. A través de métodos estáticos de la clase Assert podré informar el resultado fallido de cada test.
- d. Todas las anteriores.
- e. Ninguna de las anteriores.

13. Si tengo *class MiClase<T, S> : A<T>, B<S>, C<T, S> where T : MiTipo { }*:

- a. A, B y C sólo pueden ser interfaces.
- b. A y B son clases.
- c. A puede ser una clase o una interfaz, B y C deben ser interfaces.
- d. A y B son interfaces, C es una clase.
- e. T y S deberán ser MiTipo.
- f. S deberá ser del MiTipo o de algún tipo que herede o implemente este tipo de dato.
- g. El código es erróneo.

14. Los archivos binarios me servirán para:

- a. Serializar objetos.
- b. Serializar clases.
- c. Guardar en formato de caracteres sólo imprimibles distintos tipos de datos.
- d. Generar archivos sólo con la extensión .bin.
- e. Todas las anteriores.

15. Al ejecutar un Thread:

- a. Este finalizará y liberará el bloque de memoria que ocupaba al culminar su ejecución normal.
- b. Finalizará si se culmina la ejecución del hilo principal.
- c. Podremos ejecutar el método Abort más de una vez para el mismo método sin errores.
- d. Proporcionaremos un medio apropiado para que los objetos puedan señalar cambios de estado que pueden resultar útiles para los clientes de ese objeto.
- e. Todas las anteriores.
- f. Ninguna de las anteriores.

16. Generics:

- a. Debe utilizarse sólo uno por cada clase, método, atributo o propiedad donde se implemente.
- b. No son utilizados por las clases del sistema.
- c. Al generar la clase, reemplazaré por un comodín lo que podría ser un tipo de dato específico.
- d. Los métodos pueden recibir tipos genéricos, pero no retornarlos.
- e. Las interfaces no pueden implementarlos.
- f. Todas las anteriores.
- g. Ninguna de las anteriores.

17. Según el siguiente código:

- a. Marque con un **CÍRCULO** la línea incorrecta.
- b. Cómo sería esta línea correcta: _____

```
interface IExponer<S>
{
    S Datos { get; }
}
class ClaseBase<V>
{
    public V entero;
    public ClaseBase(V i)
    {
        this.entero = i;
    }
}
class ClaseEjecutor<int> : ClaseBase<T>, IExponer<T>
{
    public ClaseEjecutor(T e)
        : base(e)
    {
    }
    public T Datos
    {
        get
        {
            return base.entero;
        }
    }
}

static void Main(string[] args)
{
    ClaseEjecutor<string> ejecutor = new ClaseEjecutor("Hola");
    Console.WriteLine(ejecutor.Datos);
}
```

18. Los métodos de extensión son:

- a. Válidos sólo para clases.
- b. Métodos estáticos que se utilizan como métodos de instancia de la clase extendida.
- c. Métodos estáticos que se utilizan como métodos estáticos de la clase extendida.
- d. Métodos que pueden retornar cualquier tipo, pero solo reciben el tipo de la clase al que se agregarán (anteponiéndole *this*).
- e. Todas las anteriores.
- f. Ninguna de las anteriores.

19. Al generar un método de extensión:

- a. Anulo los métodos de la clase extendida.
- b. Hago referencia a la clase extendida mediante la palabra reservada *base*.
- c. Agrego funcionalidades a una clase preexistente.
- d. Creo una nueva clase que hereda de una clase preexistente.
- e. Ninguna de las anteriores.

20. Mediante la serialización:

- a. Guardo en memoria la estructura y los datos de un objeto.
- b. Almaceno en disco datos de un objeto en particular.
- c. Cargo en un archivo la estructura completa de una clase.
- d. Se crea dinámicamente una instancia de un tipo, se enlaza el tipo a un objeto existente y se puede tener acceso a sus campos y propiedades.
- e. Todas las anteriores.
- f. Ninguna de las anteriores.