

Programación II – REPASO Segundo Parcial

*Obligatorio

PRIMERA PARTE

* Seleccione la opción correcta

* Cada pregunta tiene una sola respuesta correcta

Excepciones: *

1 punto

- ☐ El bloque finally sólo se ejecuta cuando se produce una excepción.
- ☒ El bloque catch sólo se ejecuta si se produjo una excepción dentro del bloque try
- ☐ El bloque catch se ejecuta si se produjo una excepción sin importar si sucede dentro o fuera del bloque try

Según el patrón 'AAA' para la construcción de Unit Test. La sección 'Arrange': *

1 punto

- ☐ Realiza la llamada al método a probar con los parámetros preparados para tal fin.
- ☐ Comprueba que el método ejecutado se comporta tal y como teníamos previsto que lo hiciera.
- ☒ Inicializa los objetos y establece los valores de los datos que vamos a utilizar en el Test.

Si una clase implementa dos interfaces que coinciden en la firma de un método pero la implementación es diferente para cada interfaz: *

1 punto

- ☐ Se implementan las dos interfaces de forma implícita
- ☒ Se implementa al menos una de las interfaces de forma explícita
- ☐ Una clase no puede implementar más de una interfaz
- ☐ Una clase no puede implementar interfaces que tengan métodos con la misma firma

Cual de las siguientes declaraciones restringe el parámetro T a que sea de tipo referencia? *

1 punto

- ☒ class Mock<T> where T: class
- ☐ class Mock<T> where T: object
- ☐ class Mock<T> where T = new()
- ☐ class Mock<T> where T: new()

Marque la correcta sobre serialización en XML: *

1 punto

- ☒ La clase a serializar debe tener un constructor sin parámetros
- ☐ Se pueden serializar atributos públicos y privados.
- ☐ Se pueden serializar atributos y propiedades públicas y privadas.
- ☐ Todas son correctas

La Primary Key / Clave Primaria: *

1 punto

- ☐ Es auto-incremental por defecto
- ☒ Puede ser de cualquier tipo que admita el motor de base de datos.
- ☐ Puede ser null.
- ☐ Sólo puede ser de tipo int.

Eventos: *

1 punto

- ☐ Los eventos permiten que una clase u objeto notifique a otras clases u objetos cuando ocurre algo de interés
- ☐ Un evento puede tener varios suscriptores/manejadores. Un suscriptor/manejador puede manejar varios eventos de varios emisores.
- ☐ Los eventos que no tienen suscriptores/manejadores nunca se generan.
- ☒ Todas son correctas.

¿Cuál es la razón de utilizar consultas parametrizadas cuando trabajamos con las clases SqlConnection y SqlCommand? *

1 punto

- ☒ Evitar la inyección de sql
- ☐ Generar código más limpio y ordenado
- ☐ La consultas parametrizadas son la unica forma que tenemos de realizar consultas SQL
- ☐ Todas son correctas

¿Cuál es la función de los delegados? *

1 punto

- ☐ Nos permiten encapsular las referencias a métodos de una instancia dentro de un objeto de tipo delegado
- ☐ Nos permite encadenar varios métodos manejadores disparando un solo evento
- ☐ Nos permiten definir la firma que todos nuestros métodos manejadores van a tener dándonos una seguridad de tipos
- ☒ Todas son correctas

Marque la respuesta correcta *

1 punto

```
1 referencia
public class Pista
{
    List<Thread> hilos;
    bool sigueLaCarrera;

    0 referencias
    public Pista()
    {
        this.hilos = new List<Thread>();
        this.sigueLaCarrera = false;
    }

    0 referencias
    public void Correr(D deportista)
    {
        if (this.sigueLaCarrera)
        {
            Thread nuevoHilo = new Thread(deportista.Moverse);
            this.hilos.Add(nuevoHilo);
            nuevoHilo.Start();
        }
    }
}
```

- ☒ Si ejecuto ".Abort()" una vez que terminó la ejecución del método Moverse, se lanzará una excepción.
- ☐ Una vez finalizado el hilo podré volver iniciarlo ejecutando "nuevoHilo.Start()"
- ☐ Dará error en tiempo de compilación ya que el método Start de Thread no recibe argumentos.
- ☐ Todas son correctas

Atrás

Siguiente

Nunca envíes contraseñas a través de Formularios de Google.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) · [Términos del Servicio](#) · [Política de Privacidad](#)