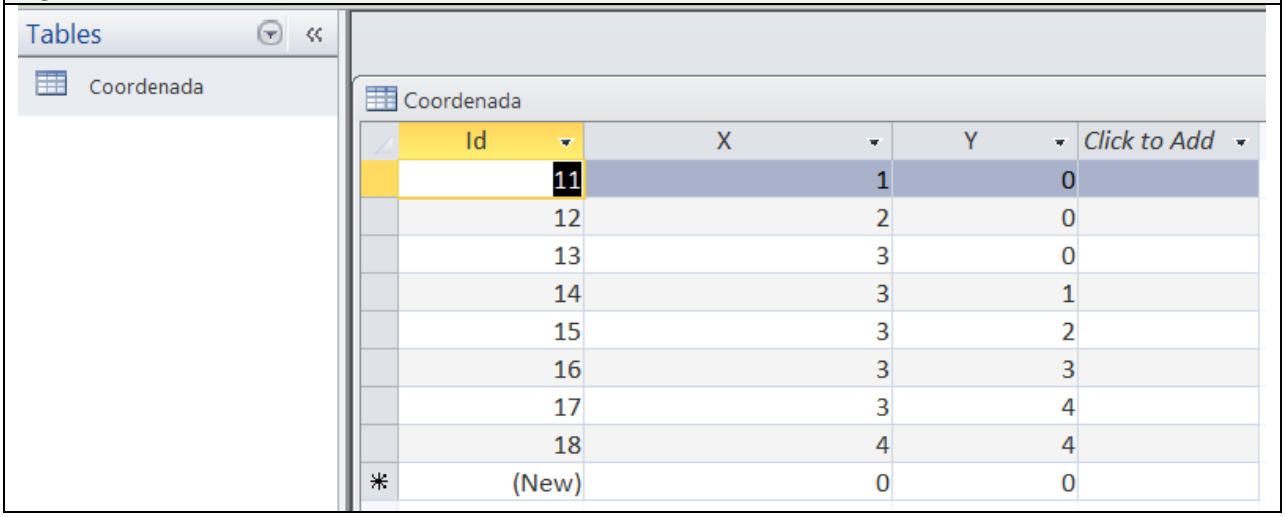
Enunciado:

Tema: ADO - Bases de datos.

Utilizando el código prexistente de nuestra aplicación víbora, se pide modificarlo para que cada movimiento de la misma sea grabado en una base de datos del tipo Microsoft Access cuyo archivo es **CoordenadaDB.mdb**, y su ubicación en el file system (en disco) será "…\Unidad 9\02-Vibora\AplicacionVibora\Database".

La **Figura 1**, muestra la estructura de la tabla "Coordenada", cada coordenada se almacena como un valor **X** y otro de **Y**. Esto quiere decir que la víbora se movió de la coordenada "1; 0" a la "2; 0" y luego a la "3; 0" y así sucesivamente, de esta manera podríamos reproducir el camino que la víbora recorrió cosa que es nuestro objetivo.

Figura 1



La **Figura 2**, muestra la clase que debemos desarrollar llamada "**Persistidor**" y cuya responsabilidad es la lectura y escritura de los registros en la tabla "Coordenada".

Figura 2

```
16  public static class Persistidor
17  {
18      /// <summary>
19      /// Graba las coordenadas x,y de la cabeza de
20      /// la víbora en una base de datos en disco.
21      /// </summary>
22      /// <param name="vibora"></param>
23      public static void GrabarCoordenadas(Vibora vibora)...
45
46      /// <summary>
47      /// Carga el camino recorrido por la víbora desde una base de datos en disco.
48      /// </summary>
49      /// <returns></returns>
50      public static List<Asterisco> LeerCaminoGrabado()...
77
78      /// <summary>
79      /// Elimina el camino grabado en una base de datos.
80      /// </summary>
81      public static void EliminarCaminoGrabado()...
94  }
```

La **Figura 3**, muestra el prototipo y la descripción de cada uno de los métodos que componen la clase "**Persistidor**".

Figura 3	
Prototipo de los métodos.	Descripción
`public static void GrabarCoordenadas(Vibora vibora)`	Graba las coordenadas x,y de la cabeza de la víbora en la base de datos en disco.
`public static List<Asterisco> LeerCaminoGrabado()`	Carga el camino recorrido por la víbora leído desde la base de datos.
`public static void EliminarCaminoGrabado()`	Elimina el camino grabado en la base de datos.

La **Figura 4**, muestra la rutina de atención al evento **OnAfterMover**, que lo único que hace es grabar las coordenadas que posee la cabeza de la víbora en la pantalla, para esto el código solo invoca al método "`Persistidor.GrabarCoordenadas(vibora);`".

Figura 4

```
/// <summary>
/// Rutina de atención al evento OnAfterMover.
/// </summary>
/// <param name="vivora"></param>
static void vivora_OnAfterMover(Vibora vibora)
{
    Persistidor.GrabarCoordenadas(vibora);
}
```

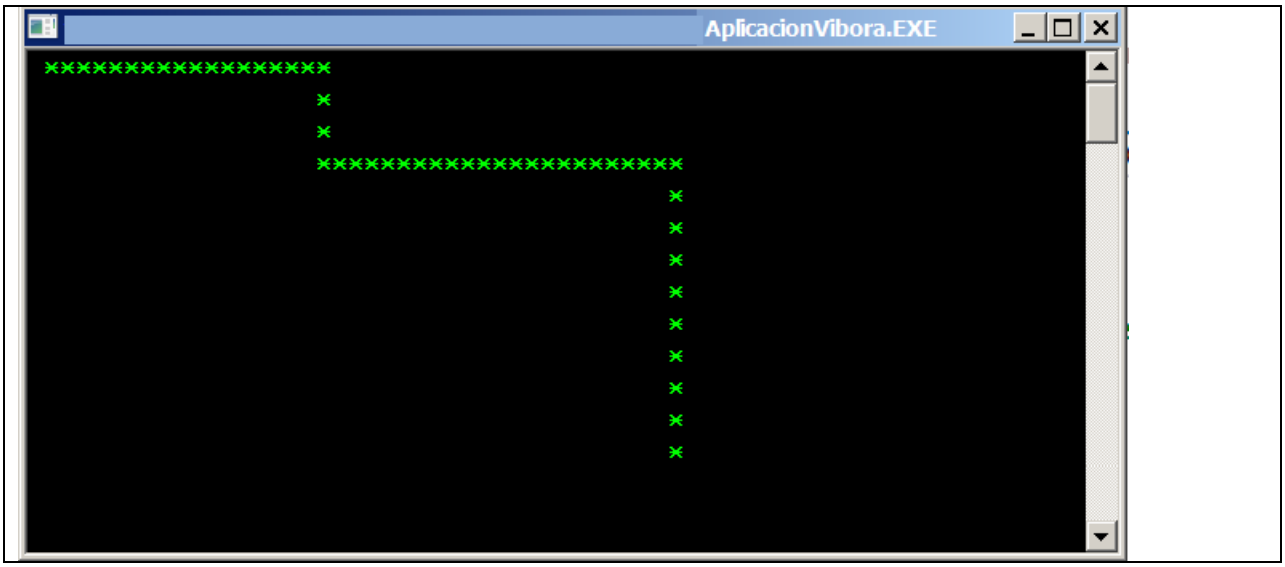
La **Figura 5**, muestra el método responsable de mostrar el camino que ha recorrido la víbora, para esto lee el camino grabado en disco en la base de datos y luego imprime cada uno de los asteriscos en la pantalla.

Figura 5

```
/// <summary>
/// Muestra el camino recorrido por la víbora leyéndolo desde un
/// archivo de texto utilizado como base de datos.
/// </summary>
private static void MostrarCaminoRecorrido()
{
    Console.Clear();
    List<Asterisco> caminoDeAsteriscos = Persistidor.LeerCaminoGrabado();
    foreach (Asterisco asterisco in caminoDeAsteriscos)
    {
        asterisco.Imprimir();
    }
    Console.ReadKey();
}
```

La **Figura 6**, muestra la aplicación ejecutándose y vemos el camino que la víbora ha recorrido, el mismo se activa presionando la tecla **Enter,** luego presionando una tecla más la víbora continua comportándose normalmente.

Figura 6


AplicacionVibora.EXE