...

PRÁCTICAS

POO en C#

Programación de Apps Grado 9°

Profesor : Ariana Gómez Contreras

Patrones de diseño

Introducción a los patrones de diseño

Un design pattern o patrón de diseño representa un diagrama de objetos que forma una solución a un problema conocido que se presenta con frecuencia. El diagrama de objetos se constituye de un conjunto de objetos descritos por clases y las relaciones que enlazan los objetos.

Los patrones dan una solución a problemas de diseño de aplicaciones en el ámbito de la programación orientada a objetos. Se trata de soluciones conocidas y probadas cuyo diseño es producto de la experiencia de los programadores. No existe un aspecto teórico en los patrones, en especial no existe una formalización (a diferencia de los algoritmos).

EJERCICIOS

El patrón Singleton tiene como objetivo asegurar que una clase sólo posee una instancia y proporcionar un método de clase de acceso global.

Las clases y objetos que participan en este patrón son:

Singleton: Define una operación de instancia que permite a los clientes acceder a su instancia única.

La instancia es una operación de clase, responsable de crear y mantener su propia instancia única.

Codigo en c#

```
using System;

namespace DoFactory.GangOfFour.Singleton.Structural
{


  /// Patron Singleton
  class MainApp
  {

    static void Main()
    {
      // no se puede usar el operador new, ya que el constructor tiene acceso protected
      Singleton s1 = Singleton.Instance();
      Singleton s2 = Singleton.Instance();

      // Probando la misma instancia
      if (s1 == s2)
      {
        Console.WriteLine("2 objetos, misma instancia");
      }

      //Espera indicación del usuario
      Console.ReadKey();
    }
  }

  /// Clase Singleton

  class Singleton
  {
    private static Singleton _instance;

    // Constructor is 'protected'
    protected Singleton()
    {
    }

    public static Singleton Instance()
    {

      if (_instance == null)
      {
        _instance = new Singleton();
      }

      return _instance;
    }
  }
}
```