**Patrones de diseño**

Los patrones de diseño representan las mejores prácticas utilizadas por los desarrolladores de software orientado a objetos experimentados. Son soluciones a los problemas generales que enfrentan los desarrolladores de software durante el desarrollo de software.

**GOF Ganf of Four 🡪 1994**

* Erich Gamma
* Richard Helm
* Ralph Johnson
* John Vlissides

Libro: Design Patterns - Elements of Reusable Object-Oriented Software.

De acuerdo con estos autores patrones de diseño se basan principalmente en los siguientes principios de diseño orientado a objetos.

* Programa a una interfaz no una implementación.
* Favorecer composición de objetos sobre la herencia.

“Crear clases de las que se utilicen sus métodos pero estas no puedan utilizarlas, es decir, usar interfaces en lugar de herencia”.

**Categorías de los patrones de diseño**

|  |  |
| --- | --- |
| **SN** | **Patrón y Descripción** |
| 1 | **Creational Patterns**  Estos patrones de diseño proporcionan una forma de crear objetos al tiempo que oculta la lógica de la creación, en lugar de crear instancias de objetos directamente a través de la nueva opreator. Esto da más flexibilidad en el programa de decidir qué necesitan ser creados para un caso de uso dado objetos. |
| 2 | **Structural Patterns**  Estos patrones de diseño de clase preocupación y la composición de objetos. Concepto de herencia se utiliza para componer las interfaces y definir maneras de componer objetos para obtener nuevas funcionalidades. |
| 3 | **Behavioral Patterns**  Estos patrones de diseño se refiere específicamente con la comunicación entre objetos. |
| 4 | **J2EE Patterns**  Estos patrones de diseño se refiere específicamente a la capa de presentación. Estos patrones son identificados por Sun Java Center. |