

Conceptos Básicos de OO

para introducción de
Patrones de Diseño



Conceptos Básicos de OO

para introducción de
Patrones de Diseño



Abstracción:

- Propiedades de los objetos.
- Tipos.
- El mundo abstracto y el mundo físico.
- Estructura y comportamiento

Polimorfismo:

lo usamos cuando queremos trabajar un objeto con el tipo base, no con el derivado, (subtipo).

además, en el polimorfismo ocurren varias cosas...

- Se usa composición combinada con herencia.
- Existe una amplia reutilización de la interface.
- Permite escribir funcionalidad que no dependa de tipos específicos sino generales.
- Se extiende la aplicación fácilmente agregando nuevos subtipos.
- Trata a subtipos como si fueran supertipos.

Encapsulamiento:

permite ocultar aspectos que no se desean que estén al alcance de algún tipo de usuario.

Encapsulamos estructura y comportamiento, o sea

- clases, atributos y métodos....
 - mediante visualización,
 - mediante clases abstractas e interfaces

Composición

Igual que la herencia sirve para extender funcionalidad, los objetos dentro de una composición, son definidos generalmente como "privado" aunque pueden ser definidos como "público"; depende de las intenciones de encapsulamiento.

al efectivizar composición, a la funcionalidad de la clase contenedora la "extiende" en funcionalidad.

siempre es mejor extender con composición, porque la misma es llevada a cabo en tiempo de ejecución.

La composición no rompe con el encapsulamiento

Herencia

Podemos clonar una clase, (tanto en estructura como comportamiento), y además extenderla en funcionalidad.

No tiene flexibilidad en cuanto a que el compilador debe resolver restricciones en tiempo de compilación. Rompe con el encapsulamiento

la clase tiene un tipo, las subclases tienen un subtipo que pertenece al supertipo, (tipo de la clase).

la implementación y la ejecución es llevada siempre por los subtipos, más allá que se hereda estructura y comportamiento.

todos los mensajes que se pueden enviar a los objetos de la clase, se pueden enviar a los objetos de las subclases, porque son del mismo tipo

