

THE ART OF ENBUGGING

- Una de las mejores formas de evitar futuros errores es diseñar el código de modo que las clases y módulos tengan responsabilidades claras, bien definidas, aisladas y una semántica que se entienda bien. Esto es difícil y requiere experiencia, lo que quiere decir hacerlo mal muchas veces e ir aprendiendo.
- El objetivo fundamental es escribir código tímido que no revela más de lo necesario, se mantiene para sí mismo y nunca mostraría sus “privados” a sus “amigos”.
- Tell, don't ask.
- El código de procedimiento tiende a obtener información y luego tomar decisiones basadas en ella. El código orientado a objetos le dice a los objetos que hagan cosas. La lógica que está implementando es probablemente responsabilidad del objeto llamado, no suya como persona que llama. Tomar decisiones fuera del objeto viola su encapsulación y tiende a tener errores. Queremos decirle a los objetos qué hacer, no preguntarles su estado.
- Que la persona que llama no sepa demasiado sobre cómo se ejecutará su comando significa que hemos reducido el acoplamiento. Los métodos de consulta no deben tener efectos secundarios, solo brindar información sobre el estado del objeto que puede servir, por ejemplo, para los tests.
- También se desea hablar con la menor cantidad posible de objetos ya que sino tenderá más a romperse. Una herramienta útil para este problema se llama “Ley de Demeter para funciones”. Esta buena idea sugiere que un objeto solo debería llamar:
 - A sí mismo
 - Cualquier parámetro pasado en el método.
 - Cualquier objeto que creó.
 - Cualquier objeto de componente contenido directamente.
- Programar cerca del dominio del problema.
- A largo plazo, el acoplamiento de clase superior es inaceptable, ya que aumenta las posibilidades de que cualquier cambio que se realice rompa algo en otro lugar.
- En aquellos casos en las que la velocidad es primordial y el acoplamiento alto es aceptable, acóplarlo al máximo. Dejar claro en la documentación que estas clases particulares están unidas entre sí y por qué.