

Ароматы дизайна

1. Жёсткость — сложно внести изменения.
2. Хрупкость — внёс изменения и всё сломалось.
3. Косность — части кода могут пригодиться где-либо ещё, но их никак не выдрать.
4. Вязкость..
 - a. ..программы — сложно вносить правки в код;
 - b. ..окружения — сложно и долго пересобирать программу; как быстрее.
5. Ненужная сложность.
6. Ненужные повторения.
7. Непредсказуемость.

S.O.L.I.D.

1. Single responsibility — на каждый объект должна быть возложена одна единственная обязанность. Примером ошибки может быть божественный объект.
2. Open-closed — все классы, функции и т. д. должны проектироваться так, чтобы для изменения их поведения, нам не нужно было изменять их исходный код. Примером является функция, которая завязана на конкретной БД, например (БД можно передавать интерфейсным параметром).
3. Liskov substitution — объекты в программе могут быть заменены их наследниками без изменения свойств программы. Предусловия не могут быть усилены в подклассе. Постусловия не могут быть ослаблены в подклассе. Пример: прямоугольник и квадрат.
4. Interface segregation — много специализированных интерфейсов лучше, чем один универсальный. Примером является интерфейс, который делает с товаром всё (например, у книги не может быть размера).
5. Dependency inversion — зависимости должны строиться относительно абстракций, а не деталей. Примером является класс, который использует другой класс, инстанцируя его в себе (не обязательно). Решение — интерфейсные параметры.