Software Pattern 236700

Home Work 0 - Dry Part

Examples of a misuse of inheritance:

1. “Properties extends Hashtable” – easier implementation shouldn’t lead the decision of extending a class. Instead, the relation “is a” (Liskov principle), which we learned in OOP course, should guide us to determine the right abstraction in each situation we inspect inheritance solution. Also, the example above is an “implementation reveler”, means it give a hint about the implementation of the Class - Properties: using a hash table to map certain strings to other certain strings.
2. Once creating a class, the notion of inheritance should be in mind because inheritance might violate encapsulation if not planned right. If a class is not supposed to be extended the healthy solution is to make the class final in terms of inheritance, so confused users will not break the behavior of some functionality. And in the other direction, if a class abstractly might be extended, you should build around that notion (implementation).
3. No or poor documentary. Avoid user mistakes/confusion by documenting the class properly for allowance of inheritance: documentation of method dependency, guidelines and edge cases for implementation inheritance of our class.
4. Bad naming of classes like many classes in J2SE libraries. Naming a class as an intent reveler, classes like AbstractSet, AbstractMap give a hint to the user that those classes are designed or even expected to be extended.

Examples of a misuse of exceptions:

1. A method is forced to declare the exceptions it possibly can throw even if it practically will never throw them.
2. Fail slow, exception thrown somewhere in the middle of execution. Try to disallow putting garbage into your methods, but if it happened anyway (passed compilation) an exception should be thrown as fast as possible.