

Class Quality Checklist

Abstract Data Types

- Have you thought of the classes in your program as Abstract Data Types and evaluated their interfaces from that point of view?

Abstraction

- Does the class have a central purpose?
- Is the class well named, and does its name describe its central purpose?
- Does the class's interface present a consistent abstraction?
- Does the class's interface make obvious how you should use the class?
- Is the class's interface abstract enough that you don't have to think about how its services are implemented? Can you treat the class as a black box?
- Are the class's services complete enough that other classes don't have to meddle with its internal data?
- Has unrelated information been moved out of the class?
- Have you thought about subdividing the class into component classes, and have you subdivided it as much as you can?
- Are you preserving the integrity of the class's interface as you modify the class?

Encapsulation

- Does the class minimize accessibility to its members?
- Does the class avoid exposing member data?
- Does the class hide its implementation details from other classes as much as the programming language permits?
- Does the class avoid making assumptions about its users, including its derived classes?
- Is the class independent of other classes? Is it loosely coupled?

Inheritance

- Is inheritance used only to model "is a" relationships?
- Does the class documentation describe the inheritance strategy?
- Do derived classes adhere to the Liskov Substitution Principle?

- Do derived classes avoid "overriding" non overridable routines?
- Are common interfaces, data, and behavior as high as possible in the inheritance tree?
- Are inheritance trees fairly shallow?
- Are all data members in the base class private rather than protected?

Other Implementation Issues

- Does the class contain about seven data members or fewer?
- Does the class minimize direct and indirect routine calls to other classes?
- Does the class collaborate with other classes only to the extent absolutely necessary?
- Is all member data initialized in the constructor?
- Is the class designed to be used as deep copies rather than shallow copies unless there's a measured reason to create shallow copies?

Language-Specific Issues

- Have you investigated the language-specific issues for classes in your specific programming language?