

6.7 CHECKLIST: Class Quality

6.7.1 Abstract Data Types

6.7.1.1 have you thought of the classes in your program as abstract data types and evaluate their interfaces from that point of view?

6.7.2 Abstraction

6.7.2.1 Does the class have a central purpose?

6.7.2.2 Is the class well-named, and does its name describe its central purpose?

6.7.2.3 Does the class's interface present a consistent abstraction?

6.7.2.4 Is the class's interface make obvious that you don't have to think how its services are implemented? Can you treat the class as a black box?

6.7.2.5 Are the class's service complete enough that other classes don't have to meddle with its internal data?

6.7.2.6 Has unrelated information been moved out of the class?

6.7.2.7 Have thought about subdividing the class into component classes, and have you subdivided it as much as you can?

6.7.2.8 Are you preserving the integrity of the class's interface as you modify the class?

6.7.3 Encapsulation

6.7.3.1 Does the class minimize accessibility to its members?

6.7.3.2 Does the class avoid exposing member data?

6.7.3.3 Does the class hide its implementation as much as the programming language permits?

6.7.3.4 Does the class avoid making assumption about its users, including its derived classes?

6.7.3.5 Is the class independent of other classes? Is it loosely coupled?

6.7.4 Inheritance

6.7.4.1 Is inheritance used only to model "is a" relationships - that is, do derived classes adhere to the Loskov Substitution Principle?

6.7.4.2 Does the class documentation describe the inheritance strategy?

6.7.4.3 Does derive class avoid "overriding" non-overridable routines?

6.7.4.4 Are inheritance tree fairly shallow?

6.7.4.5 Are all data members in the base class private rather than protected?

6.7.5 Other Implementation Issues

6.7.5.1 Does the class contains about 7 data members or fewer?

6.7.5.2 Does the class minimize direct or indirect routine calls to other classes(minimize direct or indirect 调用其他类的子程序)?

6.7.5.3 Does the class collaborate with other classes only to extent absolutely necessary?

6.7.5.4 Is all member data initialized in the constructor?

6.7.5.5 Is the class design to be used as deep copies rather than shallow copies unless there's a measured reason to create shallow copies?

6.7.6 Language-Specific Issues

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