

2. Defensive Programming

Defensive Programming: Introduction

- CON52-J. Document thread-safety and use annotations where applicable
- DCL53-J. Minimize the scope of variables
- DCL56-J. Do not attach significance to the ordinal associated with an enum
- DCL58-J. Enable compile-time type checking of variable arity parameter types
- DCL59-J. Do not apply public final to constants whose value might change in later releases
- DCL60-J. Avoid cyclic dependencies between packages
- ERR51-J. Prefer user-defined exceptions over more general exception types
- ERR53-J. Try to gracefully recover from system errors
- FIO51-J. Identify files using multiple file attributes
- IDS50-J. Use conservative file naming conventions
- MET54-J. Always provide feedback about the resulting value of a method
- MSC50-J. Minimize the scope of the @SuppressWarnings annotation
- MSC53-J. Carefully design interfaces before releasing them
- NUM52-J. Be aware of numeric promotion behavior
- OBJ51-J. Minimize the accessibility of classes and their members
- OBJ52-J. Write garbage-collection-friendly code

