Coding Standards

|  |  |  |
| --- | --- | --- |
| Coding standard | Cases | Examples |
| Class | Start with uppercase , usually noun | Public class Animal |
| Interface | Start with uppercase, usually adjective | Public interface Runnable |
| Methods | Start with lower case and camel case | Public void getName (){} |
| Variables | Start with lower case and camel case | Private int calculateSum; |
| Constants | Only in uppercase word are separated by underscores | Final int DATABSE\_VERSION; |

* Self-explanatory:
* Meaningful distinctions:
* Pronounceable:

Ex: generationTimestamp better than genStamp

Best practice

|  |  |
| --- | --- |
| **Best Practice** | **Examples** |
| Use proper name Conventions | Public class Teacher{  String techerName;  } |
| Use underscore in lengthy numerical values | Int num=58\_845\_754; |
| Use StringBuilder for String concatenation | StringBuilder sql=”Select name, age, gender”;  Sql.append(”from Student where id=1”); |
| Using foreach loop | String names={“Same”,”Peter”,”john”};  For(String name:names){  System.out.println(name); |
| Proper handing of null point exception | private int getListOfEmployees(File[] files) {  if (files == null)  throw new NullPointerException("File list cannot be null"); |
| Avoid empty catch blocks |  |