**Item1 : Consider static factory methods instead of constructors**

**用mysql例子举例：**

Class.*forName*("com.mysql.jdbc.Driver");

在客户端使用这个时，由于反射中默认是加载并初始化这个类（com.mysql.jdbc.Driver），所以这个类的静态初始化块会被执行，而当我们查看com.mysql.jdbc.Driver的实现时会发现如下这段代码，即调用jdk的DriverManager的API把自己给加载进来并到java.sql.DriverManager注册下。

**static** {

**try** {

java.sql.DriverManager.*registerDriver*(**new** Driver());

} **catch** (SQLException E) {

**throw** **new** RuntimeException("Can't register driver!");

}

}

这样，当下面的代码中有对java.sql.DriverManager的引用时，这里的jdbc操作都是委托给了这里加载的com.mysql.jdbc.Driver。

完整代码实现如下：

**import** java.sql.DriverManager;

**import** com.mysql.jdbc.Connection;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

Connection con = (Connection) DriverManager.*getConnection*(

"jdbc:mysql://127.0.0.1:3306/test", "root", "root");

} **catch** (Exception e) {

}

}

}

具体static factory methods样例 思路请参考代码